

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КАЛИНИНГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

В. В. Капустин

ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Учебно-методическое пособие по выполнению лабораторных работ
для студентов бакалавриата по направлению
подготовки 09.03.01 Информатика и вычислительная техника,
профиль «Автоматизированные системы
обработки информации и управления»

Калининград
Издательство ФГБОУ ВО «КГТУ»
2022

УДК 338

Рецензент:

кандидат технических наук,
профессор кафедры цифровых систем и автоматики института
цифровых технологий ФГБОУ ВО «Калининградский государственный
технический университет» В. А. Петрикин

Капустин, В. В.

Вычислительная техника: учеб.-метод. пособие по выполнению лабораторных работ для студентов бакалавриата по направлению подготовки 09.03.01 Информатика и вычислительная техника, профиль «Автоматизированные системы обработки информации и управления» / **В. В. Капустин.** – Калининград: Изд-во ФГБОУ ВО «КГТУ», 2022. – 89 с.

В учебно-методическом пособии даны методические указания по самостоятельному изучению дисциплины в ходе подготовки к лабораторным занятиям.

Пособие подготовлено в соответствии с требованиями утвержденной рабочей программы профессионального модуля 09.03.01 Информатика и вычислительная техника.

Учебно-методическое пособие рассмотрено и одобрено в качестве локального электронного методического материала кафедрой цифровых систем и автоматики 29 сентября 2022 г., протокол № 2

Учебно-методическое пособие по выполнению лабораторных работ рекомендовано к использованию в качестве локального электронного методического материала в учебном процессе методической комиссией института цифровых технологий ФГБОУ ВО «Калининградский государственный технический университет» 29 сентября 2022 г., протокол № 7

© Федеральное государственное
бюджетное образовательное учреждение
высшего образования «Калининградский
государственный технический
университет», 2022 г.

© Капустин В.В., 2022 г.

ОГЛАВЛЕНИЕ

1. Введение	6
2. Лабораторная работа № 1. Элементы процессоров ЭВМ. Исследование логических элементов	7
2.1. Общие сведения	7
2.2. Теоретическое введение	7
2.3. Задание к лабораторной работе	8
2.4. Методические указания и порядок выполнения работы.....	8
2.4.1. Исследование элемента NOT (операция инверсии).....	8
2.4.2. Исследование логического элемента AND (операция конъюнкции).....	11
2.4.3. Исследование логического элемента OR (операция дизъюнкции)	14
2.4.4. Исследование сложной комбинационной схемы (преобразователь кодов)	15
2.5. Требования к отчету и защите	16
3. Лабораторная работа № 2. Элементы процессоров ЭВМ. Исследование сумматоров и АЛУ	16
3.1. Общие сведения	16
3.2. Теоретическое введение	17
3.2.1. Порядок проектирования комбинационных схем.....	17
3.3. Задание к лабораторной работе	20
3.4. Методические указания и порядок выполнения работы.....	20
3.4.1. Исследование шифраторов	20
3.4.2. Исследование мультиплексора	20
3.4.3. Исследование демультимплексора	21
3.4.4. Исследование сумматоров и АЛУ	22
3.5. Требования к отчету и защите	23
4. Лабораторная работа № 3. Элементы процессоров ЭВМ. Исследование последовательностных элементов	24
4.1. Общие сведения	24
4.2. Теоретическое введение	24
4.3. Задание к лабораторной работе	27
4.4. Методические указания и порядок выполнения работы.....	27
4.4.1. Исследование асинхронного RS-триггера	27
4.4.2. Исследование синхронного RS-триггера со статическим синхрвходом	29
4.4.3. Исследование синхронного RS-триггера с динамическим синхрвходом	30
4.5. Требования к отчету и защите	31

5. Лабораторная работа № 4. Исследование элементов регистровой и оперативной памяти ЭВМ. JK-, D-триггеры, регистры, счетчики	32
5.1. Общие сведения	32
5.2. Теоретическое введение	32
5.3. Методические указания и порядок выполнения работы.....	33
5.3.1. Исследование JK-триггера	33
5.3.2. Исследование D-триггера.....	34
5.3.3. Исследование двухтактного триггера	36
5.3.4. Исследование асинхронных счетчиков.....	37
5.3.5. Исследование регистров.....	38
5.4. Требования к отчету и защите	40
6. Лабораторная работа № 5. Запоминающие устройства ЭВМ. Исследование структур и функций. Синхронные счетчики. Микропрограммные устройства	42
6.1. Общие сведения	42
6.2. Теоретическое введение	42
6.3. Задание к лабораторной работе	44
6.4. Методические указания и порядок выполнения работы.....	44
6.4.1. Исследование синхронных счетчиков	44
6.5. Требования к отчету и защите	46
7. Лабораторная работа № 6. Микропрограммные устройства и контроллеры. Организация и типовые структуры. Проектирование специализированного микропрограммного автомата	47
7.1. Общие сведения	47
7.2. Теоретическое введение	47
7.3. Задание к лабораторной работе	50
Методические указания и порядок выполнения работы	53
7.4. Требования к отчету и защите	53
8. Лабораторная работа № 7. Система конструктивных элементов персонального компьютера	55
8.1. Общие сведения	55
8.2. Теоретическое введение	55
8.2.1. Основные сведения	56
8.3. Задание к лабораторной работе	57
8.4. Методические указания и порядок выполнения работы.....	57
8.4.1. Правила техники безопасности:	57
8.4.2. Ход работы.....	58
8.5. Требования к отчету и защите	59
9. Лабораторная работа № 8. Проблемы интеграции компонентов системного ядра персонального компьютера.....	60
9.1. Общие сведения	60

9.2.	Теоретическое введение	60
9.2.1.	Проблемы системы электропитания	61
9.2.2.	Отсутствие сигналов и сообщений об ошибках.....	63
9.2.3.	Сообщения об ошибках	64
9.2.4.	Наиболее важные настройки.....	65
9.2.5.	Диагностические программы	68
9.3.	Задание к лабораторной работе	69
9.4.	Методические указания и порядок выполнения работы.....	69
9.4.1.	Ход работы.....	69
9.5.	Требования к отчету и защите	69
10.	Лабораторная работа № 9. Исследование ресурсов системного ядра персонального компьютера	71
10.1.	Общие сведения.....	71
10.2.	Теоретическое введение	71
10.2.1.	Основные сведения	72
10.3.	Задание к лабораторной работе	74
10.4.	Методические указания и порядок выполнения работы	74
10.4.1.	Работа с отладчиком и его основные команды	74
10.4.2.	Ход работы.....	76
10.5.	Требования к отчету и защите	78
11.	Лабораторная работа № 10. Установка и загрузка операционной системы	79
11.1.	Общие сведения.....	79
11.2.	Теоретическое введение	79
11.2.1.	Основные сведения	80
11.3.	Задание к лабораторной работе	82
11.4.	Методические указания и порядок выполнения работы	82
11.4.1.	Установка ОС с дистрибутива на CD или DVD.....	82
11.4.2.	Установка ОС с образа диска.....	83
11.4.3.	Установка ОС из архива	84
11.4.4.	Ход работы.....	84
11.5.	Требования к отчету и защите	85
12.	Заключение.....	86
13.	Литература	86

1. ВВЕДЕНИЕ

Данное учебно-методическое пособие предназначено для студентов направления подготовки 09.03.01 Информатика и вычислительная техника, изучающих дисциплину «Вычислительная техника».

Целью освоения дисциплины «Вычислительная техника» является формирование знаний и основ построения устройств и элементов вычислительных систем, необходимых для выбора их архитектуры, конфигурации и эксплуатации:

- изучение элементов, узлов и устройств, позволяющих реализовать функции обработки данных и управления в электронных вычислительных машинах, принципов построения запоминающих и внешних устройств и их интерфейсов;
- приобретение навыков выбора, комплексирования и эксплуатации программно-аппаратных средств в вычислительных и информационных системах;
- формирование базовых знаний, умений и навыков для успешного (в том числе самостоятельного) выбора элементной базы для построения различных архитектур вычислительных средств.

Лабораторный практикум содержит 10 лабораторных работ.

Лабораторные работы проводятся в лабораториях, оснащенных персональными компьютерами, объединенными в локальную сеть с доступом в Интернет.

В результате выполнения лабораторных работ ожидается, что студенты должны **знать:** классификацию и типовые узлы вычислительной техники;

- архитектуру микропроцессорных систем;
- основные методы цифровой обработки сигналов.
- элементную базу аналоговой и цифровой электроники, математические принципы создания и методы синтеза функциональных узлов вычислительных устройств, базовые принципы построения и принципы работы основных функциональных узлов цифровых устройств;

уметь: использовать различные средства вычислительной техники и программного обеспечения в профессиональной деятельности;

- использовать различные виды обработки информации и способы представления ее в электронно-вычислительных машинах (ЭВМ).
- работать со структурными и функциональными схемами цифровых устройств,
- использовать профессиональную терминологию;

владеть: навыками использования диагностического оборудования при анализе работы цифровых устройств и средств вычислительной техники.

2. ЛАБОРАТОРНАЯ РАБОТА № 1. ЭЛЕМЕНТЫ ПРОЦЕССОРОВ ЭВМ. ИССЛЕДОВАНИЕ ЛОГИЧЕСКИХ ЭЛЕМЕНТОВ

2.1. Общие сведения

Цель:

изучение номенклатуры и функций типовых логических элементов

Материалы, оборудование, программное обеспечение:

IBM PC-совместимый персональный компьютер, программа MicroCap 9.0 5.0 Evaluation version, или иная, также являющаяся свободно распространяемой демоверсией профессиональной программы машинного моделирования электронных схем.

Условия допуска к выполнению:

Предварительное знакомство с УМП, отсутствие существенных текущих задолженностей по лабораторному практикуму.

Критерии положительной оценки:

Выполнение типового задания, оформление отчета по работе, ответы на вопросы для самопроверки.

Планируемое время выполнения:

Аудиторное время выполнения (под руководством преподавателя): 4 ч.

Время самостоятельной подготовки: 2 ч.

2.2. Теоретическое введение

В цифровой электронике используются три базовых логических элемента, это элемент NOT - логическое отрицание (инверсия), элемент AND - логическое умножение (И), и элемент OR - логическое сложение (ИЛИ).

Типовые логические элементы и устройства служат основой для создания ЭВМ и автоматов дискретного действия. Системой логических элементов называется функционально полный набор логических элементов, объединенных общими электрическими, конструктивными и технологическими параметрами и использующих одинаковый тип межэлементных связей. Системы элементов содержат элементы для выполнения логических операций, запоминающие элементы, элементы, реализующие функции узлов ЭВМ, а также элементы для усиления, восстановления и формирования сигналов стандартной формы.

В логических устройствах сигнал на входе и выходе каскада является бинарным. Он может принимать только два значения – логического нуля «0» и логической единицы – «1». Значения «0» и «1» являются символическими и не соответствуют числовым значениям напряжения, выражаемым в вольтах. Например, при использовании выходного напряжения ключевого каскада уровнем логического «0» может служить напряжение на коллекторе насыщенного транзистора, уровнем логической «1» – напряжение на коллекторе запертого транзистора.

Комбинационные логические схемы состоят из элементов, выходной сигнал которых зависит только от входных сигналов в рассматриваемый момент времени.

Собственно логическими элементами называют элементы, реализующие функции булевой алгебры. Базовые логические элементы реализуют минимальный набор функций, достаточный для построения любых устройств цифровой электроники.

Устройства на основе логических элементов можно также разделить на комбинационные и последовательностные.

Комбинационные устройства состоят из элементов, выходной сигнал которых зависит только от входных сигналов в рассматриваемый момент времени.

Последовательностные устройства содержат элементы памяти, таким образом, их выходной сигнал зависит не только от входных сигналов в рассматриваемый момент времени, но и от их значений в предыдущие моменты времени.

Литература:

Кардашев, Г.А. Цифровая электроника на персональном компьютере. Electronics Workbench и Micro-Cap / Г. А. Кардашев. - Москва : Горячая линия-Телеком, 2003. Стр. 5-34

Суханова, Н.В. Основы электроники и цифровой схемотехники [Электронный ресурс] : учеб. пособие / Н.В. Суханова ; Министерство образования и науки РФ, Воронежский государственный университет инженерных технологий. - Воронеж : Воронежский государственный университет инженерных технологий, 2017. - 97 с. (ЭБС «Университетская библиотека онлайн») Стр. 68-73

Контрольные вопросы для самопроверки:

1. Почему из всего множества возможных булевых функций реализованы только вышеуказанные две?
2. Как можно реализовать иные логические функции?
3. Какое максимальное число входов может быть у логических элементов?
4. Почему данный вид логических элементов называется потенциальной логикой?
5. Какое значение напряжения питания используют данные логические элементы?
6. Какие значения могут иметь уровни напряжения логического нуля и логической единицы?

2.3. Задание к лабораторной работе

1. Изучите методические указания и последовательно выполните все пункты исследований.
2. Проверьте корректность работы логических элементов п. 2.4.1-2.4.3, сравнив выходные реакции моделей с заданными таблицами истинности. Проведите собственные исследования, увеличив число задействованных входов элементов до 4-х и скорректируйте для этих случаев таблицы истинности. Снимите скриншоты работающих схем.
3. Соберите собственный вариант преобразователя кодов п. 2.4.4, проанализируйте его поведение на последовательности сигналов, поступающих с генератора слов и постройте таблицу истинности. Продемонстрируйте работу схемы преподавателю. Снимите подтверждающие работу схемы скриншоты.
4. Оформите отчет в виде файла с названием «Фамилия, Лаб 1.doc».

2.4. Методические указания и порядок выполнения работы

2.4.1. Исследование элемента NOT (операция инверсии)

Рассмотрим самый простой элемент NOT. Элемент имеет всего один вход и один выход. В исходном состоянии, когда на входе присутствует 0, то на выходе присутствует 1.

Логические элементы изображаются на схемах в виде прямоугольника, внутри которого указывается тип логического элемента, так для элемента NOT ставится цифра 1. Прямоугольник обозначения логического элемента располагается всегда таким образом,

что входы элемента находятся слева, а выходы справа. Следует обратить внимание, что выход, имеющий в исходном состоянии логическую единицу, обозначается маленьким кружком.

Соберём простую электрическую схему с использованием элемента NOT. Для этого, сначала расставим на поле необходимые элементы, как показано на рисунке 2.1. Элемент "Voltage source - источник напряжения" находится на панели Sources, там же, где и заземление. Источник предназначен для получения сигнала логической 1, т. е. +5 В.

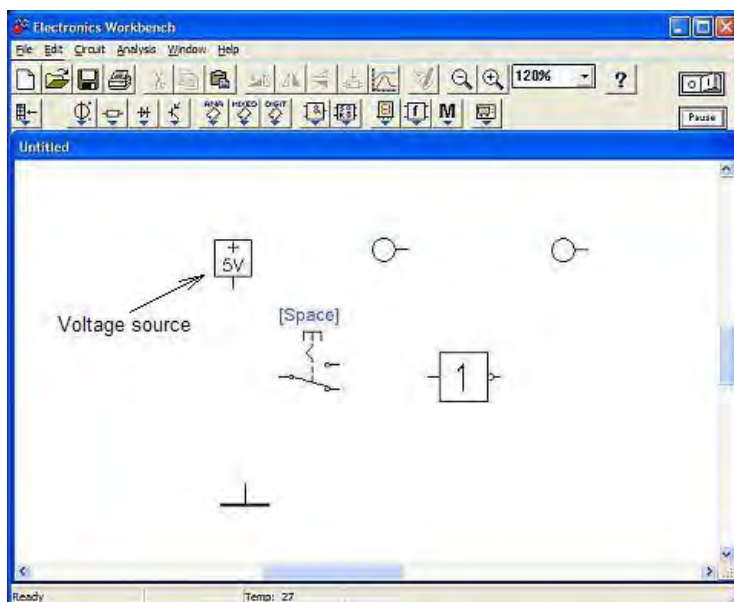




Рисунок 2.1. Рабочее окно программы

Некоторые элементы схемы, перед рисованием схемы, следует настроить.

Так, переключатель следует развернуть на 180 град. Для этого следует выделить переключатель и нажать кнопку  или, нажав правую кнопку мыши, выбрать в локальном меню опцию "Flip Horizontal". Индикаторные светодиоды следует повернуть выводами вниз. Для этого выделить светодиод и нажать несколько раз кнопку  или в локальном меню выбрать "Rotate". В итоге должна получиться такая картина:

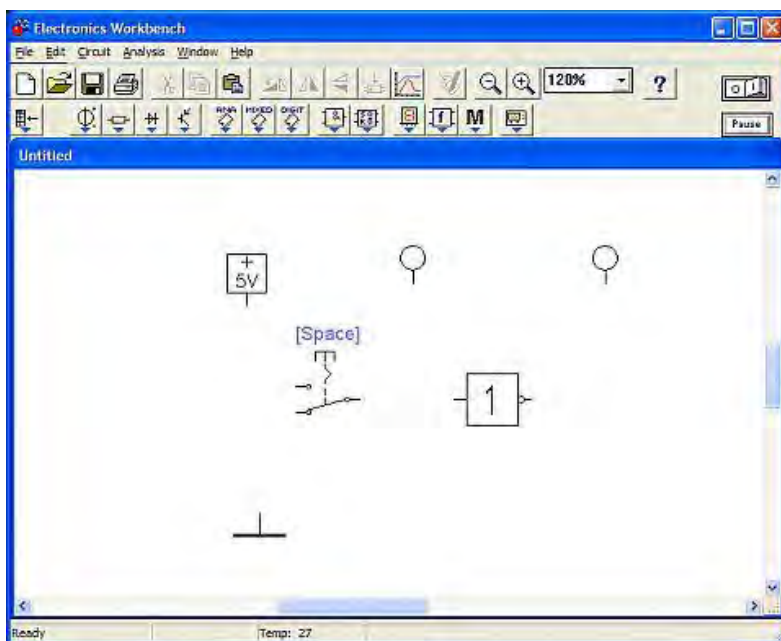


Рисунок 2.2 Элементы схемы

Далее, способом, описанным в предыдущем разделе, следует произвести соединения проводниками так, как показано на рисунке 2.3.

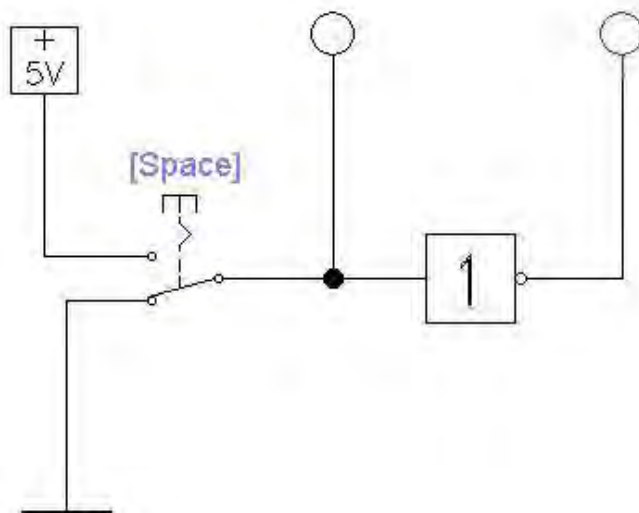



Рисунок 2.3 Подключение инвертора

Для проведения эксперимента следует включить выключатель питания электрической схемы  установив его в положение I.

Итак, мы видим, что индикатор, подключенный к выходу элемента, загорелся (стал красного цвета). На выходе элемента присутствует значение 1. Вход элемента заземлён через контакты переключателя, то есть на входе присутствует 0 (рисунок 2.4, А). Надпись "Space" на переключателе означает, что переключать его нужно нажатием клавиши "Пробел" на клавиатуре. Переключим переключатель. Вход логического элемента соединится с источником напряжения, то есть на входе элемента появится значение 1 (о чём свидетельствует загорание индикатора подключенного к входу), а сам логический элемент выполнит свою прямую обязанность, переключится в состояние 0 (рисунок 2.4,

Б). Такая работа логического элемента в логике называется **"Инверсия"**, а сам логический элемент - **Инвертор**.

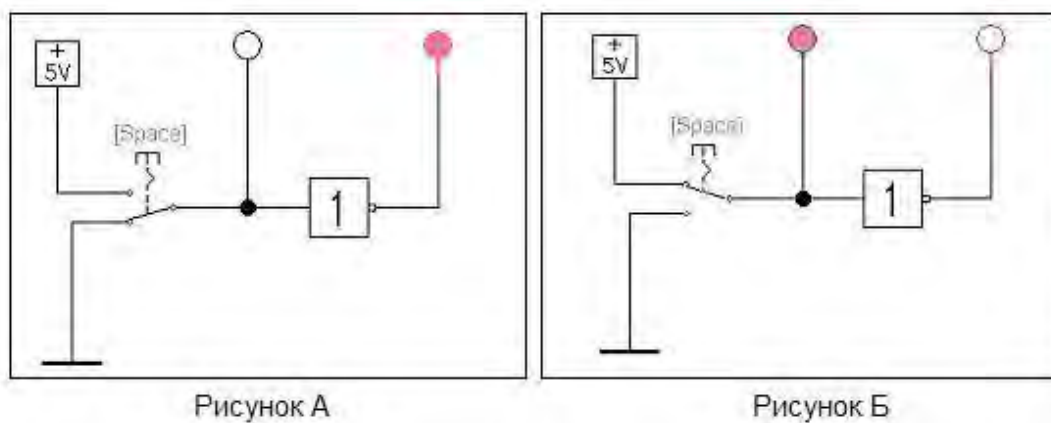


Рисунок 2.4 Работа инвертора

2.4.2. Исследование логического элемента AND (операция конъюнкции)

Логический элемент AND является многоходовым элементом, то есть у этого элемента должно быть число входов 2 и более. Название элемента описывает его функциональное назначение: если на входе 1 и на входе 2 и на входе N присутствует логическая единица, то на выходе элемента появляется логическая единица. Во всех остальных случаях на выходе элемента присутствует нуль. Логический элемент "И" на схеме обозначается значком &.

Соберём уже известным нам способом, схему, показанную на рисунке.

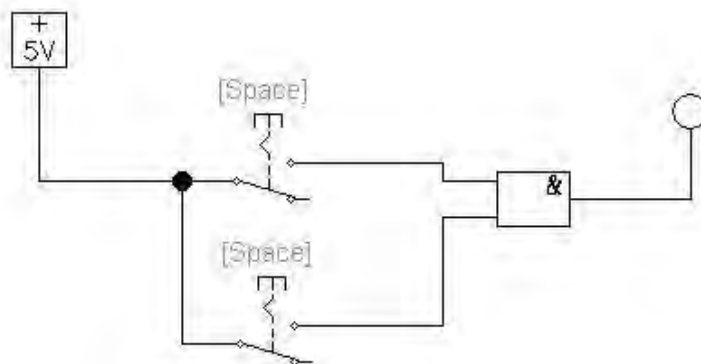


Рисунок 2.5 Схема «2-И»

Сразу начать тестирование полученной схемы нельзя, так как оба переключателя будут включаться одновременно при нажатии клавиши Пробел. Изменим назначение клавиш для переключателей, назначим им цифровые клавиши соответствующие входам элемента. Для этого следует выделить, например, верхний переключатель, нажать правую кнопку мыши и в локальном меню выбрать опцию "Component properties" (Свойства компонента). На экране появится окно настроек переключателя "Switch Properties".

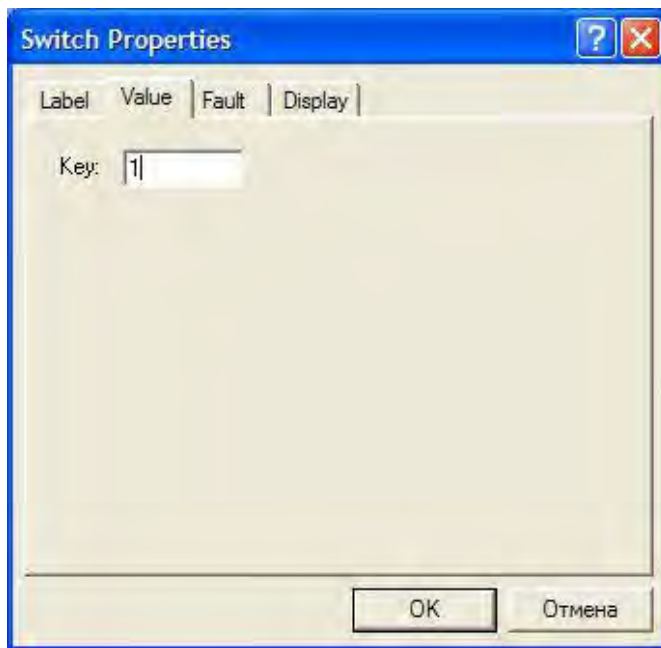


Рисунок 2.6 Окно настроек

В поле Key, вместо надписи Space, следует ввести 1 и нажать кнопку ОК. Теперь верхний переключатель будет переключаться при нажатии на клавиатуре клавиши 1. Аналогичные изменения следует произвести с нижним переключателем, но ввести значение 2. Теперь можно приступить к эксперименту - произвести все возможные переключения и убедиться, что элемент "И" срабатывает только тогда, когда оба переключателя включены. Переключать переключатели лучше согласно данным в представленной ниже таблице (знакомой как «Таблица истинности»).

Таблица 2.1

№ Перекл.	Входы / Переключатели		Выход
	1	2	
1	0	0	0
2	1	0	0
3	0	1	0
4	1	1	1

Добавим ещё один логический вход нашему элементу AND. Для этого следует выделить логический элемент, нажать правую кнопку мыши и выбрать опцию в локальном меню "Component properties". В окне свойств элемента выбрать закладку "Number of Inputs" (число входов) и установить флажок на значении 3. Нажать ОК.

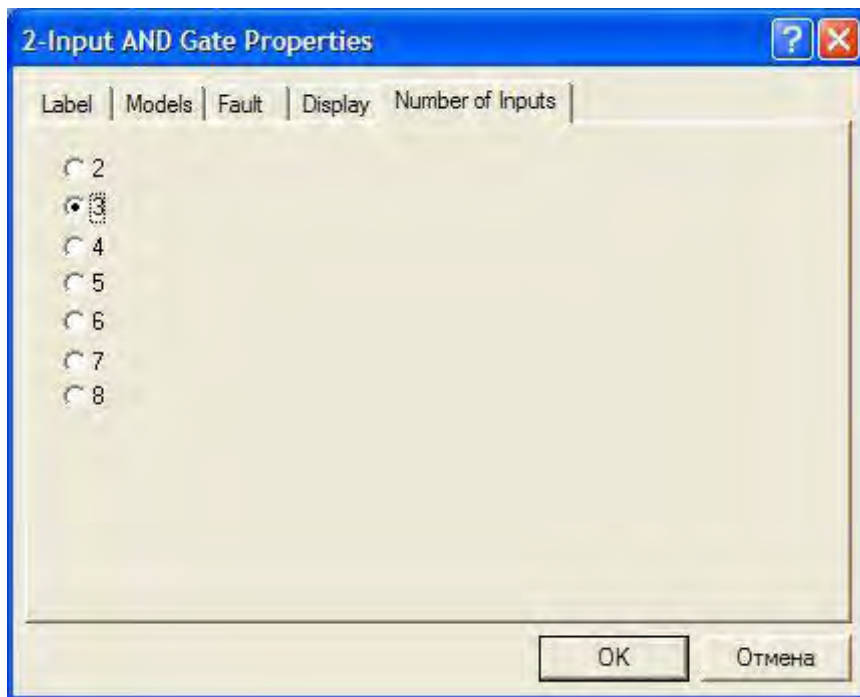


Рисунок 2.7 Выбор количества входов

На поле схемы следует добавить ещё один переключатель и назначить ему клавишу 3. Должна получиться схема, показанная на рисунке 2.8.

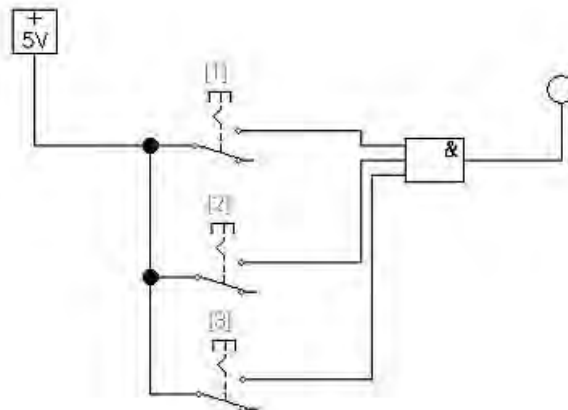


Рисунок 2.8 Трехвходовой логический элемент «И»

Для постановки эксперимента следует воспользоваться таблицей переключений.

Таблица 2.2

№ Перекл.	Входы / Переключатели			Выход
	1	2	3	
1	0	0	0	0
2	1	0	0	0
3	0	1	0	0
4	1	1	0	0
5	0	0	1	0
6	1	0	1	0
7	0	1	1	0
8	1	1	1	1

Очевидно, что число возможных переключений возросло с 4 до 8, но логика работы схемы не изменилась - логическая единица появляется на выходе только тогда, когда на всех входах присутствует логическая единица. Элемент "И" выполняет операцию, которая в логике называется **Конъюнкция**.

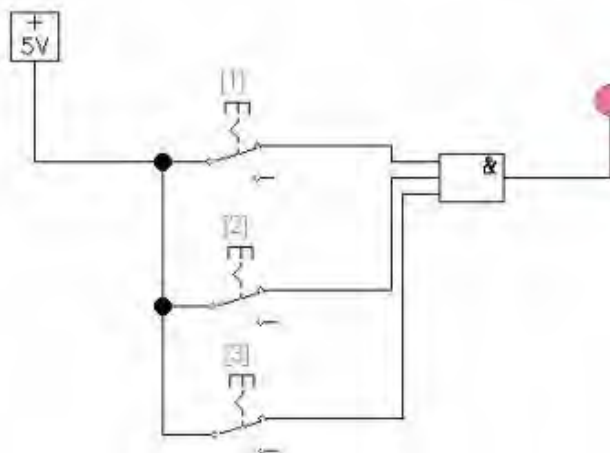


Рисунок 2.9 Работа трехвходового логического элемента

2.4.3. Исследование логического элемента OR (операция дизъюнкции)

Для исследования свойств логического элемента "ИЛИ", немного изменим предыдущую схему. Выделим и удалим элемент "И" и заменим его на элемент "ИЛИ". Элемент "ИЛИ" обозначается так же, как и элемент "НЕ" - 1, только на выходе не рисуется кружок. Элемент "ИЛИ", так же как и элемент "И", является многовходовым элементом.

На поле чертежа создадим элемент "ИЛИ" с тремя входами так, что бы получилась такая схема:

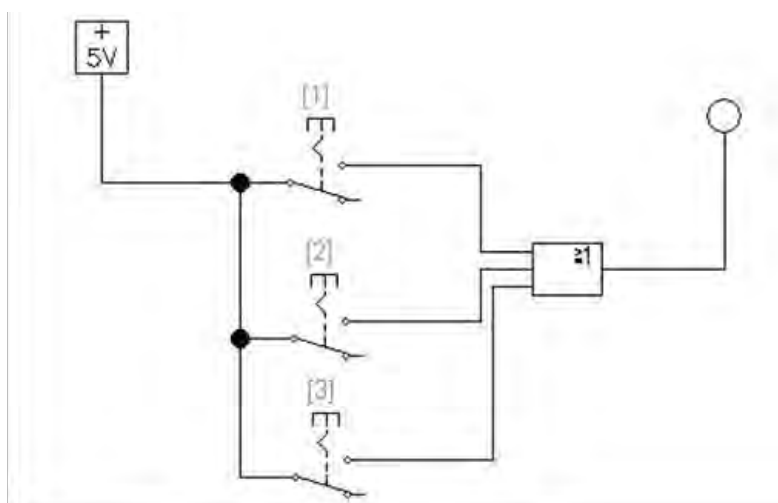


Рисунок 2.10 Трехвходовой логический элемент «ИЛИ»

При проведении эксперимента воспользуемся таблицей переключений, сходной с таблицей 2.2, и убедимся, что 1 на выходе элемента присутствует тогда, когда хотя бы на одном из входов присутствует 1. Такая логика работы элемента "ИЛИ" называется **Дизъюнкция**.

Все три перечисленных выше логических элемента AND, OR и NOT являются базовыми для построения сложных логических схем, таких как сумматоры, шифраторы и дешифраторы, мультиплексоры и многие другие.

2.4.4. Исследование сложной комбинационной схемы (преобразователь кодов)

Изученные логические элементы могут использоваться в качестве компонентов для создания более сложных логических схем, к которым относятся преобразователи кодов. Это – обширный класс устройств, характеризующийся множеством выходов Y , сигналы на которых определяются значениями сигналов, поданных на множество входов X .

Фрагмент схемы подобного преобразователя показан ниже (рисунок 2.11).

Обратите внимание на состав и взаимосвязи элементов.

Особо отметим, что на каждый вход логических элементов поступает сигнал строго с одного выхода предыдущего каскада!

На входы схемы подаётся множество сигналов $\{X_i\}$ (в данном задании - четыре).

Схема состоит из трёх каскадов.

Первый каскад – инверторы, включённые таким образом, что далее подаются парафазные сигналы: прямой X и инверсный \bar{X} .

Второй каскад – логические элементы «И-НЕ» с несколькими входами (в данном задании от 2 до 4), на которые поступают сигналы X и \bar{X} , причём комбинации сигналов у каждого элемента различны.

Третий каскад - логические элементы «И-НЕ» с несколькими входами (в данном задании от 2 до 4), на которые поступают сигналы со второго каскада.

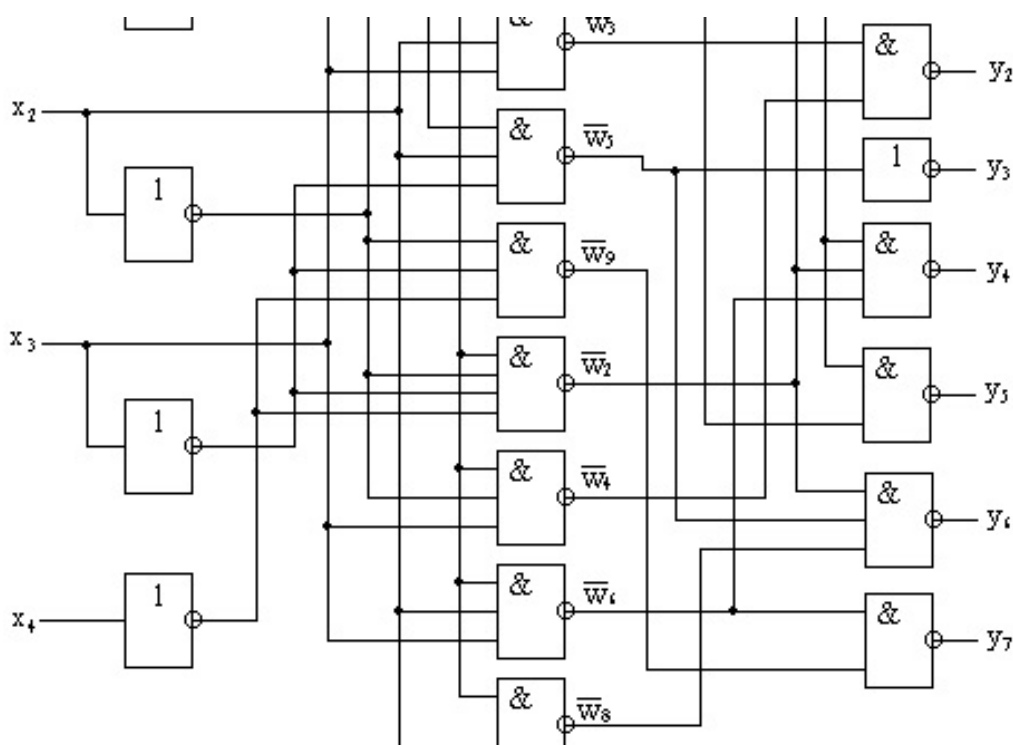


Рисунок 2.11 Фрагмент сложной комбинационной схемы

Для проведения эксперимента самостоятельно разработайте на основе примера собственную схему преобразователя, имеющего 4 входа X и 5-7 выходов Y (конкретные значения уточните у преподавателя).

Для формирования входных сигналов следует применить переключатели, включенные по схеме, аналогичной представленной на рисунке 2.10, либо воспользоваться генератором слов, работающим в пошаговом режиме.

Индикация состояний на выходах осуществляется с помощью индикаторных лампочек.

Постройте для своей схемы Таблицу истинности и ДНФ для каждого из выходов.

2.5. Требования к отчету и защите

В отчете указываются название, цель работы. Описание выполненных лабораторных заданий с результатами в виде скриншотов, сохраненных результатов моделирования в файле формата соответствующей программы моделирования и выводами по каждому заданию.

На защите проверяются приобретенные знания теоретического и практического материала с демонстрацией результатов моделирования и по ответам на контрольные вопросы для самопроверки.

Отчёт должен содержать:

1. Формулировку цели работы.
2. Краткие сведения о свойствах и отличительных чертах различных классов комбинационных элементов.
3. Изложение хода исследований комбинационных элементов (согласно п. 2.4.1 - 2.4.4 работы) в виде скриншотов отдельных шагов работы, полученных таблиц истинности и логических выражений (ДНФ).

Отчет предоставляется в электронном виде через ЭИОС

3. ЛАБОРАТОРНАЯ РАБОТА № 2. ЭЛЕМЕНТЫ ПРОЦЕССОРОВ ЭВМ. ИССЛЕДОВАНИЕ СУММАТОРОВ И АЛУ

3.1. Общие сведения

Цель:

изучение номенклатуры и принципов проектирования комбинационных элементов и схем компьютера

Материалы, оборудование, программное обеспечение:

IBM PC-совместимый персональный компьютер, программа MicroCap 9.0 5.0 Evaluation version, или иная, также являющаяся свободно распространяемой демоверсией профессиональной программы машинного моделирования электронных схем.

Условия допуска к выполнению:

Предварительное знакомство с УМП, отсутствие существенных задолженностей по лабораторному практикуму.

Критерии положительной оценки:

Выполнение типового задания, оформление отчета по работе, ответы на вопросы для самопроверки.

Планируемое время выполнения:

Аудиторное время выполнения (под руководством преподавателя): 4 ч.

Время самостоятельной подготовки: 2 ч.

3.2. Теоретическое введение

Значительная часть электронных узлов, выделяемых в составе структур компьютеров, относятся к комбинационным схемам. Их выходной сигнал определяется только текущей входной комбинацией, т. е. не зависит от истории выполнения ими команд, а закон функционирования описывается исключительно таблицей истинности.

Помимо упомянутых ранее «преобразователей кодов», к типовым комбинационным относят такие функциональные элементы, как шифраторы и дешифраторы, мультиплексоры и демultipлексоры, а также сумматоры и арифметико-логические устройства. К комбинационным схемам можно отнести и целые структуры, выполненные на основе программируемых логических матриц.

3.2.1. Порядок проектирования комбинационных схем

При проектировании схем, выполняющих ту или иную логическую функцию, необходимо соблюсти следующие требования:

- обеспечить реализацию соответствующей функции на заданном множестве входных сигналов;
- обеспечить минимизацию аппаратных затрат на реализацию этих схем;
- сократить номенклатуру используемых логических элементов.

Последнее требование реализуется путем выбора соответствующей системы элементов. В настоящее время основные серии интегральных логических микросхем (ИМС) включают в себя элементы, составляющие некоторый функционально полный логический базис, а также дополнительные элементы, реализующие часто встречающиеся логические функции. В качестве функционально полных базисов используются, как правило, одноэлементные базисы "И-НЕ" либо "ИЛИ-НЕ".

Рассмотрим пример проектирования комбинационной схемы на примере дешифраторов, условные графические обозначения которых приведены ниже.

Дешифраторами (DC) называются устройства, обычно используемые для преобразования двоичных чисел в комбинации кода «1-из-N». Входы дешифратора предназначаются для подачи двоичных чисел и обозначаются их весами, выходы последовательно нумеруются десятичными числами от 0 до N-1.

При подаче на входы двоичного числа появляется сигнал на определенном выходе, номер которого соответствует входному числу.

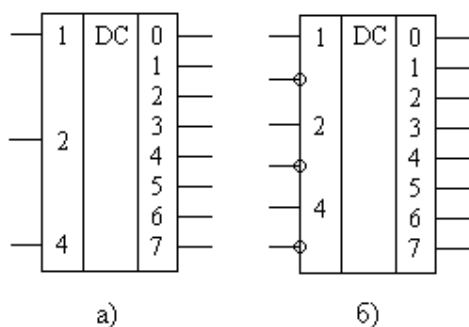


Рисунок 3.1 Дешифраторы

Этап 1. Представление функции, выполняемой проектируемой схемой, в каноническом виде, т. е. в виде таблицы истинности или одной из совершенных нормальных форм записи.

Обычно на этом этапе словесное описание функции заменяется таблицей истинности. Так как проектируется схема с несколькими выходами, то необходимо представить в таблице истинности каждый ее выход (таблица 3.1)..

Обозначим в таблице входы схемы литерой X, а выходы – Y.

Таблица 3.1

X1	X2	X4	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y6
0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
1	1	0	0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Этап 2. Минимизация логической функции.

Целью минимизации многовыходных функций является синтез схемы устройства, имеющего минимальный общий состав оборудования, то есть следует проводить минимизацию одной функции с учетом возможного использования части полученного оборудования для минимизации другой функции.

Минимизацию можно проводить различными известными из курса дискретной математики способами: методом Квайна, его модификацией – методом Квайна – Мак-Класки, методом карт Карно или диаграмм Вейча.

Очевидно, минимизацию следует выполнять лишь при исходной избыточности в функции, например – наличии дублируемых выходных сигналов, наличии неопределенного их значения, неполного входного набора и т. п., т. е. для рассматриваемого примера данный этап излишен.

Этап 3. Представление (минимизированной) логической функции в минимальной дизъюнктивной нормальной форме (ДНФ).

Для рассматриваемого примера система уравнений, описывающих функции всех выходных сигналов, будет иметь вид:

$$Y_0 = \overline{X_1} \& \overline{X_2} \& \overline{X_4}$$

$$Y_1 = X_1 \& \overline{X_2} \& \overline{X_4}$$

.....

$$Y_7 = X_1 \& X_2 \& X_4$$

Этап 4. Перевод функции в базис, в котором будет строиться схема. Реализация схемы.

Поскольку в большинстве серий ИМС качестве функционально полных базисов используются, как правило, одноэлементные базисы "И-НЕ", выходные сигналы Y0- Y7 целесообразно представить в негативной логике, а для формирования входных парафазных сигналов X1 – X4 в необходимых случаях воспользоваться инверторами.

Фрагмент схемы подобного дешифратора представлен ниже.

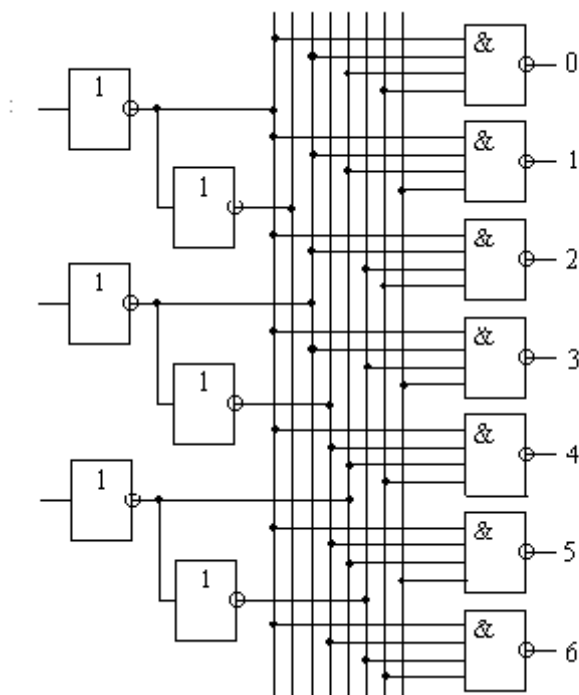


Рисунок 3.2 Фрагмент схемы дешифратора

Этап 5. Моделирование схемы.

Моделирование проводится в целях практического подтверждения правильности использованных проектных решений и выполняется способом, рассмотренным в материалах лабораторной работы № 1.

Далее в разделе «Методические указания и порядок выполнения работы» описываются иные типовые комбинационные схемы, проектирование которых проводится на основе вышеизложенной методики.

Литература:

Кардашев, Г.А. Цифровая электроника на персональном компьютере. Electronics Workbench и Micro-Cap / Г. А. Кардашев. - Москва : Горячая линия-Телеком, 2003. Стр. 115-140

Суханова, Н.В. Основы электроники и цифровой схемотехники [Электронный ресурс] : учеб. пособие / Н.В. Суханова ; Министерство образования и науки РФ, Воронежский государственный университет инженерных технологий. - Воронеж : Воронежский государственный университет инженерных технологий, 2017. - 97 с. (ЭБС «Университетская библиотека онлайн») Стр. 90-93.

Контрольные вопросы для самопроверки:

1. Почему некоторые выводы комбинационных схем могут иметь обозначения А, D, E? От чего зависит выбор обозначения?
2. В чём состоят отличия шифраторов и дешифраторов?
3. Чем различаются мультиплексор и демultipлексор?
4. Какие коды используются шифраторы?
5. Сколько входов и выходов должен иметь одноразрядный полный сумматор?
6. В чём состоят различия сумматора и полусумматора?

3.3. Задание к лабораторной работе

1. Изучите теоретическую часть методических указаний и повторите все описанные этапы проектирования.
2. Выполните этапы 1 – 5 изложенной методики на примере шифратора / мультиплексора / демультиплексора / сумматора (по согласованию с преподавателем).
3. Проверьте правильность выполнения задания, сравнив выходные реакции моделей с характерными для данных классов комбинационных схем таблицами истинности. Снимите скриншоты работы схемы.
4. Оформите отчет в виде файла с названием «Фамилия, Лаб 2.doc».

3.4. Методические указания и порядок выполнения работы

3.4.1. Исследование шифраторов

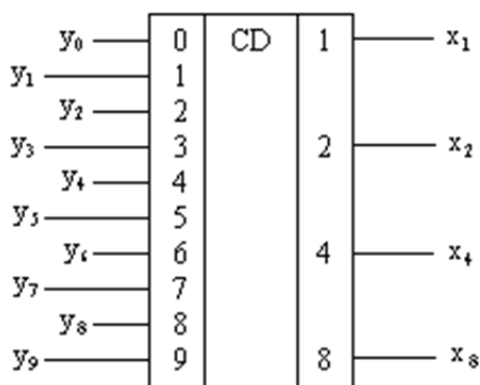


Рисунок 3.3 Шифратор

Шифратор, (называемый также кодером - CD) - устройство, обычно осуществляющее «обратное» дешифратору, например - преобразование десятичных чисел, представленных в коде «1-из-N», в двоичную систему счисления.

Пусть в шифраторе имеется m входов, последовательно пронумерованных десятичными числами (0, 1, 2, 3, ..., $m - 1$), и n выходов. Тогда подача сигнала на один из входов приводит к появлению на выходах n -разрядного двоичного числа, соответствующего номеру активированного входа.

Выберите из библиотеки элементов системы моделирования ИМС, реализующую функции шифратора (дешифратора) и проведите исследования, подавая на входы Y полный набор комбинаций, фиксируя при этом выходные сигналы X .

Постройте таблицу истинности и опишите её с помощью ДНФ.

Объясните, почему у шифратора в данном случае используется «обратная» система обозначений входов и выходов.

3.4.2. Исследование мультиплексора

Мультиплексор (MUX, MS) — устройство, имеющее несколько входов данных D (*Data*), один или более управляющих входов и один выход данных. Мультиплексор позволяет передавать сигнал с выбранного входа на выход; при этом выбор осуществляется подачей соответствующей комбинации управляющих сигналов, поступающих на селекторные входы A (*Address*).

Дополнительные входы E (*enable*) могут использоваться для расширения функций мультиплексора, в том числе – для создания многокаскадных схем.

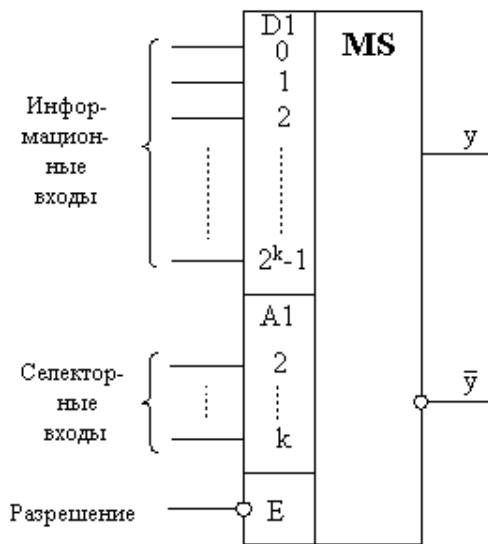


Рисунок 3.4 Мультиплексор

Структурно мультиплексор включает в свой состав коммутатор, обеспечивающий подключение одного из нескольких входов X к выходу Y устройства, а также управляющую схему, выбирающую конкретный вход. Элементы, реализующие функции коммутатора и управляющей схемы, могут объединяться на этапе минимизации схемы устройства.

Обратите внимание: обозначения А, D, Е имеют отсылку на функциональное предназначение выводов. В иных случаях могут быть использованы традиционные обозначения для входов и выходов: X и Y соответственно.

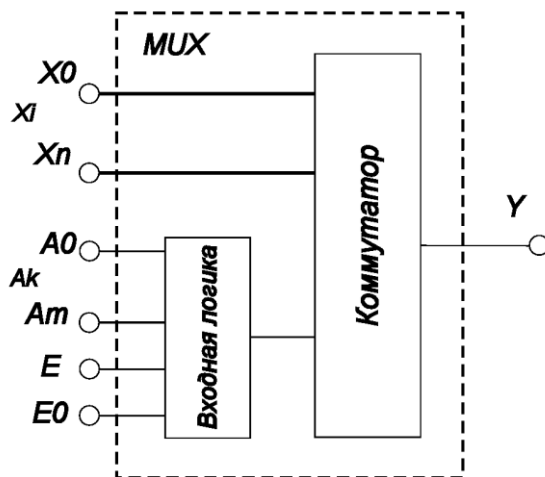


Рисунок 3.5 Схема мультиплексора

3.4.3. Исследование демультиплексора

Демультиплексор (DMX, DMS) — это логическое устройство, предназначенное для переключения сигнала с единственного информационного входа X на один из информационных выходов Y . Выбор выхода определяется соответствующим кодом, поступающим на входы, аналогичные используемым мультиплексором. Таким образом, демультиплексор в функциональном отношении противоположен мультиплексору.

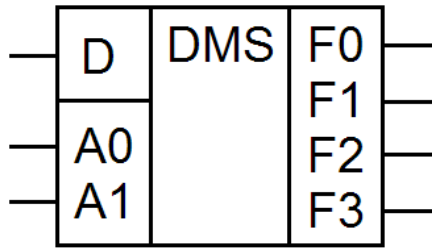


Рисунок 3.6 Демультплексор

3.4.4. Исследование сумматоров и АЛУ

Самым сложным комбинационным узлом структуры компьютера является арифметико-логическое устройство. Оно включает в свой состав элементы, обеспечивающие выполнение типового набора арифметических и логических команд над многоразрядными двоичными данными, поступающими из устройств памяти, входных устройств и иных компонентов ЭВМ.

Наиболее известным элементом АЛУ является сумматор – устройство, преобразующее поступающие по входным линиям двоичные числа в сумму этих чисел. В зависимости от специфики выполнения арифметической операции суммирования они подразделяются на четвертьсумматоры, полусумматоры и полные сумматоры.

В состав процессоров обычно входят многоразрядные параллельные полные сумматоры, ориентированные на позиционные системы счисления. Подобные устройства обладают двумя группами входов, на которые подаются одноимённые разряды двух складываемых операндов (на рисунке 3.7 обозначены как входы А и В), и входом переноса CR из предыдущего (более младшего) разряда. Выходы, обозначенные как S, предназначены для вывода результатов суммирования соответствующих разрядов, выходы CR – для вывода сигналов переполнения или переноса в следующе старшие разряды.

Далее представлены фрагмент функциональной схемы (рисунок 3.7) и условное графическое обозначение типичного многоразрядного полного сумматора (рисунок 3.8).

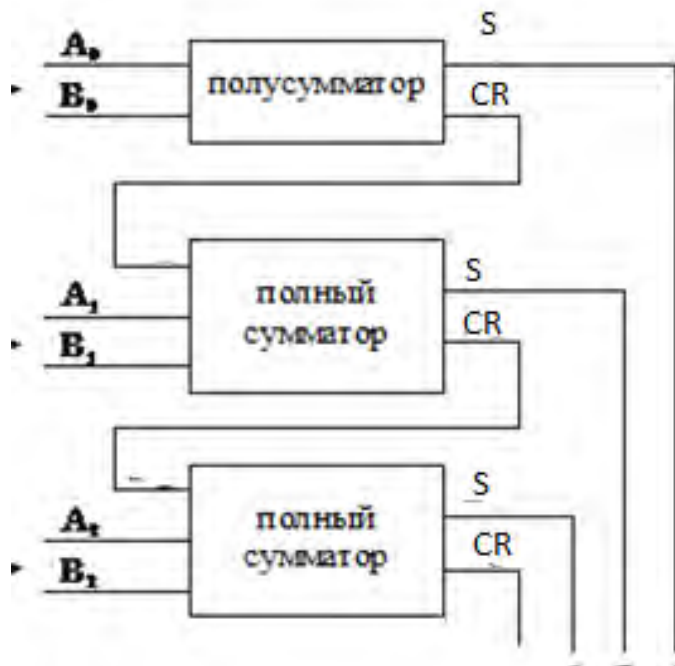


Рисунок 3.7

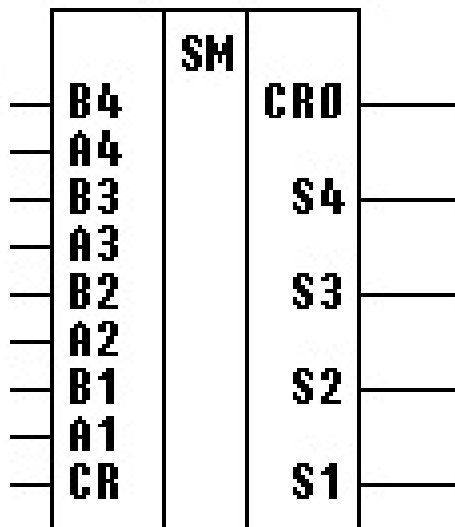


Рисунок 3.8

3.5. Требования к отчету и защите

В отчете указываются название, цель работы. Описание выполненных лабораторных заданий с результатами в виде скриншотов, сохраненных результатов моделирования в файле формата соответствующей программы моделирования и выводами по каждому заданию.

На защите проверяются приобретенные знания теоретического и практического материала с демонстрацией результатов моделирования и по ответам на контрольные вопросы для самопроверки.

Отчёт должен содержать:

1. Формулировку цели работы.
2. Детальное описание назначения и особенностей работы моделируемого элемента
3. Таблицу истинности для элемента с 9 - 11 информационными входами (или выходами, или входами + выходами – по согласованию с преподавателем).

*В качестве основы используйте Таблицу истинности «сложной комбинационной схемы», синтезированной в ходе выполнения предыдущей работы.

4. Принципиальную электрическую схему исходного устройства в базисе «И-НЕ».
5. Выводы о возможности минимизации схемы устройства, результаты проведенной минимизации.
6. Контрольный пример в виде результатов моделирования и таблицы истинности минимизированной схемы.

Отчет предоставляется в электронном виде через ЭИОС

4. ЛАБОРАТОРНАЯ РАБОТА №3. ЭЛЕМЕНТЫ ПРОЦЕССОРОВ ЭВМ. ИССЛЕДОВАНИЕ ПОСЛЕДОВАТЕЛЬНОСТНЫХ ЭЛЕМЕНТОВ

4.1. Общие сведения

Цель:

исследование изучение номенклатуры и структур типовых триггеров

Материалы, оборудование, программное обеспечение:

IBM PC-совместимый персональный компьютер, программа MicroCap 9.0 5.0 Evaluation version, или иная, также являющаяся свободно распространяемой демоверсией профессиональной программы машинного моделирования электронных схем.

Условия допуска к выполнению:

Предварительное знакомство с УМП, отсутствие существенных текущих задолженностей по лабораторному практикуму.

Критерии положительной оценки:

Выполнение типового задания, оформление отчета по работе, ответы на вопросы для самопроверки.

Планируемое время выполнения:

Аудиторное время выполнения (под руководством преподавателя): 4 ч.

Время самостоятельной подготовки: 2 ч.

4.2. Теоретическое введение

В состав структур большинства цифровых узлов и схем входят однобитные элементы памяти – триггеры. В отличие от комбинационных схем (КС) значения выходных сигналов последовательностных схем (ПС) в данный момент времени зависят не только от значений входных сигналов в этот же момент времени, но и от их предыдущих значений. Из этого следует, что ПС реализует функциональную связь уже не между отдельными значениями входных и выходных сигналов, а между их последовательностями. Поэтому, в отличие от КС, работу ПС следует рассматривать во времени.

Для того, чтобы значения выходных сигналов зависели от предыдущих значений входных, ПС должны обладать памятью, в которой сохраняется информация о предыдущих входных воздействиях. Эта информация используется в ПС в виде совокупности сигналов, вырабатываемых памятью.

Особое значение при изучении последовательностных схем имеют элементы памяти – триггеры.

Триггеры имеют два устойчивых состояния. Эти состояния определяются по логическим уровням на выходах триггера. Триггер снабжается двумя выходами: прямым Q и инверсным \bar{Q} . Состояние триггера определяет логический уровень на выходе Q. Говорят, что триггер находится в состоянии логической единицы, если на выходе Q уровень напряжения, соответствующий логической единице.

Триггеры могут иметь входы различного типа:

- R (от англ. RESET) – отдельный вход установки в состояние 0;
- S (от англ. SET) – отдельный вход установки в состояние 1;
- К – вход установки универсального триггера в состояние 0;

- J – вход установки универсального триггера в состояние 1;
- V – вход разрешения;
- C – синхронизирующий вход;
- D – информационный вход;
- T – счетный вход
- и некоторые другие.

Номенклатура триггеров может быть достаточно широкой, так как помимо собственно хранения данных, они выполняют ряд вспомогательных операций, связанных с определением момента времени или события, в которое производится выборка и запоминание данных.

Обычно название триггера дают по имеющимся у него входам: RS-триггер, JK-триггер, D-триггер и др., а по способу записи информации подразделяются на асинхронные и синхронные. Требования унификации привели к появлению многофункциональных элементов, реализующих набор типичных функций и имеющих соответствующие функциям названия.

Основу триггеров составляют простейшие запоминающие ячейки, представляющие собой симметричную структуру из двух логических элементов ИЛИ-НЕ либо И-НЕ, охваченных перекрёстной обратной связью (рисунок 4.1):

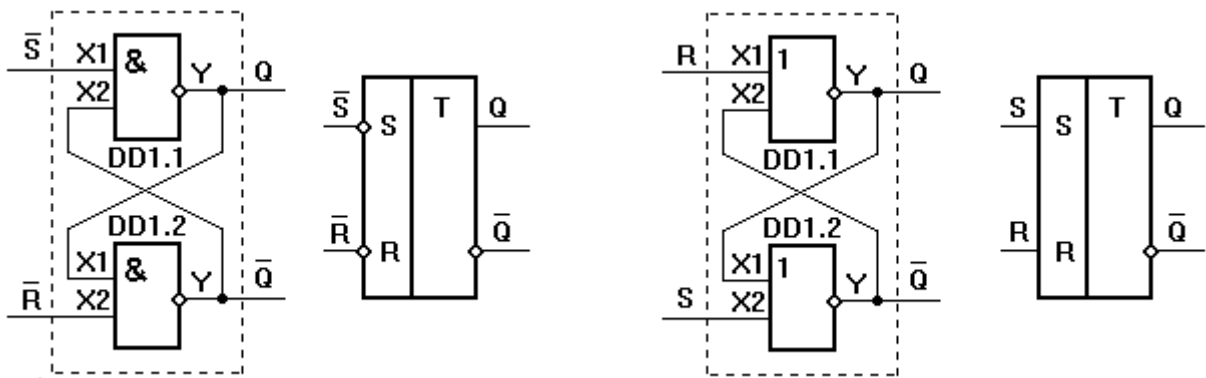


Рисунок 4.1 RS-триггеры на основе элементов И-НЕ и ИЛИ-НЕ. Структуры и условные графические обозначения.

Закон функционирования ячейки описывается следующими правилами.

Независимо от того, какую функцию выполняют логические элементы И-НЕ либо ИЛИ-НЕ, ячейки могут находиться в двух устойчивых состояниях: 1 и 0. Состоянию 1 соответствует единичный сигнал на выходе Q, состоянию 0 - единичный сигнал на выходе.

Вход, по которому ячейка устанавливается в состояние 1, обозначается буквой S, а в состоянии 0 – буквой R.

Когда на обоих информационных кодах существуют логические нули ($S=R=0$), сигналы на выходе могут иметь одно из двух сочетаний: $Q=1, \bar{Q}=0$ либо $Q=0, \bar{Q}=1$, так как каждый логический элемент ИЛИ-НЕ инвертирует входные сигналы, а переключающим сигналом служит единица. Допустим, что $Q=1$. Этот сигнал, воздействуя на вход нижнего элемента, создаёт на его выходе $\bar{Q}=0$. В свою очередь, на входах верхнего элемента два нулевых сигнала – со входа R и с выхода, обеспечивают $Q=1$. Состояние это устойчивое.

Если на один из входов подать единичный сигнал, сохраняя нулевой на другом, триггер примет состояние, которое однозначно определяется входной информацией. При входных сигналах $S=1, R=0$ триггер принимает единичное состояние $Q=1, \bar{Q}=0$, а при

$S=0, R=1$ – нулевое: $Q=0, \bar{Q}=1$. При появлении управляющего сигнала на одном из входов происходит либо опрокидывание триггера, либо подтверждение существующего состояния, если оно совпадает с требуемым.

Если одновременно подать переключающие сигналы на оба входа ($S=R=1$), на обоих выходах появятся логические нули ($Q=\bar{Q}=0$) и устройство утратит свойства триггера. Поэтому комбинацию $S=R=1$ называют неопределённой.

Переход от неопределённой комбинации к нейтральной ($S=R=0$) называют запрещённой комбинацией, так как состояние выходов при этом восстанавливается, но с равной вероятностью оно может стать единичным, так и нулевым, т.е. ведёт к непредсказуемому поведению триггера.

Триггер, который переключается сигналами логической единицы, т.е. на логических элементах ИЛИ-НЕ, называют триггером с прямыми входами (RS-триггер). Триггер, который переключается сигналами логического нуля, т.е. на логических элементах И-НЕ, называют триггером с инверсными входами. Для такого триггера неопределённая комбинация (н/о): $S=R=0$.

RS-триггеры служат базовыми для создания на их основе иных типов триггеров, функционирующих в соответствии с собственными законами.

Для представления законов функционирования ПС используются специальные языки, в том числе – формулы; таблицы, в которых записываются состояния на входах и выходах триггера в различные моменты времени; временные диаграммы и автоматные модели.

Например, модель триггера может быть представлена следующей системой функций, описывающих выходные сигналы y в последующий момент времени как функцию от входных и выходных сигналов в текущий момент:

$$y_1(t+1) = f(x_1(t), y_2(t)) = \overline{x_1(t) \& y_2(t)},$$

$$y_2(t+1) = f(x_2(t), y_1(t)) = \overline{x_2(t) \& y_1(t)}$$

На временной диаграмме (рисунок 4.2) показано, как меняются сигналы на выходах триггера при изменении сигналов на его входах.

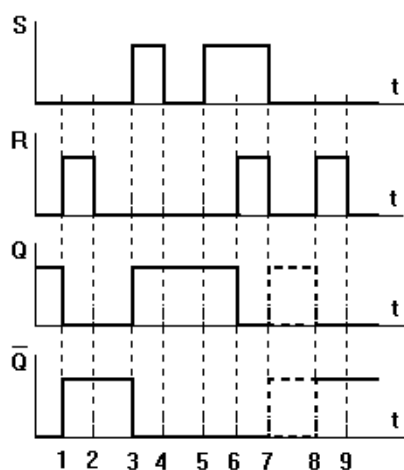


Рисунок 4.2. Временная диаграмма работы RS-триггера с прямыми входами

Автоматная модель описывает весь процесс смены состояний выходов триггера как в статике, так и в динамике (например, при переключении между его устойчивыми состояниями происходит переход через состояние 11).

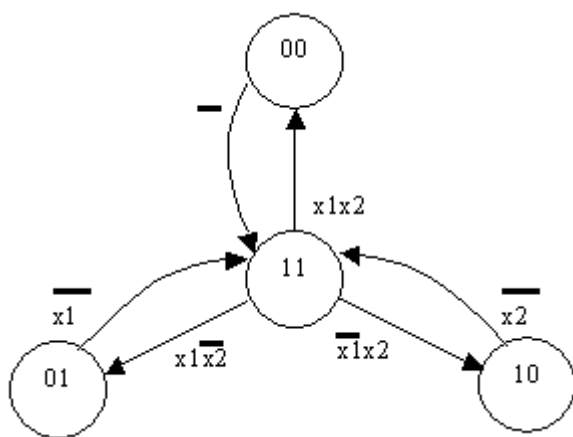


Рисунок 4.3. Автоматная модель RS-триггера с прямыми входами

Литература:

Суханова, Н.В. Основы электроники и цифровой схемотехники [Электронный ресурс] : учеб. пособие / Н.В. Суханова ; Министерство образования и науки РФ, Воронежский государственный университет инженерных технологий. - Воронеж : Воронежский государственный университет инженерных технологий, 2017. - 97 с. (ЭБС «Университетская библиотека онлайн») Стр. 73-79

Кардашев, Г.А. Цифровая электроника на персональном компьютере. Electronics Workbench и Micro-Cap / Г. А. Кардашев. - Москва : Горячая линия-Телеком, 2003. Стр. 168-208

Контрольные вопросы для самопроверки:

1. Можно ли описать логику работы триггера с помощью таблицы истинности?
2. Какие языковые средства используются при описании работы последовательностных средств?
3. В чём состоит основное отличие последовательностных и комбинационных схем?
4. В чём отличия автоматов Мили и Мура?
5. Современные триггеры являются автоматами Мура или Мили? Почему?

4.3. Задание к лабораторной работе

1. Изучите теоретическую часть методических указаний и выполните задания 4.4.1 - 4.4.3.
2. Оформите отчет в виде файла с названием «*Фамилия*, Лаб 4.doc».

4.4. Методические указания и порядок выполнения работы

4.4.1. Исследование асинхронного RS-триггера

В асинхронных триггерах состояние на выходе изменяется сразу же после изменения сигнала на информационных входах.

Модель, имитирующая работу асинхронного RS-триггера, аналогичному описанному выше, показана на рисунке 4.4. Триггер образован двумя комбинационными элементами И-НЕ, соединенными таким образом, что возникает положительная обратная

связь, благодаря чему в устойчивом состоянии выход одного элемента закрыт, а другого - открыт.

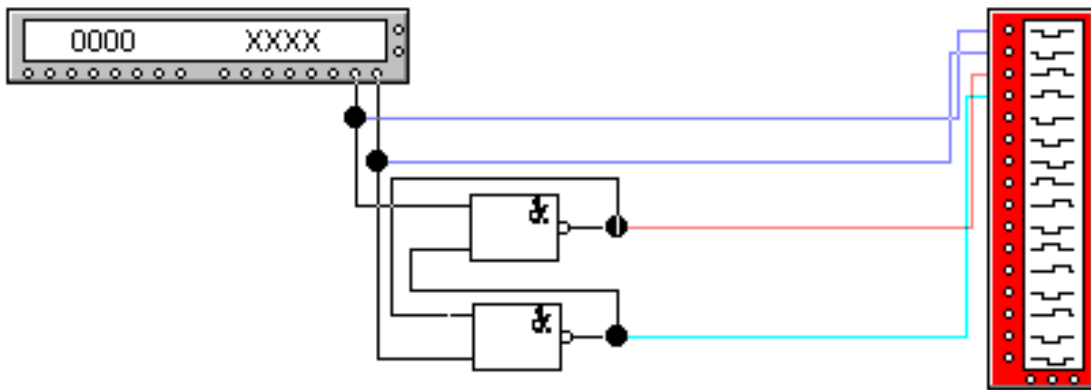


Рисунок 4.4. Схема асинхронного RS-триггера, собранная в EWB

Временная диаграмма, иллюстрирующая работу асинхронного RS-триггера, показана на рисунке 4.5.

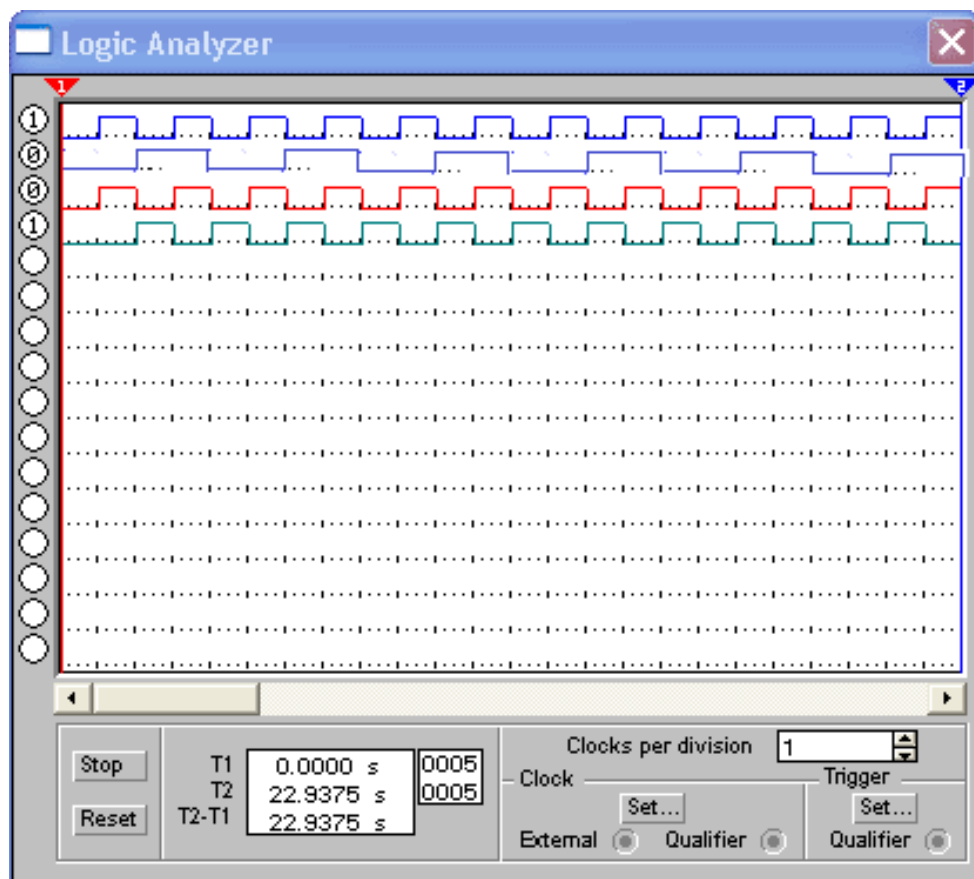


Рисунок 4.5. Временная диаграмма, иллюстрирующая работу асинхронного RS-триггера. (Обратите внимание, что выходные сигналы, обозначенные красным и бирюзовым цветом, приведены в качестве примера и могут не соответствовать правильным!).

Задание 4.4.1.

- Соберите схему согласно рисунку 4.4.
- Подайте на входы сигналы, аналогичные показанным на рисунке 4.5, и снимите диаграммы выходных сигналов (сделайте скриншоты).
- Поменяйте местами входные сигналы и вновь снимите выходные диаграммы.
- Сравните и объясните полученные результаты (*перестановка входов привела к... потому что...*).
- Сравните результаты с диаграммой, представленной на рисунке 4.2 (*Диаграмма на рисунке 4.2 демонстрирует... так как...*).
- Предложите вид диаграммы входных сигналов, позволяющей выявить все характерные особенности поведения RS-триггера (*Для демонстрации всех особенностей поведения триггера необходимо подавать на его входы сигналы в следующей последовательности... На первом этапе определяется... На втором выявляется... На третьем устанавливается... Четвертый показывает...*). Подсказка: «А не правда ли, что для проверки функционирования триггера следует последовательно проверить все переходы между состояниями, как показано на рисунке 4.3.?»

4.4.2. Исследование синхронного RS-триггера со статическим синхрвходом

В синхронных триггерах для передачи сигнала с информационных входов на выход требуется специальный синхронизирующий импульс.

Синхронные триггеры подразделяются на триггеры со статическим синхрвходом (управлением) и триггеры с динамическим синхрвходом (управлением). На рисунках 4.6 а и в приведены варианты реализации принципиальных электрических схем в различных базисах, а на рисунке 4.6. б - условное графическое обозначение синхронного триггера с прямыми RS-входами и статическим синхрвходом.

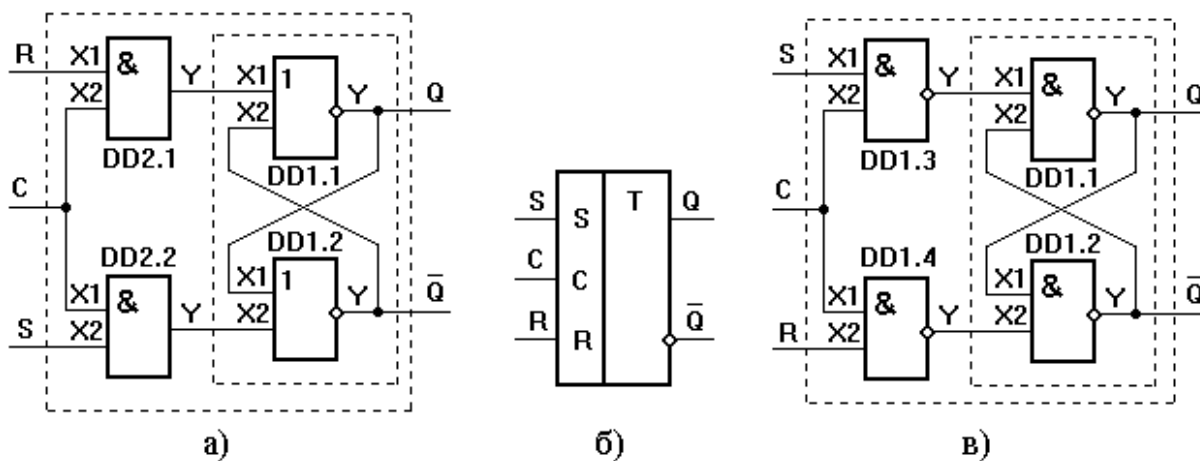


Рисунок 4.6. Синхронный RS-триггер со статическими входами

Из анализа работы логических элементов видно, что при $C=0$ на входах R, S асинхронного триггера на элементах DD1.1 и DD1.2 действуют сигналы логического нуля (логический ноль является пассивным логическим уровнем для логических элементов 2ИЛИ-НЕ), поэтому при любых комбинациях сигналов на входах R, S синхронного RS-триггера состояние триггера не меняется. При $C=1$ рассмотренный синхронный RS-триггер работает точно так же, как рассмотренный ранее асинхронный RS-триггер с прямыми входами.

Данный синхронный RS-триггер относится к триггерам со статическим синхровходом, так описанные условия работы остаются неизменными все время, пока поддерживается текущее значение сигнала на входе С.

Задание 4.4.2.

- Проанализируйте взаимозависимость сигналов, подаваемых на входы S, R и С триггера. В какой последовательности и с какой частотой должны изменяться данные сигналы в процессе исследования поведения триггера? (Сигнал с первого выхода генератора слов, имеющий наибольшую частоту, необходимо подавать на вход... так как... Сигнал со следующего выхода подадим ...)
- По рисунку 4.6 на основе рисунка 4.4 самостоятельно соберите схему для исследования работы триггера.
- Получите с помощью EWB временную диаграмму, наглядно демонстрирующую особенности функционирования данного триггера. Сделайте необходимые скриншоты.

1.

4.4.3. Исследование синхронного RS-триггера с динамическим синхровходом

В триггерах с динамическим управлением передача сигналов с информационных входов на выходы осуществляется либо по фронту, либо по спаду (срезу) синхронизирующего импульса.

На рисунке 4.7, а приведена принципиальная схема синхронного RS-триггера с динамическим входом, а его условное графическое обозначение – на рисунке 4.7, б.

Если в обозначении синхронного RS-триггера с динамическим входом стрелка на входе С направлена к триггеру, то передача сигналов с информационных входов на выходы происходит по фронту импульса, а если стрелка направлена от обозначения триггера – передача сигнала осуществляется по спаду.

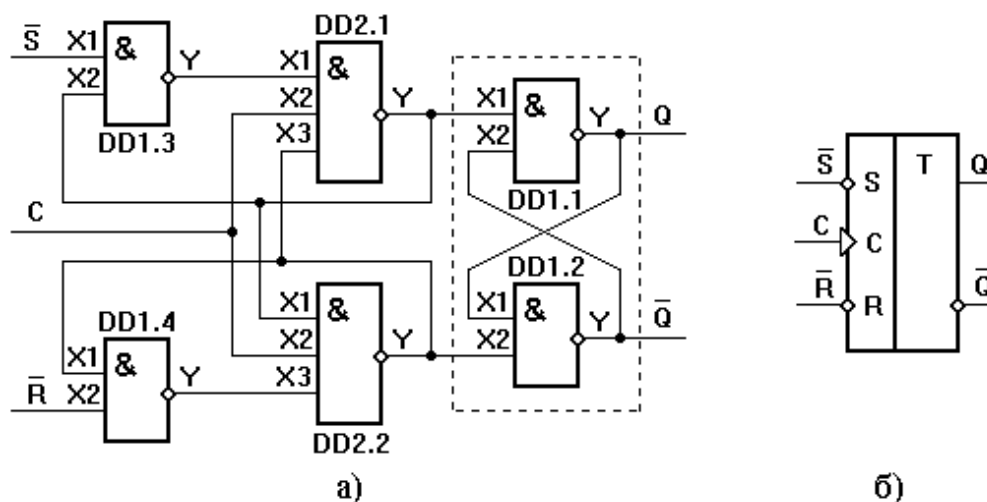


Рисунок 4.7. Синхронный RS-триггер с динамическим синхровходом

Из анализа работы логических элементов видно, что состояние триггера не меняется как при изменении сигналов на входах S и R при C=0, так и при C=1, если триггер переключился по фронту синхронизирующего импульса. При C=0 на выходах элементов DD2.1, DD2.2 будут сигналы логической единицы, и состояние на выходе триггера изменяться не будет при любых изменениях сигналов на входах R и S.

Установим на инверсном входе S уровень логического нуля, на инверсном входе R уровень логической единицы, и сигнал на входе C изменим с логического нуля на логическую единицу. На выходе элемента DD2.1 появится сигнал логического нуля и триггер перейдет в единичное состояние, или состояние триггера не изменится, если он находился в единичном состоянии. Оставляя на входе C сигнал логической единицы, перевести триггер в нулевое состояние не удастся. Для перевода триггера в противоположное состояние обязательно необходимо подать синхронизирующий импульс.

Задание 4.4.3.

- a. Предложите последовательность сигналов, подаваемых на входы триггера, предназначенную для выявления особенностей его поведения, описанных выше. (*Проверим утверждение, что при $C=0$ на выходах элементов DD2.1, DD2.2 будут сигналы логической единицы, и состояние на выходе триггера изменяться не будет при любых изменениях сигналов на входах R и S... Для этого подадим на вход.... а затем...*).
- b. Постройте временную диаграмму для входных сигналов и на ее основе запрограммируйте генератор слов.
- c. По рисунку 4.7 соберите схему синхронного триггера и убедитесь в его корректной работе в соответствии с ожидаемой диаграммой. Сделайте необходимые скриншоты.

4.5. Требования к отчету и защите

В отчете указываются название, цель работы. Описание выполненных лабораторных заданий с результатами в виде скриншотов, сохраненных результатов моделирования в файле формата соответствующей программы моделирования и выводами по каждому заданию.

На защите проверяются приобретенные знания теоретического и практического материала с демонстрацией результатов моделирования и по ответам на контрольные вопросы для самопроверки.

Отчёт должен содержать:

1. Формулировку цели работы.
2. Описание назначения и особенностей работы моделируемого элемента.
3. Изложение хода исследования, подтвержденного скриншотами схем и временными диаграммами.
4. Выводы по работе.

5. ЛАБОРАТОРНАЯ РАБОТА № 4. ИССЛЕДОВАНИЕ ЭЛЕМЕНТОВ РЕГИСТРОВОЙ И ОПЕРАТИВНОЙ ПАМЯТИ ЭВМ. JK-, D-ТРИГГЕРЫ, РЕГИСТРЫ, СЧЕТЧИКИ

5.1. Общие сведения

Цель:

изучение номенклатуры и структур типовых элементов с памятью

Материалы, оборудование, программное обеспечение:

IBM PC-совместимый персональный компьютер, программа MicroCap 9.0 5.0 Evaluation version, или иная, также являющаяся свободно распространяемой демоверсией профессиональной программы машинного моделирования электронных схем.

Условия допуска к выполнению:

Предварительное знакомство с УМП, отсутствие существенных текущих задолженностей по лабораторному практикуму.

Критерии положительной оценки:

Выполнение типового задания, оформление отчета по работе, ответы на вопросы для самопроверки.

Планируемое время выполнения:

Аудиторное время выполнения (под руководством преподавателя): 8 ч.

Время самостоятельной подготовки: 2 ч:

5.2. Теоретическое введение

RS-триггеры являются основой для построения более сложных схем с памятью. В общем случае подобные схемы отличаются наличием комбинационной части, используемой для задания логики формирования сигналов переключения внутреннего RS-триггера, и выходных сигналов, зависящих как от состояния RS-триггера, так и от входных сигналов.

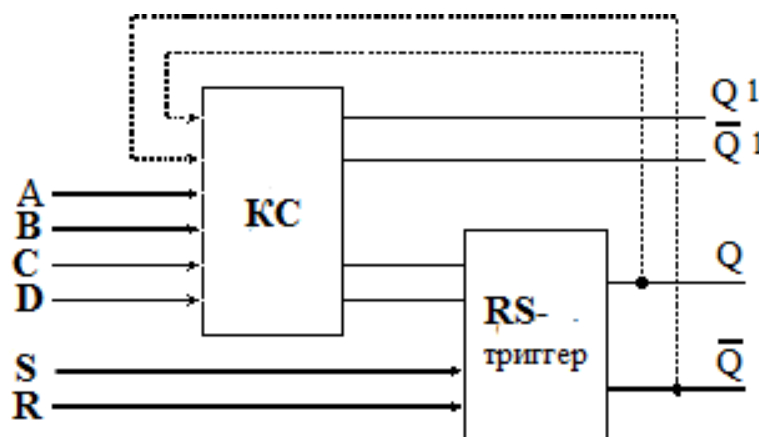


Рисунок 5.1. Общая структура триггера

На рисунке 5.1 приведен пример структурной схемы триггера, логика функционирования которого определяется входными сигналами A–D, где:

A, B, C, D – входы комбинационной схемы (КС), определяющей логику переключения состояний триггера

Q, \bar{Q} – выходы внутреннего RS-триггера

Q1, $\bar{Q}1$ – внешние выходы триггера

Литература:

Суханова, Н.В. Основы электроники и цифровой схемотехники [Электронный ресурс] : учебное пособие / Н.В. Суханова ; Министерство образования и науки РФ, Воронежский государственный университет инженерных технологий. - Воронеж : Воронежский государственный университет инженерных технологий, 2017. - 97 с. (ЭБС «Университетская библиотека онлайн») Стр. 73-79

Кардашев, Г.А. Цифровая электроника на персональном компьютере. Electronics Workbench и Micro-Cap / Г. А. Кардашев. - Москва : Горячая линия-Телеком, 2003. Стр. 209-262

Контрольные вопросы для самопроверки:

1. В чём заключаются отличия статического и динамического синхровходов?
2. Что такое двухтактный триггер?
3. Каково предназначение синхровхода?
4. Существуют ли асинхронные D-триггеры?
5. Почему JK-триггер получил название универсального?
6. Как из JK-триггера получить триггер со счётным входом?

5.3. Методические указания и порядок выполнения работы

1. Изучите теоретическую часть методических указаний и выполните задания 5.1 -5.5.
2. Оформите отчет в виде файла с названием «*Фамилия*, Лаб 5.doc».

5.3.1. Исследование JK-триггера

Из схемы JK-триггера (рисунок 5.2), соответствующей структуре, показанной на рисунке 4.1, следует, что его состояние зависит не только от сигналов на входах J и K, но и от логически связанных с ними сигналов Q и \bar{Q} .

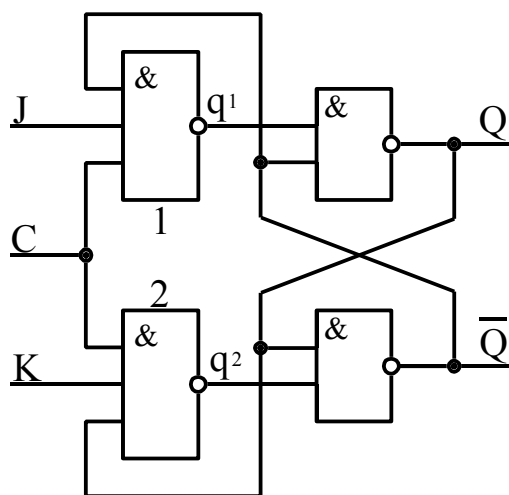


Рисунок 5.2. JK-триггер

Функциональная особенность JK-триггера состоит в том, что при всех входных комбинациях, кроме одной $J=K=1$, он действует подобно RS-триггеру, причём вход J играет роль входа S, а K-вход соответствует R-входу.

При $J=K=0$ на выходах элементов 1 и 2 будет $q_1=q_2=1$ (независимо от значений сигналов Q и \bar{Q}), а что представляет нейтральную комбинацию для триггерной ячейки, которая хранит записанную ранее информацию. Когда $J \neq K$, выходное состояние триггера будет определяться логическим элементом 1 или 2, на всех входах которого действует логическая 1.

Входная комбинация $J=K=1$ при любом состоянии триггера вызывает его переброс. Действительно, если $Q=1$, а $\bar{Q}=0$, то $q_1=1$, а $q_2=0$ (так как $K=Q=1$). Сигнал $q_2=0$ переключит триггерную ячейку. Переброс будет также иметь место при выходном состоянии $Q=0$, а $\bar{Q}=1$. В этом случае окажется $q_1=0$, $q_2=1$, и триггерная ячейка опрокинется, принимая противоположное состояние $Q=1$, а $\bar{Q}=0$.

Таким образом, подобно RS-триггеру, в JK-триггере J и K – это входы установки триггера в единицу или ноль. В отличие от RS-триггеров, в JK-триггере наличие двух единичных управляющих сигналов ($J=K=1$) приводит к переходу триггера в противоположное состояние. Причём, начиная с момента опрокидывания триггера, управляющее действие сигналов на входах J и K прекращается, так как изменяются сигналы на выходах логических элементов 1 и 2 (q_1, q_2).

Задание 5.3.1.

- Изучите схему рисунка 5.2. На основе рисунка 4.8 постройте автоматную модель триггера.
- На основе автоматной модели предложите последовательность сигналов J, C, K , обеспечивающую выявление всех характерных особенностей работы триггера. Опишите каждый такт временной диаграммы.
- Синтезируйте схему в EWB, подайте с генератора слов соответствующие последовательности на входы, зафиксируйте поведение схемы с помощью логического анализатора, сделав скриншоты.
- Сравните полученные результаты с ожидаемыми.

5.3.2. Исследование D-триггера

В синхронных RS-триггерах со статическим управлением остается неоднозначность состояния на выходе триггера, если с входов R, S одновременно убирать активные уровни сигналов. Для устранения неоднозначности в схему синхронного RS-триггера добавляют логический элемент «НЕ». Получившийся триггер является D-триггером со статическим управлением (рисунок 5.3, а).

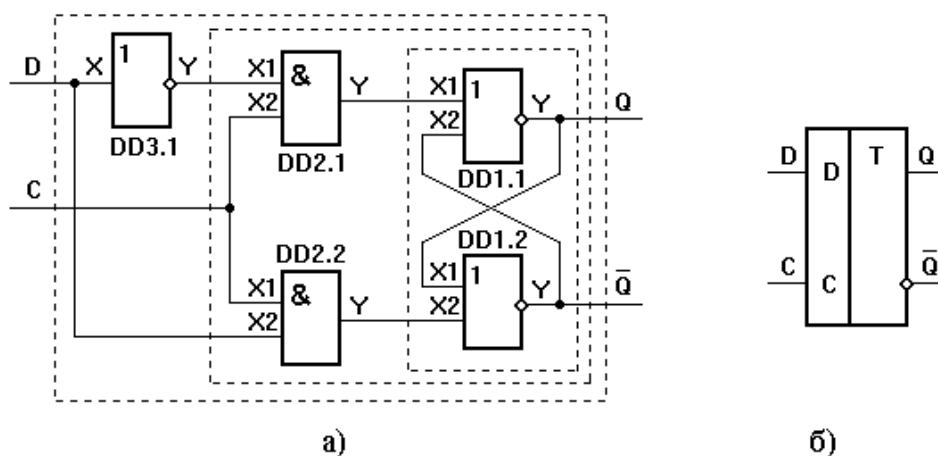


Рисунок 5.3. Структуры D-триггера со статическим синхрвходом

Такой триггер также называется триггером задержки, так как при разрешающем сигнале на синхровходе он устанавливается в состояние, соответствующее ранее поступившему сигналу на вход D.

В паузах между синхроимпульсами логические элементы DD2.1 и DD2.2 схемы управления закрыты и на их выходах существуют сигналы $Y=0$, что служит нейтральной комбинацией для триггерной ячейки, триггер хранит записанную информацию.

Аналогичными свойствами обладает другая реализация триггера, приведенная на рисунке 5.3, в.

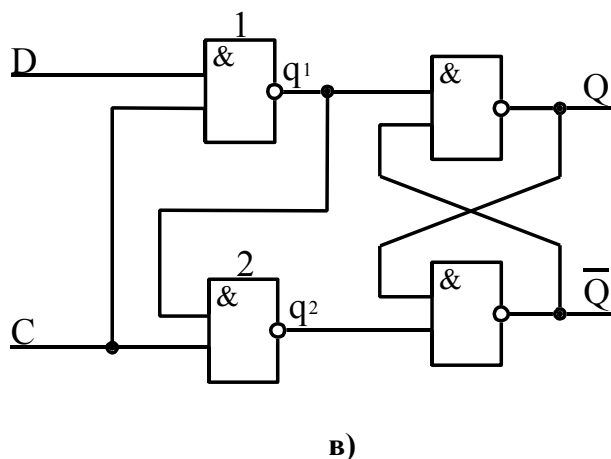


Рисунок 5.3. Структуры D-триггера со статическим синхровходом

Отличительной чертой элементов со статическим синхровходом является то, что при высоком уровне на входе C выходные сигналы фактически повторяют сигнал, поступающий на вход D. Этот недостаток устранен в D-триггерах с динамическим управлением (синхровходом). В них передача информации с информационных входов на выходы осуществляется либо по фронту синхронизирующего импульса, либо по спаду синхронизирующего импульса.

Функциональная схема D-триггера с передачей информации с входа на выход триггера по фронту синхронизирующего импульса приведена на рисунке 5.4 а, а его условное обозначение на принципиальных схемах – на рисунке 5.4 б.

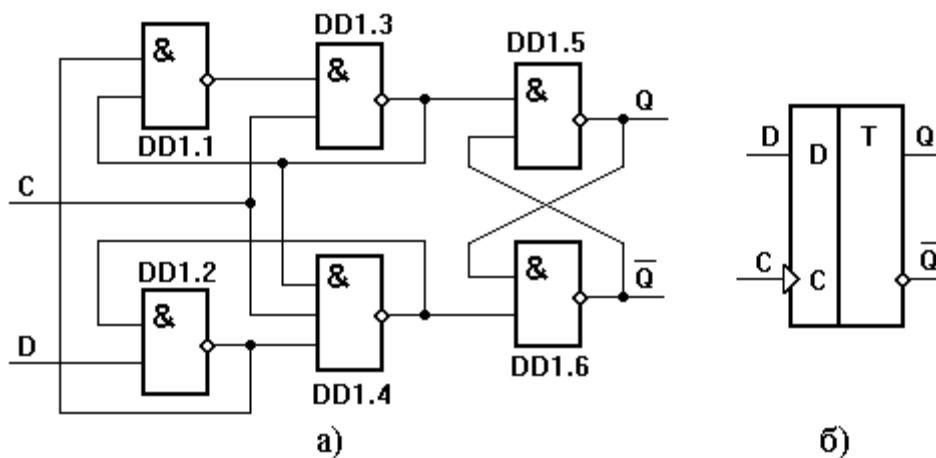


Рисунок 5.4. D-триггеры с динамическим управлением

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

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

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

В структуре схемы рисунка 5.4 фактически используются три RS-триггера, выполненные соответственно на логических элементах: DD1.2 и DD1.4; DD1.1 и DD1.1; DD1.5 и DD1.6. При этом первый триггер управляется непосредственно входным сигналом, состояние следующего зависит от первого, и они оба определяют состояние выходного триггера, сохраняющего значение сигнала D до очередного обновления по фронту синхроимпульса.

Сходным поведением обладают так называемые двухтактные (или «Master-Slave») триггеры.

Задание 5.3.2.

- Соберите схемы 5.3, в и 5.4, подайте на входы С и D с генератора слов последовательности, показанные на рисунке 5.5, зафиксируйте поведение схемы с помощью логического анализатора, сделав скриншоты.

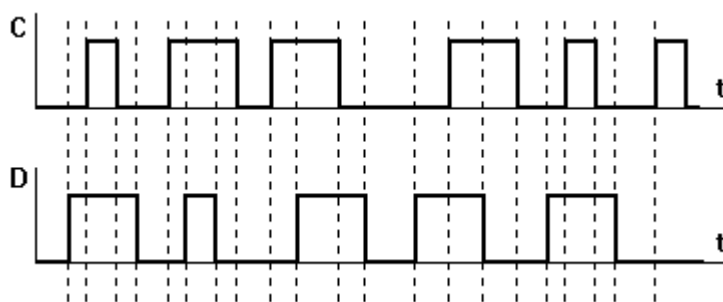


Рисунок 5.5. Тестовая последовательность сигналов генератора слов

- Сравните полученные временные диаграммы, объясните полученные результаты.

5.3.3. Исследование двухтактного триггера

Двухтактный триггер состоит из двух триггеров: главного («Master») и вспомогательного («Slave»). Иногда главный триггер называют ведущим, а вспомогательный ведомым. В качестве триггеров каскадов применяются однотипные триггеры со статическими синхривходами – например, RS, D, JK.

Цепь синхронизации управляет каскадами поочередно, таким образом, что по окончании синхронизирующего (тактового) импульса вспомогательный триггер переписывает информацию с выхода главного триггера.

На рисунке 5.6 приведена схема двухтактного RS-триггера, в котором, по сравнению с одноктактным прототипом, устранена неопределенность, возникающая при одновременном снятии активных логических сигналов с входов R и S.

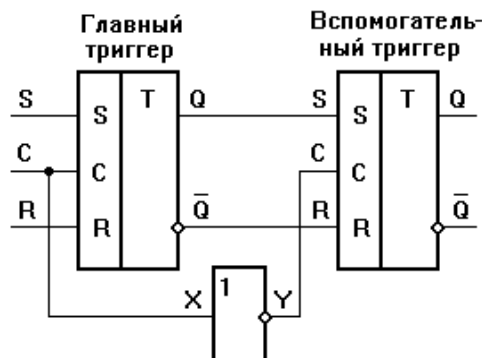


Рисунок 5.6. Двухтактный RS-триггер

Задание 5.3.3.

- Соберите схему двухтактного D-триггера на основе схем триггеров, приведенных на рис. 5.3..
- Подайте с генератора слов на входы схемы последовательности, показанные на рисунке 5.5, зафиксируйте поведение схемы с помощью логического анализатора, сделав скриншоты.
- Сравните результаты с полученными при выполнении задания 5.2, объясните результаты.

5.3.4. Исследование асинхронных счетчиков

Счетчиком называют цифровое устройство, обеспечивающее подсчет числа электрических импульсов. Коэффициент пересчета счетчика равен минимальному числу импульсов, поступивших на вход счетчика, после которых состояния на выходе счетчика начинают повторяться. Таким образом, простейший одноразрядный счетчик должен переключаться в противоположное состояние по приходу каждого очередного синхроимпульса. Очевидно, данными способностями будет обладать двухтактный RS-триггер, охваченный обратными связями, как показано на рисунке 5.7.

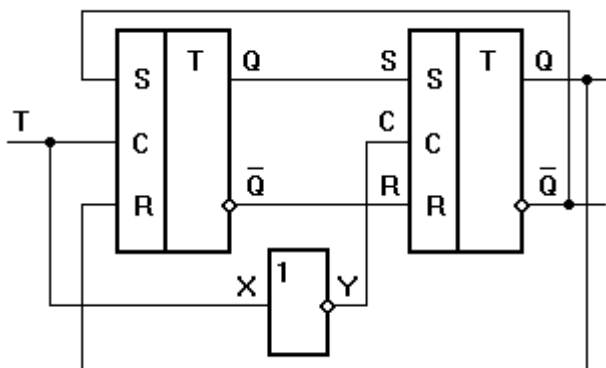


Рисунок 5.7. Счетный триггер (Т-триггер) на базе двухтактного RS-триггера

T-триггер может быть также построен на основе других типов триггеров соответствующим заданием обратных связей с помощью КС.

Последовательным включением отдельных каскадов можно построить многоразрядный асинхронный счетчик (под асинхронностью в данном случае понимается неодновременность появления сигналов на выходах различных разрядов).

Счетчик называют суммирующим, если после каждого очередного импульса цифровой код на выходе счетчика увеличивается на единицу. В вычитающем счетчике после каждого импульса на входе счетчика цифровой код на выходе уменьшается на единицу. Счетчики, в которых возможно переключение с режима суммирования на режим вычитания, называются реверсивными.

Наряду с двоичными, существуют счетчики с иными модулями счета (например, двоично-десятичные), а также использующие не позиционные системы (например, код Грея).

Задание 5.4.4.

- a. Соберите схему T-триггера на основе D-триггера (рис. 4.4 или синтезированного согласно заданию 4.3). Проверьте его работоспособность. Уточните, по фронту или по срезу сигнала происходит переключение.
- b. Найдите в библиотеке элементов EWB триггеры, функционально аналогичные собранному, и постройте на их основе многоразрядные двоичные суммирующий и вычитающий счетчики. Зафиксируйте поведение схем с помощью логического анализатора, сделав скриншоты.

5.3.5. Исследование регистров

Набор триггеров образует регистровое запоминающее устройство (регистр памяти, регистр).

Помимо фиксации и хранения многоразрядного слова данных, регистры, как правило, выполняют их обработку. С этой целью триггеры, образующие регистр, дополняются комбинационной схемой (аналогично рисунку 4.1), с помощью которой обеспечивается выполнение следующих типичных операций над словами:

- установка регистра в начальное состояние (сброс);
- прием слова в регистр;
- передача слова из регистра;
- поразрядовые логические операции;
- сдвиг слова влево или вправо на заданное число разрядов;
- преобразование последовательного кода слова в параллельный и обратно.

Внутренняя структура типичного регистра приведена на рисунке 5.8.

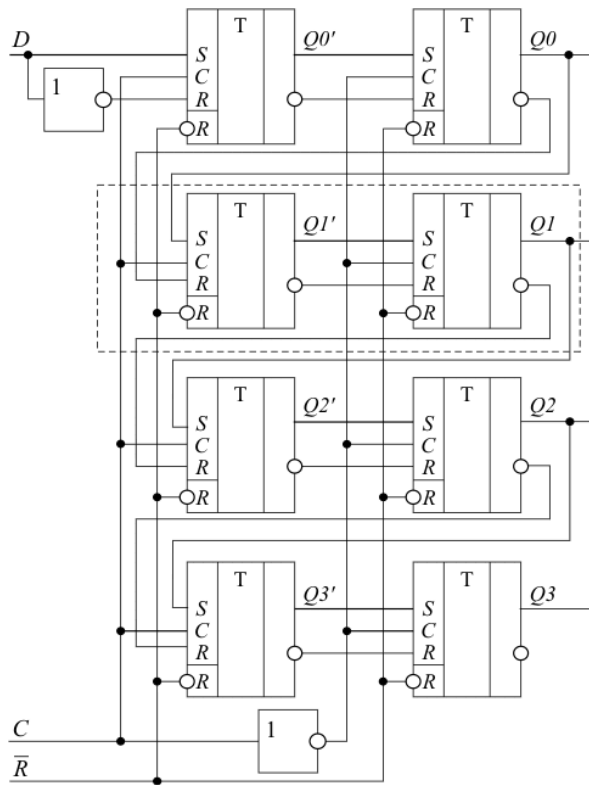


Рисунок 5.8. Внутренняя структура типичного регистра сдвига, выполненного на основе RS-триггеров

Поскольку выполняемые функции являются стандартными для большинства применений, регистры в настоящее время представлены достаточно широкой номенклатурой в ИМС различных серий.

В библиотеку EWB входят несколько разновидностей регистров, в том числе – многофункциональный восьмиразрядный регистр 74198 (отечественный аналог - К155ИР13). Условные графические обозначения регистра, соответствующие отечественным стандартам и используемые в EWB, приведены на рисунке 4.9. Обратите внимание на несовпадение обозначений некоторых выводов!

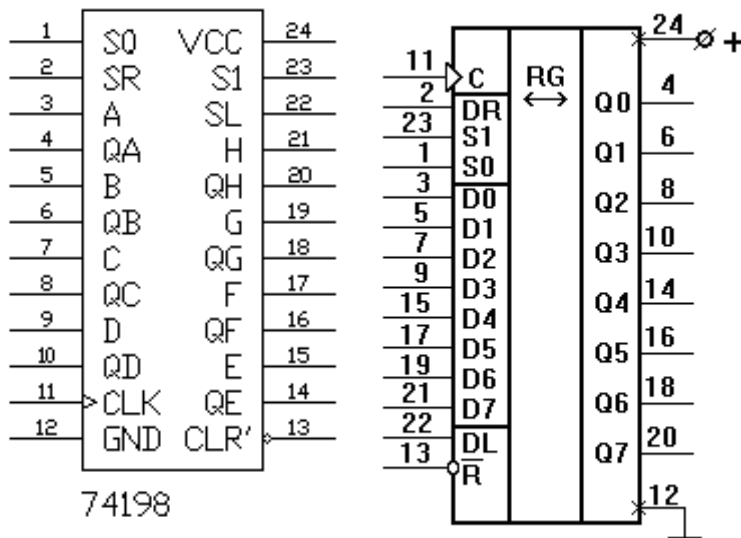


Рисунок 5.9. Регистр 74198 (отечественный аналог - К155ИР13)

Универсальный сдвиговый регистр применяется для преобразования информации из последовательной формы представления в параллельную и, наоборот, из параллельной в последовательную. Это необходимо, например, в сетевых картах, накопителях на жестких дисках, устройствах USB.

Регистр имеет как восемь входов для параллельной записи, так и соответствующие им восемь выходов параллельной информации. Сдвиг осуществляется по положительному фронту тактового сигнала С, причем сдвиг возможен как в сторону старших разрядов (вправо), так и в сторону младших разрядов (влево). Для наращивания разрядности у регистра IP13 имеются последовательные информационные входы DR и DL, сигналы с которых вдвигаются, соответственно, в младший и в старший разряды. Предусмотрен сброс всех выходов регистра в нуль по нулевому сигналу на входе –R. При единице на входе SR и нуле на входе SL по фронту сигнала С происходит сдвиг в сторону старших разрядов. При нуле на входе SR и единице на входе SL по фронту сигнала С происходит сдвиг в сторону младших разрядов. При обоих единичных сигналах на входах SR и SL по фронту сигнала С происходит параллельная загрузка информации в регистр..

Для преобразования информации из последовательной формы в параллельную необходимо на вход С подавать синхронизирующие импульсы, согласованные во времени с информационными импульсами на одном из последовательных входов. Для преобразования информации из параллельной формы представления в последовательную предварительно в регистр записывают информацию с информационных входов D0-D7, затем осуществляют сдвиг информации в нужную сторону, а сигнал снимают с выхода Q0 или Q7.

Входы				Функция
С	-R	SR	SL	
X	0	X	X	Сброс
0 → 1	1	1	0	Сдвиг вправо
0 → 1	1	0	1	Сдвиг влево
0 → 1	1	0	0	Хранение
0 → 1	1	1	1	Параллельная запись

Задание 5.3.5. Исследуйте работу регистра сдвига согласно следующей методике.

- Осуществить передачу информации с входов D0-D7 на выходы Q0-Q7 подачей синхронизирующего импульса положительной полярности на вход С. На входе R должен быть сигнал логической "1".
- Осуществить ввод информации (логических нулей и логических единиц) с входа DR (на входе R должен быть сигнал логической "1").
- Осуществить ввод информации (логических нулей и логических единиц) с входа DL (на входе R должен быть сигнал логической "1").
- Проверить режим хранения информации (на входе R должен быть сигнал логической "1").
- Произвести установку регистра в "0".
- Изобразить временную диаграмму работы регистра в режиме последовательного ввода информации с входа DR или DL.

5.4. Требования к отчету и защите

В отчете указываются название, цель работы. Описание выполненных лабораторных заданий с результатами в виде скриншотов, сохраненных результатов

моделирования в файле формата соответствующей программы моделирования и выводами по каждому заданию.

На защите проверяются приобретенные знания теоретического и практического материала с демонстрацией результатов моделирования и по ответам на контрольные вопросы для самопроверки.

Отчёт должен содержать:

1. Формулировку цели работы.
2. Описание назначения и особенностей работы моделируемых элементов.
3. Изложение хода исследований, подтвержденного скриншотами схем и временными диаграммами.
4. Выводы по работе.

Отчет предоставляется в электронном виде через ЭИОС

6. ЛАБОРАТОРНАЯ РАБОТА №5. ЗАПОМИНАЮЩИЕ УСТРОЙСТВА ЭВМ. ИССЛЕДОВАНИЕ СТРУКТУР И ФУНКЦИЙ. СИНХРОННЫЕ СЧЕТЧИКИ. МИКРОПРОГРАММНЫЕ УСТРОЙСТВА

6.1. Общие сведения

Цель:

Изучение принципов построения микропрограммных автоматов, приобретение навыков синтеза микропрограммных устройств управления.

Материалы, оборудование, программное обеспечение:

IBM PC - совместимый персональный компьютер, программа MicroCap 9.0 5.0 Evaluation version, или иная, также являющаяся свободно распространяемой демоверсией профессиональной программы машинного моделирования электронных схем.

Условия допуска к выполнению:

Предварительное знакомство с УМП, отсутствие существенных текущих задолженностей по лабораторному практикуму.

Критерии положительной оценки:

Выполнение типового задания, оформление отчета по работе, ответы на вопросы для самопроверки.

Планируемое время выполнения:

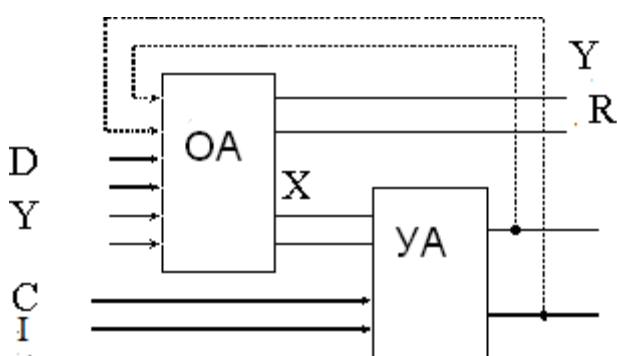
Аудиторное время выполнения (под руководством преподавателя): 8 ч.

Время самостоятельной подготовки: 2 ч.

6.2. Теоретическое введение

Структура триггера, приведенная ранее на рисунке 5.1., соответствует классической структуре микропрограммного устройства (см. рисунок 6.1).

Подобные устройства принято делить на две части: операционный автомат (ОА) и управляющий автомат (УА).



ОА – Операционный Автомат

УА – Управляющий Автомат

D – входные сигналы ОА / обрабатываемые данные

C – синхросигнал

R – выходные сигналы ОА / результат операции

X – сигналы, оповещающие УА о ходе выполнения микрооперации

Y – сигналы управления ОА, микрокоманды

I – внешняя команда задания режима работы устройства / код операции

Рисунок 6.1. Общая структура микропрограммного устройства

Микропрограммное устройство выполняет последовательность микроопераций (микропрограмму), порядок и логика выполнения которых задается УА, а конкретный вид и результат действий на каждом шаге микропрограммы зависит от ОА. Выполнение микропрограммы как правило синхронизируется внешним тактовым генератором, за счет чего переключение всех элементов и, соответственно, выработка всех выходных сигналов происходит строго одновременно. Подобные устройства получили название синхронных автоматов.

ОА предназначен для выполнения заданного набора микроопераций над входными данными D , выдачи результатов выполнения операций R , выработки значений логических условий $X=(x_0, x_1, \dots, x_m)$, которые являются оповещающими сигналами для управляющего автомата.

УА генерирует последовательность микрокоманд $Y=(y_1, y_2, \dots, y_n)$, определяемую поступившим кодом операции в соответствии с заданной микропрограммой и со значениями логических условий X . Каждая микрокоманда инициирует выполнение соответствующей микрооперации в ОА, реализуемой типовыми функциональными узлами обработки двоичных сигналов (например, сумматорами, дешифраторами, шифраторами, арифметико-логическими устройствами (АЛУ), схемами сравнения и т.п.).

Работа таких устройств обычно состоит в реализации некоторого алгоритма обработки информации, т.е. в выполнении упорядоченной последовательности определенных операций над поступающими данными. В некоторых случаях целью функционирования подобного автомата может являться сам процесс перехода между состояниями.

При построении устройств данного типа целесообразно использовать принцип микропрограммного управления, заключающийся в следующем:

1) любая операция, реализуемая устройством, рассматривается как сложное действие, которое разделяется на последовательность элементарных действий, называемых микрооперациями;

2) для управления порядком следования микроопераций используются логические условия X_i , принимающие в зависимости от результатов выполнения микроопераций значения 1 или 0;

3) процесс выполнения операций в устройстве описывается в форме алгоритма, представленного в терминах микроопераций и логических условий и называемого микропрограммой;

4) микропрограмма используется как форма представления функции устройства, на основе которой определяются его структура и порядок функционирования.

Особенностью принципа микропрограммного управления является возможность его применения для построения не только собственно устройств управления, но и любых других устройств, функционирование которых может быть задано в виде алгоритма или автоматной модели..

Литература:

Суханова, Н.В. Основы электроники и цифровой схемотехники [Электронный ресурс] : учеб. пособие / Н.В. Суханова ; Министерство образования и науки РФ, Воронежский государственный университет инженерных технологий. - Воронеж : Воронежский государственный университет инженерных технологий, 2017. - 97 с. (ЭБС «Университетская библиотека онлайн») Стр. 73-79.

Кардашев, Г.А. Цифровая электроника на персональном компьютере. Electronics Workbench и Micro-Cap / Г. А. Кардашев. - Москва : Горячая линия-Телеком, 2003. Стр. 204-263.

Контрольные вопросы для самопроверки:

1. Что называется микропрограммным устройством? В состав каких узлов ЭВМ могут входить устройства с микропрограммным управлением?
2. В чём состоят преимущества микропрограммного управления?
3. Каково назначение Операционного Автомата (ОА) в микропрограммном устройстве?
4. Каково назначение Управляющего Автомата (УА) в микропрограммном устройстве?
5. Каким образом можно реализовать счётчик на основе принципов микропрограммного управления?
6. Какие специфические операции должен выполнять счётчик команд ЭВМ? Как их можно описать с помощью языка автомата Мура?

6.3. Задание к лабораторной работе

1. Изучите теоретическую часть методических указаний и выполните задание 6.4.1.
2. Оформите отчет в виде файла с названием «Фамилия, Лаб 6.doc»..

6.4. Методические указания и порядок выполнения работы

6.4.1. Исследование синхронных счетчиков

Функционирование двухразрядного реверсивного синхронного счетчика может быть описано автоматной моделью, показанной на рисунке 6.2.

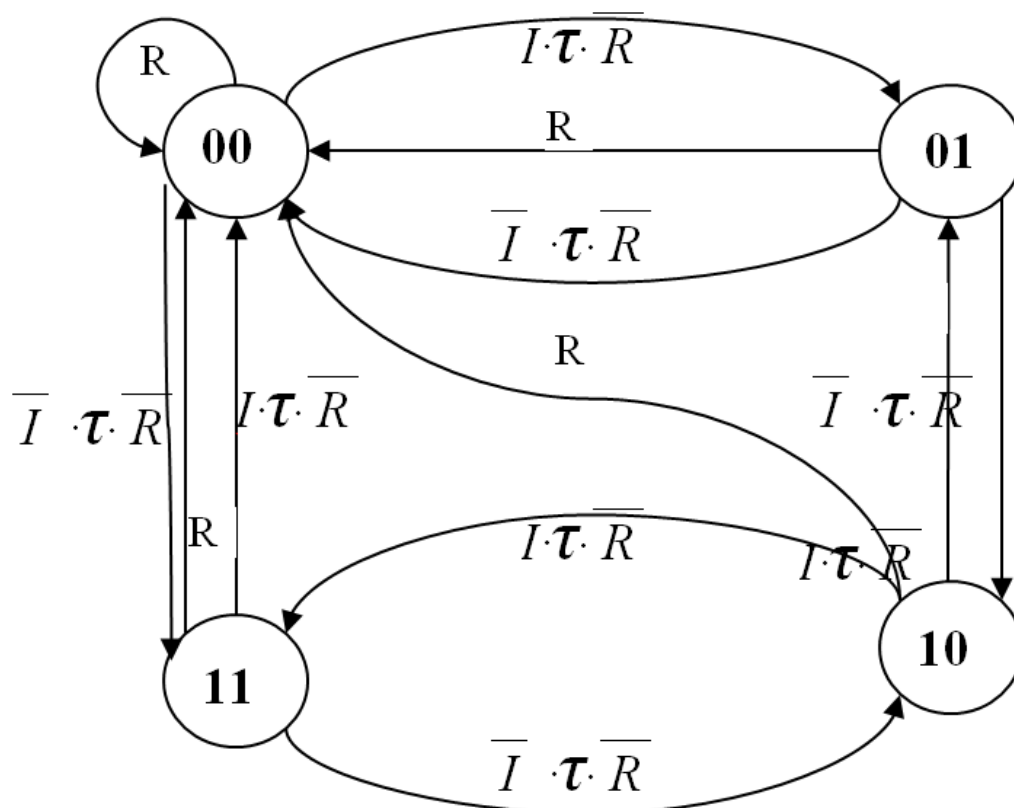


Рисунок 6.2. Автоматная модель двухразрядного реверсивного синхронного счетчика

Вершины графа соответствуют кодам состояния счетчика (00...11), дуги – разрешенным переходам между состояниями. В режиме прямого счета (код операции $I=1$) при соблюдении условия $I \cdot \tau \cdot \overline{R}$ по каждому синхросигналу происходит циклический переход между состояниями в направлении $00 \rightarrow 01 \rightarrow 10 \rightarrow 11 \rightarrow 00$ и т.д., в режиме обратного счета (код операции $I=0$), при соблюдении условия $\overline{I} \cdot \tau \cdot \overline{R}$ – в обратном порядке. Дуги, отмеченные как R , обозначают асинхронный сброс.

Номенклатура счетчиков, входящих в библиотеку элементов EWB, включает четырехразрядный двоичный реверсивный счетчик с предустановкой 155ИЕ13 (зарубежный аналог – SN74191)

Условные графические обозначения ИМС в соответствии с отечественным ГОСТ и используемого в EWB приведены на рисунке 6.3. (**Обратите внимание на отличия в системе обозначения выводов!**).

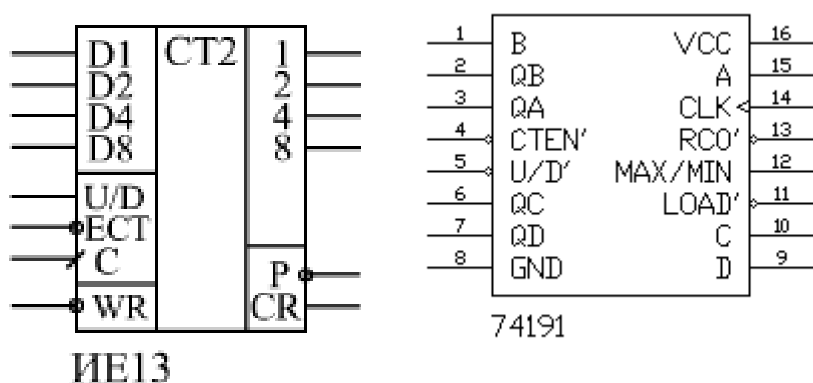


Рисунок 6.3. Условные графические обозначения ИМС 155ИЕ13 и SN74191

Микросхемы реверсивные, то есть допускают как прямой, так и обратный счет. Подсчет числа импульсов происходит по положительному фронту тактового сигнала C при нулевом уровне на входе разрешения счета ECT . Прямой счет осуществляется при нулевом уровне на входе управления U/D , обратный — при единичном уровне на входе U/D . Переключение уровней на входах U/D и ECT допускается только при положительном сигнале на тактовом входе C .

Положительный сигнал на выходе параллельного переноса CR появляется при переполнении (то есть достижении максимального числа, равного 15) при прямом счете или при достижении нуля при обратном (инверсном) счете. Имеется также выход последовательного переноса P , отрицательный импульс на котором вырабатывается при положительном сигнале CR и повторяет отрицательный импульс на тактовом входе C .

При каскадировании (совместном включении для увеличения разрядности), например, двух счетчиков, тактовые входы C обоих счетчиков объединяются, а сигнал переноса первого счетчика подается на вход разрешения счета (ECT) второго счетчика. В результате второй счетчик будет считать каждый шестнадцатый входной тактовый импульс (так как он будет срабатывать только при переносе от первого счетчика). Выходные сигналы второго счетчика будут переключаться по фронту общего тактового сигнала одновременно с выходными сигналами первого счетчика. Условием правильной работы будет в данном случае следующее: за период тактового сигнала должен успеть выработаться сигнал переноса первого счетчика.

Сброс счетчиков в нуль не предусмотрен, зато имеется возможность асинхронной параллельной записи информации по нулевому уровню сигнала параллельной записи – WR .

Входы				Режим
-WR	U/D	-ECR	C	
0	X	X	X	Параллельная запись
1	X	1	X	Хранение
1	0	0	0 → 1	Прямой счет
1	1	0	0 → 1	Обратный счет

Задание 6.4.1. Исследуйте работу синхронного счетчика на примере ИМС К555ИЕ13 (или её аналога).

- На основе описания, постройте автоматную модель работы счетчика.
- Предложите схему каскадирования счетчиков, необходимого для создания электронных цифровых часов (прямой счет, модули счета разрядов - 24 ч, 60 мин). Укажите на автоматных моделях разрядов счетчика соответствующие переходы и их условия.
- Разработайте схему для автоматической предустановки (сброса) счетчиков по достижении содержимым счетчика максимального значения.
- Модифицируйте схему для режима обратного счета, при котором, по достижении содержимым счетчика значения 00:00, оно заменяется на 23:59.
- Проверьте правильность решений, смоделировав схемы на EWB.

6.5. Требования к отчету и защите

В отчете указываются название, цель работы. Описание выполненных лабораторных заданий с результатами в виде скриншотов, сохраненных результатов моделирования в файле формата соответствующей программы моделирования и выводами по каждому заданию.

На защите проверяются приобретенные знания теоретического и практического материала с демонстрацией результатов моделирования и по ответам на контрольные вопросы для самопроверки.

Отчёт должен содержать:

1. Формулировку цели работы.
3. Описание назначения и особенностей работы моделируемых элементов.
4. Изложение хода исследований, подтвержденного скриншотами схем и временными диаграммами.

Отчет предоставляется в электронном виде через ЭИОС

7. ЛАБОРАТОРНАЯ РАБОТА № 6. МИКРОПРОГРАММНЫЕ УСТРОЙСТВА И КОНТРОЛЛЕРЫ. ОРГАНИЗАЦИЯ И ТИПОВЫЕ СТРУКТУРЫ. ПРОЕКТИРОВАНИЕ СПЕЦИАЛИЗИРОВАННОГО МИКРОПРОГРАММНОГО АВТОМАТА

7.1. Общие сведения

Цель:

Изучение принципов работы микропрограммных автоматов, исследование типовых структур, проектирование специализированного микропрограммного автомата.

Материалы, оборудование, программное обеспечение:

IBM PC-совместимый персональный компьютер, программа MicroCap 9.0 5.0 Evaluation version, или иная, также являющаяся свободно распространяемой демоверсией профессиональной программы машинного моделирования электронных схем.

Условия допуска к выполнению:

Предварительное знакомство с УМП, отсутствие существенных текущих задолженностей по лабораторному практикуму.

Критерии положительной оценки:

Выполнение типового задания, оформление отчета по работе, ответы на вопросы для самопроверки.

Планируемое время выполнения:

Аудиторное время выполнения (под руководством преподавателя): 4 ч.

Время самостоятельной подготовки: 2 ч.

7.2. Теоретическое введение

Известно, что для синтеза комбинационных схем требуется набор логических элементов, реализующих некоторую полную систему логических функций (базис). Особенность последовательностных схем состоит в том, что они обладают памятью, для реализации которой требуются дополнительные элементы. В качестве таких элементов используются триггеры.

Опишем структуру синтезируемого автомата, относящегося к одной из разновидностей последовательностных схем.

Автомат состоит из набора N элементарных автоматов (триггеров T_1, T_2, \dots, T_n), комбинация состояний которых в каждый момент времени определяет внутреннее состояние всего автомата в целом. Под воздействием входных сигналов автомат должен переходить из одного состояния в другое. Для изменения состояния автомата необходимо переключить один или несколько триггеров, определяющих состояние автомата. Переключение триггеров осуществляется подачей сигналов управления $D, R/S$ или J/K в соответствии с типом триггера. Так как новое состояние автомата определяется тем, каково было его прежнее состояние и каков набор входных сигналов, то и сигналы $D, R/S$ или J/K являются функциями выходных сигналов триггеров (Q_1, Q_2, \dots, Q_n) и входных сигналов автомата (X_1, X_2, \dots, X_n).

Для формирования сигналов управления триггерами используется комбинационное устройство (комбинационная схема). Структура этого устройства определяет функцию переходов автомата. Функция выходов реализуется другим комбинационным устройством, формирующим выходные сигналы автомата (Y_1, Y_2, \dots, Y_n).

В автомате Мили выходные сигналы являются функциями его входных сигналов (X_1, X_2, \dots, X_n) и выходных сигналов триггеров (Q_1, Q_2, \dots, Q_n).

В автомате Мура выходные сигналы являются функциями только выходных сигналов триггеров (Q_1, Q_2, \dots, Q_n).

Функции переходов и выходов автомата могут быть заданы в форме таблиц переходов и выходов или с помощью графов.

В столбцах таблицы 7.1. (для каждого варианта) указаны состояния автомата, а в строках - входные сигналы. На пересечении столбца и строки в таблице переходов указано новое состояние, в которое переходит автомат.

В таблице 7.2. – значения выходных сигналов Y_1, Y_2 как функция от отмеченных состояний автомата (*не триггеров!*).

Граф состоит из узлов, отождествляемых с отдельными состояниями автомата. Связи между узлами показывают переходы автомата из одного состояния в другое под воздействием входных сигналов. На каждой связи сверху указывается входной сигнал, вызывающий данный переход (*и, для автомата Мили, сигнал, формируемый на его выходе до перехода его в новое состояние*).

Таким образом, для синтеза автомата необходимо выполнить следующее:

- 1) определить, какая комбинация состояний триггеров будет соответствовать каждому из внутренних состояний автомата, т.е. провести кодирование внутренних состояний автомата;
- 2) синтезировать комбинационное устройство формирования сигналов управления триггерами $D, R/S$ или J/K , используя таблицу переходов;
- 3) синтезировать комбинационное устройство, формирующее выходные сигналы автомата Y , используя таблицу выходов.

Литература:

Торгонский, Л.А. Проектирование центральных и периферийных устройств ЭВС [Электронный ресурс] : учеб. пособие / Л.А. Торгонский, П.Н. Коваленко ; Министерство образования и науки Российской Федерации, Томский Государственный Университет Систем Управления и Радиоэлектроники (ТУСУР). - Томск : Эль Контент, 2012. - Ч. II. Микропроцессорные ЭВС. - 176 с. (ЭБС «Университетская библиотека онлайн»). Стр. 28-40.

Кардашев, Г.А. Цифровая электроника на персональном компьютере. Electronics Workbench и Micro-Cap / Г. А. Кардашев. - Москва : Горячая линия-Телеком, 2003. Стр. 285-304.

Контрольные вопросы для самопроверки:

1. Каким автоматом является соответствующий полученному индивидуальному заданию микропрограммный автомат?
2. От чего зависит функция переходов?
3. Какое минимальное количество триггеров требуется для кодирования состояний проектируемого автомата?
4. Что произойдет, если возникнет комбинация состояний триггеров, не сопоставленная ни одному из внутренних состояний автомата (то есть, ни одной из вершин графа)? По какой причине это может произойти?

5. Руководствуясь какими критериями, следует выполнять кодирование состояний с помощью триггеров?
6. Триггеры какого типа лучше использовать? Почему?
7. Согласно каким критериям необходимо выполнять оптимизацию схемы?

7.3. Задание к лабораторной работе

Таблица 7.1. Функции переходов

Вариант	Вход X	Состояния – Переходы по значениям условия X							
		A0	A1	A2	A3	A4	A5	A6	A7
01	0	A1	A7	-	A4	A3	-	-	A0
	1	A3	A4	-	A1	A3	-	-	A7
02	0	A2	A5	A0	-	-	A1	A2	-
	1	A1	A6	A6	-	-	A6	A0	-
03	0	-	A3	-	A1	-	A7	A5	A3
	1	-	A6	-	A3	-	A3	A7	A1
04	0	A1	A7	A7	-	A2	-	-	A4
	1	A2	A1	A4	-	A0	-	-	A7
05	0	A3	-	-	A6	-	A3	A0	A5
	1	A6	-	-	A7	-	A0	A5	A3
06	0	-	A4	A5	-	A1	A2	A6	-
	1	-	A2	A1	-	A6	A4	A5	-
07	0	-	A5	A7	-	-	A6	A1	A2
	1	-	A6	A2	-	-	A7	A5	A1
08	0	A7	-	A6	A0	-	-	A2	A6
	1	A2	-	A0	A7	-	-	A3	A3
09	0	A1	A4	-	-	A5	A0	-	A1
	1	A4	A7	-	-	A0	A7	-	A7
10	0	-	A2	A7	A1	A3	-	-	A4
	1	-	A3	A2	A7	A1	-	-	A3
11	0	-	A3	A1	A5	-	A7	-	A7
	1	-	A5	A3	A2	-	A1	-	A3
12	0	A2	-	A4	-	A5	A6	A6	-
	1	A0	-	A5	-	A0	A2	A4	-
13	0	A5	-	-	A3	-	A0	A7	A6
	1	A3	-	-	A5	-	A5	A0	A7
14	0	A4	-	-	A6	A3	-	A7	A0
	1	A3	-	-	A0	A7	-	A4	A3
15	0	-	A3	-	A3	A5	A1	-	A4
	1	-	A4	-	A1	A7	A7	-	A5
16	0	A1	A5	A0	-	-	A2	A5	-
	1	A5	A1	A6	-	-	A6	A0	-
17	0	-	A3	-	A7	-	A3	A5	A1
	1	-	A5	-	A6	-	A1	A7	A7
18	0	A1	A7	-	A0	-	-	A6	A6
	1	A6	A0	-	A1	-	-	A7	A3
19	0	A3	-	-	A3	-	A0	A7	A5
	1	A5	-	-	A0	-	A7	A5	A6
20	0	-	A6	A5	-	A2	A2	A5	-
	1	-	A5	A4	-	A6	A1	A1	-

Таблица 7.1. Функции переходов (продолжение)

Вариант	Вход X	Состояния – Переходы по значениям условия X							
		A0	A1	A2	A3	A4	A5	A6	A7
21	0	-	A5	A6	-	-	A2	A1	A7
	1	-	A2	A5	-	-	A1	A6	A6
22	0	A2	-	A6	A0	-	-	A7	A7
	1	A6	-	A3	A3	-	-	A0	A6
23	0	A4	A1	-	-	A7	A0	-	A5
	1	A0	A7	-	-	A5	A7	-	A4
24	0	-	A3	A1	A3	A7	-	-	A7
	1	-	A2	A4	A7	A2	-	-	A4
25	0	-	A2	A3	A1	-	A2	-	A3
	1	-	A1	A5	A7	-	A7	-	A5
26	0	A4	-	A4	-	A0	A6	A5	-
	1	A2	-	A0	-	A5	A5	A2	-
27	0	A2	A6	A2	-	-	A6	A0	-
	1	A1	A0	A5	-	-	A5	A5	-
28	0	-	A3	-	A1	-	A3	A5	A6
	1	-	A5	-	A3	-	A1	A7	A7
29	0	A1	A4	-	-	A0	A5	-	A5
	1	A0	A5	-	-	A7	A4	-	A1
30	0	A5	-	-	A0	-	A3	A7	A6
	1	A3	-	-	A6	-	A5	A5	A0
31	0	-	A1	A4	-	A5	A2	A5	-
	1	-	A4	A5	-	A1	A6	A2	-
32	0	-	A5	A4	-	A1	A5	-	A4
	1	-	A2	A5	-	A2	A7	-	A5
33	0	A3	-	A3	A7	-	-	A0	A6
	1	A2	-	A0	A6	-	-	A7	A2
34	0	A5	A0	-	-	A1	A7	-	A5
	1	A0	A1	-	-	A4	A5	-	A4
35	0	-	A3	-	A1	-	A7	A5	A3
	1	-	A6	-	A3	-	A3	A7	A1
36	0	-	A2	A3	A5	-	A2	-	A1
	1	-	A3	A1	A7	-	A7	-	A5
37	0	A5	-	A0	-	A2	A5	A6	-
	1	A2	-	A4	-	A6	A0	A5	-
38	0	A3	-	-	A5	-	A3	A7	A0
	1	A5	-	-	A7	-	A3	A0	A3
39	0	A0	-	-	A6	A3	-	A7	A3
	1	A4	-	-	A0	A7	-	A4	A6
40	0	-	A3	-	A1	A1	A4	-	A5
	1	-	A7	-	A5	A4	A5	-	A3

Таблица 7.1. Функции переходов (окончание)

Вариант	Вход X	Состояния – Переходы по значениям условия X							
		A0	A1	A2	A3	A4	A5	A6	A7
41	0	A3	-	A0	A3	A2		A6	-
	1	A2	-	A4	A0	A6		A3	-
42	0	A3	A0	-	A5	-	A3	A1	
	1	A5	A3	-	A1	-	A3	A0	
43	0	A0	A3	-	A6		-	A7	A3
	1	A1	A7	-	A0		-	A1	A6
44	0	A2	A6	A2	A6	-		A0	-
	1	A1	A0	A3	A3	-		A3	-
45	0	A3	A3	-	A1	-		A0	A6
	1	A1	A0	-	A3	-		A7	A7
46	0		A3	-	A1	A2	A5	-	A5
	1		A5	-	A2	A7	A4	-	A3
47	0	A3	A0	-	A0	-		A7	A6
	1	A1	A6	-	A6	-		A5	A0
48	0	-	A1	A4			A2	A5	A6
	1	-	A5	A6			A1	A7	A7
49	0	A3		A4	-	A2	A5	-	A5
	1	A2		A5	-	A7	A4	-	A3
50	0	A5	-	-		A0	A4	A7	A6
	1	A4	-	-		A6	A5	A5	A0
51	0	A0		A4	-	A5	A2	A5	-
	1	A4		A5	-	A0	A6	A2	-
52	0	-		A4	-	A6	A5	A5	A4
	1	-		A6	-	A2	A7	A2	A5
53	0	A3	A3		A7	-	-	A0	A6
	1	A1	A0		A6	-	-	A7	A1
54	0	A5	A0	A3	-	A1	A3	-	
	1	A0	A1	A4	-	A4	A5	-	
55	0	A3	A3	-	A1	-		A0	A6
	1	A1	A0	-	A3	-		A7	A7
56	0	A1	A2	A2	-		A5	-	A5
	1	A2	A5	A7	-		A7	-	A1
57	0	A7	A3	-	A0	-		A7	A6
	1	A3	A1	-	A6	-		A1	A0
58	0	-	A3	A4	-	A5	A2	A5	-
	1	-	A4	A5	-	A1	A6	A1	-
59	0	A0	-	-	A4	A3	-	A0	A3
	1	A4	-	-	A0	A7	-	A4	A6
60	0	A5	A3	-	A4	A1	A4	-	
	1	A3	A0	-	A5	A4	A5	-	

Таблица 7.2. Функции выходов

Вариант	Выходы <i>Y</i>								
		<i>A0</i>	<i>A1</i>	<i>A2</i>	<i>A3</i>	<i>A4</i>	<i>A5</i>	<i>A6</i>	<i>A7</i>
1, 6	<i>Y1</i>	+		+	+		+		
	<i>Y2</i>		+		+		+		+
2, 7	<i>Y1</i>		+	+		+		+	
	<i>Y2</i>	+		+	+			+	
3, 8	<i>Y1</i>		+			+	+		+
	<i>Y2</i>	+			+	+	+		
4, 9	<i>Y1</i>		+	+		+			+
	<i>Y2</i>	+		+			+	+	
5, 0	<i>Y1</i>			+	+				+
	<i>Y2</i>	+	+				+		+

Методические указания и порядок выполнения работы

На основе задания, выбранного из таблиц переходов и выходов в соответствии с вариантом:

1. Построить блок-схему алгоритма микропрограммного автомата. Номер блока – в соответствии с состоянием *A* из Табл. 7.1, операция – в соответствии со значением выходного сигнала из Табл. 7.2.
2. Построить граф автомата Мура, представить состояния автомата вершинами, переходы – дугами.
3. Выполнить двоичное кодирование внутренних состояний автомата. Разметить вершины в соответствии с выполненным кодированием.
4. Построить и минимизировать выражения для вычисления сигналов управления триггерами и выходных сигналов как функции от входных сигналов и состояния автомата.
5. Построить принципиальную электрическую схему автомата в базисе триггеров заданного типа и логических элементов И-НЕ.
6. Смоделировать работу автомата с помощью EWB.

7.4. Требования к отчету и защите

В отчете указываются название, цель работы. Описание выполненных лабораторных заданий с результатами в виде скриншотов, сохраненных результатов моделирования в файле формата соответствующей программы моделирования и выводами по каждому заданию.

На защите проверяются приобретенные знания теоретического и практического материала с демонстрацией результатов моделирования и по ответам на контрольные вопросы для самопроверки.

Отчёт должен содержать:

1. Исходное задание (строки из табл. 7.1 и 7.2 в соответствии с вариантом).
2. Блок-схему алгоритма микропрограммного автомата.
3. Граф автомата Мура.

4. Обзор типов, сравнение характеристик, выбор и обоснование типа используемых триггеров.
5. Таблицу истинности и выражения, описывающие вычисление сигналов управления триггерами D_{1-3} , R_{1-3}/S_{1-3} , J_{1-3}/K_{1-3} (в соответствии с типом выбранного триггера, по Табл. 1).
6. Выражения, описывающие вычисление выходных сигналов Y_1 и Y_2 (по Табл. 2).
7. Карты Карно с минимизацией выражений для вычисления сигналов D_{1-3} , R_{1-3}/S_{1-3} , J_{1-3}/K_{1-3} , Y_1 , Y_2 .
8. Минимизированные ДНФ сигналов D_{1-3} , R_{1-3}/S_{1-3} , J_{1-3}/K_{1-3} , Y_1 , Y_2 .
9. Обзор номенклатуры комбинационных элементов и схем EWB.
10. Принципиальную электрическую схему автомата в соответствующем базисе.
11. Результаты моделирования работы автомата (поведение при подаче набора входных сигналов X , обеспечивающего последовательный переход автомата между состояниями по всем дугам графа).

8. ЛАБОРАТОРНАЯ РАБОТА № 7. СИСТЕМА КОНСТРУКТИВНЫХ ЭЛЕМЕНТОВ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА

8.1. Общие сведения

Цель:

Знакомство с системой конструктивных элементов персонального компьютера, практическое освоение правил и особенностей монтажа и демонтажа основных компонентов ПК.

Материалы, оборудование, программное обеспечение:

Комплект плат и устройств IBM PC-совместимого персонального компьютера, инструменты и лист бумаги для составления технологической карты (чек-листа) процессов демонтажа и монтажа.

Условия допуска к выполнению:

Изучение правил техники безопасности, предварительное знакомство с УМП, отсутствие существенных текущих задолженностей по лабораторному практикуму.

Критерии положительной оценки:

Выполнение типового задания, оформление отчета по работе, ответы на вопросы для самопроверки.

Планируемое время выполнения:

Аудиторное время выполнения (под руководством преподавателя): 4 ч.

Время самостоятельной подготовки: 4 ч.

8.2. Теоретическое введение

Литература:

<https://intuit.ru/studies/courses/60/60/info> Архитектура и организация ЭВМ

<https://intuit.ru/studies/courses/92/92/lecture/28396> Современные архитектуры вычислительных систем

Горнец, Н.Н. ЭВМ и периферийные устройства. Компьютерные и вычислительные системы : учеб. / Н. Н. Горнец, А. Г. Рощин . - Москва : Академия, 2012. - 235 с. Стр. 169-198.

Контрольные вопросы для самопроверки:

1. Докажите, что структуры и функционирование основных элементов ПК в основном аналогичны микропрограммным автоматам.

2. Объясните требования унификации и стандартизации основных конструктивно-технологических решений архитектуры ПК (на примере таких понятий, как форм-фактор, стандартные интерфейсы, многоуровневая архитектура с использованием виртуализации)

3. Чем обеспечивается соблюдение таких требований, как открытость, адаптируемость, конфигурируемость под конкретные задачи (на примере таких понятий, как различные форм-факторы, разный состав и функциональность элементов чипсета, настраиваемость отдельных элементов, модульность конструкции, наличие BIOS SETUP, применение процедуры «раскрутки»)

4. Подтвердите эволюционное развитие архитектур ПК (на примере таких понятий, как совершенствование алгоритмов обработки данных и технологий элементной базы

ПЭВМ, приводящие к модификации старых и появлению новых интерфейсов, компонентов, структур и архитектур – процессоры, память, накопители, UEFI...).

5 Подтвердите фактами, полученными в ходе выполнения лабораторной работы, что аппаратная структура ПК с момента их появления существенно изменилась, однако совместимость сохраняется за счёт использования микропрограммной и программной эмуляции новыми устройствами режимов работы «традиционных» компонентов, которая обеспечивается средствами BIOS и операционной системы.

8.2.1. Основные сведения

Открытая магистрально-модульная архитектура, положенная в основу современных ПК, предполагает использование конструктивных модулей следующих уровней:

- первого уровня – типовой элемент замены, например – планка памяти, процессор или иной элемент, подключаемый к разъему на кросс-плате;
- второго уровня – блок, например – системная плата ПК, дискретная видеокарта или накопитель, расположенные в системном блоке с источником питания;
- третьего уровня – корпус (стойка) – непосредственно конструктив, объединяющий ранее упомянутые модули в единое целое, обеспечивающий как автономное их функционирование, так и работу в составе системы

Системная интеграция должна обеспечиваться механической, электрической и информационной совместимостью отдельных компонентов различных уровней, что требует понимания и учета стандартов, регламентирующих их характеристики.

Конструктивные особенности, вызванные необходимостью учета специфических требований к монтажу компонентов, электропитанию и охлаждению ПК, привели к появлению стандартов системного дизайна, которые в настоящее время могут быть представлены спецификациями AT, ATX, VTX и некоторыми другими.

Стандарты определяют следующие параметры модулей:

- геометрические размеры материнских плат;
- общие требования по положению разъемов и отверстий на корпусе;
- форма и положение ряда разъемов (преимущественно питания);
- геометрические размеры блока питания;
- положение блока питания в корпусе;
- электрические характеристики блока питания.

Внешними признаками, позволяющими различать системы данных стандартов, являются:

- тип выключателя цепи первичного электропитания (механический или электронный);
- конструкция разъема питания на системной плате (разъемный однорядный или неразъемный двухрядный);
- вид отверстия на задней панели корпуса для разъемов системной платы (для панели разъемов или только для клавиатуры и мыши);
- количество и вид интерфейса системной шины;
- расположение процессора и разъемов системной шины относительно панели разъемов и т.п.

Внутри каждого стандарта компоновки могут различаться конкретным фактором корпуса или внутреннего блока.

Корпуса в зависимости от расположения, размера и пропорций могут подразделяться на FullDesktop, BabyDesktop, SlimDesktop, SuperTower, Big- (Full)Tower,

MidiTower, MiniTower, MicroTower и иные подобные. Дополнительными характеристиками являются число и расположение посадочных мест для накопителей, расположение индикаторов, элементов управления и интерфейсных разъемов на лицевой панели и т.п.

Блоки питания, помимо принадлежности к стандарту, отражающейся в номенклатуре напряжений питания и конструкции разъема для подключения к системной плате, различаются отдаваемой электрической мощностью, направленностью воздушного потока, геометрическими размерами.

Системные платы, в зависимости от размера и расположения монтажных отверстий, могут носить уточняющие обозначения – mini-, micro-, nano- и т.п., с указанием конкретного стандарта ATX, VTX, DTX или ITX.

Накопители, размещаемые в монтажных корзинах или салазках, классифицируются в соответствии с форм-фактором, указывающим ширину посадочного места (в дюймах).

Контроллеры, размещаемые в слотах расширения, подразделяются на полноразмерные, низкопрофильные, двойной ширины и иные, с учетом специфики размещения и монтажа разъемов.

8.3. Задание к лабораторной работе

1. Ознакомьтесь с методическими указаниями, а правила техники безопасности заучите наизусть.
2. Изучите конструкцию, состав и расположение элементов представленных в лаборатории компьютеров. Отметьте общие черты и различия в их исполнении.
3. Выполните пункты 1-7 раздела «Ход лабораторной работы», используя предоставленные инструменты и оборудование.

Примечание: при невозможности выполнения п. 2-3 задания в лаборатории ознакомьтесь с конструктивно-технологическими особенностями компьютеров и ходом их сборки-разборки по самостоятельно найденным видео и иным доступным источникам (см. ссылки). При желании (и наличии навыков, на собственный страх и риск) проведите техническое обслуживание собственного ПК (снимите крышку корпуса, почистите внутренности, сфотографируйте результат).

4. Аккуратно оформите отчет в электронной форме.

8.4. Методические указания и порядок выполнения работы

8.4.1. Правила техники безопасности:

При выполнении любых действий, связанных с манипуляцией с внутренними элементами ПК, следует неукоснительно руководствоваться приведенными правилами!

1. Прежде, чем открывать системный блок, отключите питание системного блока и отсоедините все кабели питания (в некоторых случаях целесообразно отключить и убрать все внешние кабели и шнуры).
2. Снимите с рук все металлические предметы (кольца и т.п.).
3. Не трогайте детали внутри системного блока и периферийных устройств, если компьютер подключен к цепи первичного электропитания и пока не исчезнет заряд на конденсаторах блока питания.
4. Не размещайте внутри системного блока никакие посторонние детали или иные предметы.
5. Обязательно отключайте питание (кабель питания!) в момент, когда подключаются или отключаются платы расширения, накопители, периферийные устройства и т.п.

6. Перед тем, как взять в руки плату или дотронуться до внутренних компонентов, снимите с себя статическое электричество, дотронувшись, например, до защитного кожуха блока питания.

7. Пользуйтесь только исправными штатными инструментами и оборудованием.

8.4.2. Ход работы

Подготовьте рабочее место, инструменты и лист бумаги для составления технологической карты (чек-листа) процессов демонтажа и монтажа.

1. Изучите конструкцию компьютера, классифицируйте его в целом и отдельные компоненты на соответствие стандартам (см. «Основные сведения»).

Выделите модули, относящиеся к различным уровням иерархии конструктивных элементов. При необходимости проведите замеры.

Выполните сравнительную оценку особенностей конструкции отдельных модулей.

2. Отсоедините от системного блока все подключенные внешние устройства, зафиксировав в технологической карте размещение отключенных разъемов, положение переключателей напряжения и другие необходимые сведения.

- Не прикладывайте излишние физические усилия при разборке!
- Не вынимайте процессор из процессорного гнезда!
- При любых сомнениях обращайтесь к преподавателю!

3. Снимите с системного блока крышку, отметив число и вид монтажных винтов, и разместите системный блок в устойчивом положении, обеспечив доступ к внутренним элементам.

4. Изучите расположение внутренних съемных элементов (контроллеры периферийных устройств, накопители, шлейфы и разъемы и т.п.).

5. Зафиксируйте в технологической карте их расположение, особо обратив внимание на ориентацию ключей, местоположение контроллеров в разъемах, наличие джамперов и переключателей на системной плате.

6. Произведите полную разборку системного блока компьютера до уровня отдельных конструктивных модулей, а именно, демонтируйте, фиксируя каждый шаг:

- шлейфы, подсоединяющие отдельные внутренние компоненты;
- контроллеры (платы расширения системной платы);
- блок питания;
- кулер и процессор (при возможности);
- модули памяти;
- другие съемные элементы.

Обращайте внимание на снимаемые крепежные элементы (винты, шайбы и т.п.), не допуская их смешивания.

Зафиксируйте в технологической карте основные характеристики снятых элементов (назначение, серийный номер, обозначения на корпусах БИС).

Проведите Техническое обслуживание, заключающееся в очистке снятых элементов от пыли и других загрязнений.

7. Соберите системный блок в обратном порядке, строго руководствуясь подготовленной технологической картой, не закрывая корпус, подключите внешние кабели и представьте результаты работы преподавателю.

Собранный компьютер самостоятельно включать в сеть запрещается!

8.5. Требования к отчету и защите

1. Описание хода выполнения лабораторной работы.
2. Технологическую карту демонтажа и монтажа.
3. Описание платы + фото (изображения из статей, кадры из видео) основных сменных элементов.
4. Описание состава ядра компьютера и дополнительных компонентов.
5. Сравнение конструкции изученного компьютера и его ранних аналогов (см. ссылки в методичке).
6. Список ссылок на использованные видео и интернет-источники

9. ЛАБОРАТОРНАЯ РАБОТА № 8. ПРОБЛЕМЫ ИНТЕГРАЦИИ КОМПОНЕНТОВ СИСТЕМНОГО ЯДРА ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА

9.1. Общие сведения

Цель:

Практическое освоение методов настройки, поиска и устранения неисправностей основных компонентов системного блока.

Материалы, оборудование, программное обеспечение:

Комплект плат и устройств IBM PC-совместимого персонального компьютера: системная плата, видеосистема, клавиатура, мышь, блок питания, загрузочное устройство (флоппи-дискетод или жесткий диск), загрузочная дискета (диск, USB flash), тестовые программы, инструменты.

Условия допуска к выполнению:

Изучение правил техники безопасности, предварительное знакомство с УМП, отсутствие существенных текущих задолженностей по лабораторному практикуму.

Критерии положительной оценки:

Выполнение типового задания, оформление отчета по работе, ответы на вопросы для самопроверки.

Планируемое время выполнения:

Аудиторное время выполнения (под руководством преподавателя): 4 ч.

Время самостоятельной подготовки: 4 ч.

9.2. Теоретическое введение

Литература:

Горнец, Н.Н. ЭВМ и периферийные устройства. Компьютерные и вычислительные системы : учеб. / Н. Н. Горнец, А. Г. Рощин . - Москва : Академия, 2012. - 235 с. Стр. 169-198.

CPU Configuration

<http://www.nastrojkabios.ru/protessor/>

http://on-line-teaching.com/bios/16_CPU.html

Архитектура и организация ЭВМ

<https://intuit.ru/studies/courses/60/60/info>

Современные архитектуры вычислительных систем

<https://intuit.ru/studies/courses/92/92/lecture/28396>

Контрольные вопросы для самопроверки:

1. Что такое процедура раскрутки?
2. В какой последовательности в ходе раскрутки и загрузки происходит проверка компонентов и использование внесённых в BIOS SETUP изменений?
3. Каким образом осуществляется тестирование оперативной памяти?
4. Каким образом проверяется целостность ПЗУ?
5. Как можно проверить исправность процессора?
6. Какие функции выполняет BIOS SETUP?
7. Что такое тайминг? Каким образом его значение влияет на производительность и надёжность компьютера?
8. Как проявляются неисправности жёсткого диска?

Соответствие конструктивных элементов ПК стандартам должно гарантировать их полную совместимость. На практике же фактически рекомендательный характер этих стандартов, недокументированность ряда важных параметров, развитие технологий и рост требований к характеристикам компьютеров ведет к появлению проблем, влекущих необходимость выполнения настройки и диагностирования компонентов ПК. Особенно актуально решение проблем интеграции системного ядра – основного элемента, обеспечивающего работоспособность системы.

В состав системного ядра ПК включают подмножество элементов, достаточное для автономной работы - системную плату с установленным процессором и оперативной памятью, а также минимальный набор компонентов, обеспечивающих загрузку и контроль ее хода. Очевидно, что обязательно наличие соответствующего источника питания.

Работа посвящена знакомству с практической методикой настройки режимов, поиска и устранения неисправностей системного блока, а также с некоторыми тестовыми программами, обеспечивающими локализацию возможных неисправностей и объективную оценку характеристик компьютера.

Проблемы, связанные с определением неисправностей, могут быть разбиты на четыре основные группы:

1. Проблемы системы электропитания – система не запускается (процесс раскрутки не начался);
2. Нет сообщений об ошибках; система запускается, но при работе дает сбой и сообщения об ошибках не появляются (вероятно, исправно системное ядро, но процесс раскрутки не вышел за его пределы);
3. Система запускается, но при работе происходят сбои и появляются сообщения об ошибках (системное ядро вероятно исправно, процесс раскрутки охватил смежные устройства, некоторые из которых работают не корректно);
4. Система запускается, но не обеспечивает необходимые режимы работы.

Для того, чтобы устранить проблему, необходимо представлять, к какой группе относится проблема, и только после этого можно приступить к препарированию системного блока. Инструкцию по его вскрытию дать возможным не представляется, так как корпуса разнятся весьма значительно, но можно с уверенностью утверждать, что скорее всего потребуется крестообразная отвертка среднего размера.

9.2.1. Проблемы системы электропитания

Проблемы системы электропитания проявляются, как правило, одним из следующих образов:

- не слышно звука вентилятора, не горят индикаторы, не слышно звуковых сигналов;
- при наличии указанных выше признаков – нет изображения на мониторе, компьютер не загружается, либо не работает какой-либо накопитель;
- компьютер загружается нормально, но периодически самопроизвольно, в случайные моменты времени, сбрасывается. В этом случае обычно виноват источник первичного электропитания – электрическая сеть, и выходом из положения может являться приобретение UPS'a; но причиной может являться пробой в электронной схеме блока питания, или, к примеру, короткое замыкание в иных цепях. Но об этом позже.

Часто указанные проблемы возникают при нарушении соединения кабелей питания системного блока – как сетевого, так и выходных. Существует три места локализации подобных отказов:

- розетка;
- соединители кабелей питания и выключатель;

- источник питания и внутренние соединения.

1. Проверка розетки и кабелей питания обычно трудностей не вызывает – достаточно легкого покачивания кабеля в разъеме и переключения компьютера в другую, заведомо работоспособную, розетку. Обратите внимание на то, чтобы системный блок, монитор и принтер были подключены к одной и той же фазе сети – перераспределение токов через общий провод интерфейсных кабелей при несоблюдении этого требования может приводить к "необъяснимым" (но уже не для нас) сбоям.

Проверка выключателя иногда требует вскрытия системного блока (в древних ПК, возможно, корпуса блока питания), но при наличии дополнительного разъема для монитора можно установить причину, прозвонив омметром входные и выходные контакты силовых разъемов.

2. Проверка источника питания системы и внутренних соединений предполагает подачу питания при вскрытом корпусе системного блока. Основные шаги, позволяющие установить общую работоспособность системы, следующие:

- отключите системный блок, всю периферию и вскройте корпус;
- убедитесь, что кабель питания системной платы надежно вставлен в разъем (обратите внимание на цветовую маркировку проводов обеих половин кабеля);
- проверьте надежность подключения всех вертикально установленных плат расширения;
- проверьте правильность и надежность подключения кабелей питания и управления к накопителям;
- подайте напряжение на блок питания системного блока и проверьте, что система запустилась.

В том случае, если изложенные шаги не привели к ожидаемому результату, следует детализировать поиск:

- вновь отключите питание;
- демонтируйте из системной платы все карты расширения, отключите питание от накопителей (внимание: отключать от системной платы индикаторы не стоит !);
- подайте напряжение на блок питания системного блока и проверьте, что системная плата подала признаки жизни. Если это не так – неполадки либо с источником питания, либо в системной плате. Неисправности первого вида можно локализовать, отключив системную плату и попытавшись запустить блок питания с "эквивалентом нагрузки" (резисторы в несколько Ом и соответствующей мощности, подключенные к источникам +5 и +12 В, в крайнем случае, можно оставить подключенным винчестер). Если неисправность в системной плате – то увы... Методические указания здесь уже не помогут.

- отключите систему, добавьте видеокарту, а затем опять попробуйте оживить систему; продолжайте подключать карты, а затем и накопители до тех пор, пока опять не проявится отказ;

- выявив элемент, подключение которого вызывает перегрузку блока питания, отключите 1-2 ранее подключенных карты или накопителя, и попробуйте вновь запустить систему;

Если система достоверно не запускается при подключении определенного элемента при иных других сочетаниях компонентов – в данном элементе локализуется причина отказа; если система отказывает при превышении числа компонентов независимо от того, какой подключался последним – это означает, что у блока питания недостаточен запас мощности (обычно это проявляется при одновременной работе всех накопителей; при современных комплектующих бывает достаточно редко).

9.2.2. Отсутствие сигналов и сообщений об ошибках

Сообщения об ошибках не появляются при их фактическом наличии, если:

- компьютер запускается, но экран остается черным;
- система останавливается при загрузке и никаких сообщений не появляется.

В общем случае может быть рекомендована следующая последовательность шагов:

1. В первом случае следует предположить, что неисправность скрывается в системе видеоадаптер – видеомонитор. Особенно следует обратить внимание на видеомонитор. Прежде всего, следует оценить работоспособность монитора (и его подключение к системе, особенно в том случае, если при включении никаких особенных звуковых предупреждений не выдавалось), проверив состояния:

- индикатора включения;
- регулировок яркости и контрастности (и такое бывает! В любом случае свечение экрана должно меняться);
- в ЭЛТ-мониторах – наличие высокого напряжения (достаточно по наличию статического напряжения на поверхности кинескопа после его включения).

Если все эти признаки показывают, что монитор скорее всего работоспособен, а покачивание шнура в разъеме не приводит к ожидаемому результату, следует оценить работоспособность видеокарты. Однозначно говорит о ее неисправности один длинный и два коротких сигнала, издаваемых компьютером при загрузке. Иначе следует проделать следующие этапы (в современных компьютерах, видеокарты которых снабжены развитым BIOS'ом и фактически самотестируются, достаточно выполнить лишь некоторые основные):

- отключите питание и отсоедините шнур питания от системного блока;
- проверьте правильность и надежность установки платы видеоадаптера в слоте расширения;
- подключите питание; включив систему, попытайтесь войти в SETUP и, если это удалось, проверьте корректность установки типа дисплея;
- после удачной – с картинками – перезагрузки, произведите рестарт и проведите по шагам процедуру загрузки CONFIG.SYS и AUTOEXEC.BAT; (иногда изображение пропадает при "нестыковке" видеорежимов, реализуемых адаптером и поддерживаемых монитором).

2. Если либо компьютер древний, либо надежда еще не умерла, либо система предположительно зависает при загрузке, то следует выполнить следующую последовательность шагов:

- в дисковод A: вставьте системную дискету с DOS и произведите перезагрузку системы (либо используйте иные методы загрузки DOS с внешнего USB накопителя);
- через 30 с, если изображение не появилось, нажмите клавишу F1, проверьте, идет ли загрузка системы;
- дважды нажмите клавишу ENTER и введите команду DIR;
- если изображение не появилось и в этот раз, попробуйте сделать распечатку экрана SHIFT-PRINT SCREEN; если на принтере распечаталось содержимое экрана, то видеосистема неработоспособна (или несовместимы видеорежимы). Окончательный вывод можно сделать, подключив к системному блоку другой монитор.

Если и на принтере не распечаталось содержимое экрана в ожидаемом виде, то, вероятнее всего, система "зависает" на этапе загрузки по причине отказа системной платы. Отказы накопителей обычно идентифицируются соответствующими сообщениями на мониторе.

9.2.3. Сообщения об ошибках

Появление сообщений об ошибках оказывается возможным в случае хотя бы частичной работоспособности системного блока и видеомонитора. Сообщение может представляться в виде трех – четырехзначного цифрового кода, текста или звуковых сигналов (см. приложения). Наиболее значимыми, распространенными (и, иногда, устранимыми) являются ошибки класса 17XX – ошибки жесткого диска. Подробно их причины и устранение будут рассматриваться в последующих работах.

Ошибки иных компонентов, прежде всего устройств ввода-вывода, достаточно эффективно идентифицируются с помощью соответствующих диагностических программ (см.), при запуске которых следует руководствоваться полученным кодом.

2. Настройка в BIOS / UEFI отдельных подсистем

BIOS (Basic Input / Output System – базовая подсистема ввода/вывода, в настоящее время в компьютерах часто устанавливается более функциональный UEFI - Unified Extensible Firmware Interface) располагается в чипе флэш-памяти на материнской плате. Фактически, представляет собой компонент операционной системы, загружающийся в первую очередь.

Микросхема (одна или несколько) размещена с краю системной платы, обычно в непосредственной близости от «батарейки» – литиевого элемента, предназначенного для питания часов астрономического времени и микросхемы энергозависимой КМОП памяти, в которой располагаются триггеры настроек BIOS. Там же – джамперы для их сброса.

В микросхемах размещаются: программные средства и данные следующего назначения:

- Программы начального самотестирования (POST – Power On Self Test)
- Драйверы и обработчики прерываний
- Параметры текущей конфигурации ЭВМ
- Программы раскрутки ядра, актуализации данных о составе ЭВМ, идентификации, тестирования и предварительной настройки компонентов
- Программы инициализации компонентов, запуска устройств с собственным BIOS'ом, поиска и загрузки ОС

Некоторые версии BIOS / UEFI могут включать:

- o Средства обеспечения безопасности и антивирусной защиты
- o Основную ОС (Linux) или её отдельные программные компоненты
- o Автономные приложения (браузер, проводник, и т.п.)
- o Шпионские программы (?)

Во время запуска ПК процессор на материнской плате выполняет процедуру POST - программу начальной самопроверки и инициализации оборудования, после чего передаёт управление ОС.

Именно POST отвечает за появление или отсутствие сообщений об ошибках. Таким образом, если ПК не проходит через процесс загрузки, а также, если система работает слишком медленно, ОС "вылетает", "железо" сбоит, возможной причиной может являться некорректно настроенная BIOS.

Каждая материнская плата использует собственную версию BIOS, специально разработанную под её аппаратную начинку, поэтому можно привести только самое общее описание опций.

Для настройки BIOS, то есть запуска BIOS Setup, требуется перезагрузить систему и нажать определённую клавишу (или комбинацию клавиш) во время прохождения

процедуры POST. Обычно сообщение о том, какую именно клавишу или комбинацию клавиш нажать, отображается сразу после начала прохождения POST, а именно:

AMI BIOS – «DELETE»

Phoenix BIOS (FirstBIOS Pro) - «F2»

Award BIOS (FirstBIOS) - «DELETE» или комбинация клавиш «Ctrl+Alt+Esc»

Microid Research BIOS - «Esc»

Compaq - «F10»

Настройки подсистемы ввода/вывода подразделены на типичные функциональные группы, каждая из которых выполняет свой класс задач. Обычно это:

- общие параметры (Standard CMOS Setup, Main);
- свойства самой BIOS (BIOS Features Setup, Advanced);
- свойства процессора и чипсета (Chipset Features Setup, Chip Configuration, CPU Configuration);
- свойства интегрированных устройств (Integrated Peripherals, I/O Devices Configuration);
- свойства слотов PCI (PnP/PCI Configuration, PCI Configuration);
- управление питанием (Power Managment Setup, Power);
- пароли системы (Supervisor Password, User Password);
- сохранение и восстановление настроек (Save Setup, Load BIOS Default, Load Setup Defaults);
- выход и сохранение (Exit).

Следует иметь в виду, что приведенное деление достаточно условно: в каждом компьютере могут быть свои варианты распределения функций. Большая часть этих настроек интуитивно понятна, носит вспомогательный характер и не требует изменения. Однако некорректность значений некоторых может оказать решающее влияние на работоспособность компьютера.

9.2.4. Наиболее важные настройки.

1. Дисковая система

Параметры накопителей на жестких дисках, приводов компакт – и DVD-дисков, функционирование IDE-/SATA-контроллера чипсета – за все это отвечают опции, сосредоточенные в этом разделе. Смысл и настройки пунктов раздела интуитивно понятны и обычно не требуют подробных комментариев. Отметим, что выбирать некоторые режимы работы дисков необходимо до установки операционной системы: если ОС установлена, смена режима в BIOS приведет к появлению BSOD.

Стандартными опциями настроек являются:

IDE-/SATA-контроллер

Первое, на что стоит обратить внимание при рассмотрении дисковой подсистемы, – параметры IDE/SATA-контроллера чипсета. Это может быть устаревший двухканальный IDE-контроллер, переходное IDE/SATA-решение или современный вариант, поддерживающий только накопители с SATA-интерфейсом. Иногда используются дополнительные контроллеры, активизируемые в иных разделах BIOS.

Режимы обмена данными

От правильного выбора режима обмена данными с жестким диском во многом зависит скорость его работы. Использование более медленного режима отрицательно скажется на быстродействии при дисковых операциях. Установка же неподдерживаемого накопителем режима обмена данными обязательно приведет к проблемам.

Каналы контроллера

Количество каналов IDE/SATA-контроллера чипсета зависит от его поколения. Классические двухканальные IDE-контроллеры допускают подключение до 4 устройств, комбинированные IDE/SATA-решения и современные SATA-контроллеры до 8. Неиспользуемые каналы для ускорения загрузки рекомендуется отключать.

Геометрия дисков

Иногда бывает полезно напрямую указать тип подключенного накопителя и, если это жесткий диск, его характеристики (геометрию). На некоторых версиях BIOS это позволяет ускорить загрузку.

Массивы RAID

Если в компьютере установлено два и более жестких диска, многие современные чипсеты позволяют объединять их в RAID-массив (Redundant Array of Independent Disks – избыточный массив независимых дисков). Обычно существует возможность выбора различных уровней и технологий RAID.

AHCI (AHCI Advanced Host Controller Interface) Mode

Опция позволяет включить режим расширенного хост-контроллера для стандартного IDE/SATA-контроллера чипсета. Enabled позволяет воспользоваться преимуществами расширенного хост-контроллера SATA II – очередью команд, горячим подключением и т.д. Не следует забывать и о более высоком быстродействии дисковой подсистемы в этом режиме. Если жесткие диски поддерживают SATA II и наличествуют необходимые драйвера, имеет смысл остановиться именно на этом значении.

При значении Disabled для обращения к SATA-накопителям будет использоваться механизм, аналогичный обычному IDE-контроллеру. Это ограничивает быстродействие дисковой подсистемы, но не потребует установки драйверов, что актуально для старых операционных систем (семейства Windows 9x).

Мониторинг состояния

Современные жесткие диски способны самостоятельно прогнозировать возможный выход из строя, заранее предупреждая о необходимости замены. Данная технология называется S.M.A.R.T. (Self-Monitoring Analysis and Reporting Technology, технология самомониторинга анализа и предоставления отчетов).

2. Оперативная память

DDR2 SDRAM, DDR3 SDRAM и DDR4 SDRAM – самые распространенные на данный момент типы памяти. К несомненным их плюсам можно отнести поддержку автоматического конфигурирования. Параметры модуля записываются в специальную микросхему SPD, а BIOS, ориентируясь на эту информацию, автоматически устанавливает необходимые задержки. Вместе с тем, настройка вручную позволяет найти оптимум, гарантирующий отсутствие сбоев и при этом обеспечивающий максимально возможную скорость работы компьютера.

Возможными опциями настроек являются:

Контроль четности – ECC – позволяет гарантировать целостность сохраняемых данных (отсутствие в них ошибок), а если ошибка все же произошла, работа компьютера будет прекращена. Эта возможность особенно актуальна для серверов и мощных рабочих станций.

Чередование доступа – позволяет увеличить быстродействие при обращении к банкам оперативной памяти при наличии многоканального контроллера памяти.

Затенение, Кэширование – это перенос данных из более медленной памяти в оперативную содержимого ПЗУ или иной «медленной» памяти всевозможных карт расширения, имеющих свой BIOS. Поскольку доступ к оперативной памяти требует

заметно меньше времени, чем обращение непосредственно к ПЗУ с BIOS карты, то и обмен данными с такой картой происходит быстрее.

Тайминги – синоним латентности – суммарная временная задержка сигнала на участке «процессор-память», от которой зависит пропускная способность и, как следствие, быстродействие системы.

Подробнее про тайминги рассказывается в материалах лекции «Система памяти».

Задержка может складываться из трех таймингов, представляемых числом тактов, обычно в следующем порядке:

CAS Latency,
RAS to CAS Delay
RAS Precharge Delay

Если на модуле указаны четыре тайминга, то тогда это обычно

CAS Latency,
RAS to CAS Delay,
RAS Precharge Delay
Active Precharge Delay.

Соответствующие настройки BIOS могут располагаться в разделах наподобие "System Performance", "Memory Timings" или "Configure DRAM Timing". Как правило, значение по умолчанию для данных опций "By SPD" или "Enabled". Оно заставляет компьютер считывать рекомендованные значения с чипа SPD модуля памяти и автоматически их использовать. Для ручной настройки системы следует выставить значение опции в "Disabled" или "User Defined", после чего попытаться увеличить производительность системы, уменьшая с шагом в 1 выставленное по умолчанию значение.

Предупреждение: выполняйте изменение только одного параметра за один тестовый прогон. Тогда вы сразу же сможете определить причину нестабильной работы и вернуться к проверенному значению.

Задержка RAS-to-CAS (tRCD, RAS# to CAS# delay, DRAM RAS-to-CAS Delay)

Память можно представлять в виде двумерного массива. Для чтения данных следует указать столбец с помощью сигнала Row Address Strobe (RAS), а затем строку с помощью сигнала Column Address Strobe (CAS). Между сигналами RAS и CAS требуется определённый временной промежуток, чтобы адресация не сбилась.

Задержка RAS CAS (CL, CAS# Latency, DRAM CAS# Latency)

Во время получения данных из памяти между выставлением адреса и передачей данных следует выждать определённый временной промежуток. Он тоже указывается в тактах: 2T для двух тактов, 3T - для трёх и т.д. Меньшее значение "SDRAM CAS Latency" обеспечивает более высокую производительность.

Задержка RAS Precharge Delay (tRP, DRAM RAS# Precharge, RAS Precharge, SDRAM RAS Precharge, Row Precharge Time)

Чтобы ячейки памяти работали быстро, их необходимо правильно заряжать. Опция "SDRAM RAS Precharge Delay" указывает промежуток (в тактах) между зарядкой ячеек и отсылкой сигнала RAS.

Задержка SDRAM Precharge (tRAS, Active Precharge Delay, DRAM RAS# Activate to Precharge, Min RAS# Active Time).

Задержка "SDRAM Active Precharge Delay" тоже выставляется в тактах. Она указывает на задержку между последовательными доступами к памяти, так что её снижение может ускорить работу с памятью.

Дополнительные тайминги памяти

Общее число может достигать десятка. Дополнительные тайминги оказывают меньшее влияние на производительность, чем рассмотренные выше основные тайминги, поэтому их в большинстве случаев следует оставить по умолчанию.

QtRRD (RAS to RAS Delay) - задержка между активизацией строк разных банков.

tRC (Row Cycle Time), Row Active Time, Row Pulse Width - длительность цикла строки памяти. Полный цикл состоит из времени от начала активизации строки до ее закрытия (tRAS) и задержки для формирования нового сигнала RAS# (tRP), т. е. $tRC = tRAS + tRP$.

tWR (Write Recovery Time) - задержка между завершением операции записи и началом предзаряда.

GtWTR (Write to Read Delay) - задержка между завершением операции записи и началом операции чтения.

QtRTP (Precharge Time) - интервал между командами чтения и предварительного заряда.

tREF (Refresh period) - частота обновления памяти. Может устанавливаться в тактах или микросекундах.

tRFC (ROW Refresh Cycle Time) - минимальное время между командой обновления строки (Refresh) и командой активизации или другой командой обновления. В некоторых версиях BIOS имеется возможность установки данного тайминга для каждого модуля памяти, а параметры будут называться соответственно Trf с 0/1/2/3 for DIMM 0/1/2/3.

Напряжение питания модулей памяти

Если память работает быстрее, то ей необходимо обеспечить большую мощность. Именно поэтому вместе с ростом частоты обычно повышают и напряжение питания.

Опция "DDR Reference Voltage" позволяет изменять напряжение модулей DDR, обычно с шагом 0,1 В. Увеличение напряжения имеет смысл, если были снижены задержки или увеличена частота работы памяти, либо начали возникать проблемы со стабильной работой.

9.2.5. Диагностические программы

Существует целый ряд программ, обеспечивающих снятие конкретных характеристик различных периферийных устройств. Рационально использование диагностических программ, входящих, например, в состав пакетов CHECKIT, QAPLUSFE и т. п. Их преимуществом является, во-первых, достаточная универсальность, инвариантность к конкретной конфигурации диагностируемой системы и минимальные требования к системному ядру, во-вторых, высокая скорость проведения и достоверность процедуры, в-третьих, определенная гарантия от нештатных воздействий на работоспособные компоненты системы.

В рамках данной работы предлагается практически познакомиться с некоторыми подобными пакетами, применение которых может быть рекомендовано даже для исправных систем в целях определения их фактических параметров и, например, для оценки текущей производительности при настройке BIOS.

9.3. Задание к лабораторной работе

В ходе выполнения лабораторной работы следует обратить внимание на следующие вопросы, связанные с настройкой, поиском и устранением неисправностей основных компонентов системного блока, а именно:

1. Поиск и устранение неисправностей
 - 1.1. Выявление проблем системы электропитания
 - 1.2. Оценка работоспособности при отсутствии сигналов и сообщений об ошибках
 - 1.3. Диагностика неисправностей на основе сообщений об ошибках
2. Настройка в BIOS / UEFI основных подсистем:
 - 2.1. Дисковая система
 - 2.2. Оперативная память
 - 2.3. Процессор
3. Углублённое тестирование системы с использованием специальных диагностических программ.

9.4. Методические указания и порядок выполнения работы

9.4.1. Ход работы

1. Ознакомьтесь с методическими указаниями.
 2. Ознакомьтесь с подключением внутренних компонентов системного блока, особо отмечая подключение и цветовую маркировку кабелей; размещение контроллеров в разъемах системной платы; зарисуйте ориентацию основных компонентов.
 3. Проверьте работоспособность компьютера, руководствуясь, при необходимости, изложенной выше методикой поиска неисправностей.
 4. Изучите функции BIOS/UEFI системной платы, при необходимости уточнив их с помощью Google. Обратите внимание на возможность выбора внешнего USB накопителя в качестве загрузочного устройства
 - 5 *). Выставьте необходимые уставки и проведите тестирование собранного вами ПК с помощью предложенных диагностических программ. Оцените его производительность с помощью тестов, сравните результаты при различных уставках SETUP'a.
 - 6 *). Напишите для пакета QAPLUSFE командный файл для проведения максимально полного автоматического тестирования ПК имеющейся конфигурации и создания отчета. Подсказка: пользуйтесь опцией SCRIPT и клавишей F1.
- *) – Пункты выполняются на лабораторных компьютерах..

9.5. Требования к отчету и защите

1. Ход работы.
2. Примерный перечень неисправностей, препятствующих началу работы стандартного компьютера и выходу на этап настройки BIOS. Возможные причины и их индикация.
3. Общее описание BIOS SETUP конкретного компьютера с указанием, какими особенностями платы вызвана необходимость его наличия (наличие на плате... указывает на возможность настройки в соответствующем разделе BIOS SETUP).
4. Указание на значения (диапазон) настраиваемых параметров и причины возможных изменений.
5. Порядок выбора внешнего USB накопителя в качестве загрузочного устройства.

6. Справочно – какие параметры и в каких целях могут настраиваться у других компьютеров.

7. Детальное описание тонкой настройки отдельных разделов BIOS (по материалам из интернета):

- дисковая система,
- процессор,
- система памяти (тайминги).

8. Черновой набросок порядка этапов и шагов начала раскрутки и загрузки компьютера (в какой последовательности в ходе раскрутки и загрузки происходит проверка компонентов и использование внесённых в BIOS SETUP изменений?).

9. А какие компоненты, востребованные до момента полной загрузки ОС могут подключаться в процессе раскрутки к ядру, что потребует дальнейшего продолжения процедуры тестирования?

10. В каком предположительном порядке происходит их подключение, какие характеристики / параметры должны быть определены, и в какой последовательности?

10. ЛАБОРАТОРНАЯ РАБОТА № 9. ИССЛЕДОВАНИЕ РЕСУРСОВ СИСТЕМНОГО ЯДРА ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА

10.1. Общие сведения

Цель:

Знакомство с организацией и выполнением начальных этапов загрузки компьютера, практическое освоение методов использования стандартного системного программного обеспечения для проведения диагностирования и настройки компонентов персонального компьютера.

Материалы, оборудование, программное обеспечение:

Комплект плат и устройств IBM PC-совместимого персонального компьютера: системная плата, видеосистема, клавиатура, мышь, блок питания, загрузочное устройство (флоппи-дискетод или жесткий диск), загрузочная дискета (диск, USB flash), программа-отладчик DEBUG.

Условия допуска к выполнению:

Изучение правил техники безопасности, предварительное знакомство с УМП, отсутствие существенных текущих задолженностей по лабораторному практикуму.

Критерии положительной оценки:

Выполнение типового задания, оформление отчета по работе, ответы на вопросы для самопроверки.

Планируемое время выполнения:

Аудиторное время выполнения (под руководством преподавателя): 4 ч.

Время самостоятельной подготовки: 4 ч.

10.2. Теоретическое введение

Литература:

Горнец, Н.Н. ЭВМ и периферийные устройства. Компьютерные и вычислительные системы : учеб. / Н. Н. Горнец, А. Г. Рощин . - Москва : Академия, 2012. - 235 с. Стр. 169-198.

DEBUG. Быстрый старт

http://stilus-doctus.narod.ru/debug/debug_a.html

Использование DEBUG для проверки оборудования

<https://forum.sources.ru/index.php?showtopic=214452>

Процедура POST

<https://www.ixbt.com/peripheral/post-cards.html>

Контрольные вопросы для самопроверки:

1. Что такое прерывания INT, как их использовать в ходе тестирования оборудования
2. Какое назначение и какие конкретные действия выполняют прерывание INT 20h ?
3. Какое назначение и какие конкретные действия выполняют прерывание INT 11h ?
4. Какое назначение и какие конкретные действия выполняют прерывание INT 16h ?
5. Какое назначение и какие конкретные действия выполняют прерывание INT 19h ?
6. Какое назначение и какие конкретные действия выполняют прерывание INT 18h ?
7. Какое назначение и какие конкретные действия выполняют прерывание INT 13h и её функция 42h ?
8. Предложите алгоритм тестирования ПЗУ, ОЗУ, процессора.

9. Чем определяется последовательность проверки устройств?
10. Что должно произойти, если устройство обнаружено, но не прошло тест?
11. Какие устройства должны обязательно присутствовать в составе компьютера?
12. Какие параметры, каких именно устройств должны быть известны ОС к моменту ее запуска?

10.2.1. Основные сведения

Современный ПК представляет собой аппаратно-программную систему, для которой характерна относительная независимость аппаратного и прикладного программного обеспечения. Их взаимодействие обеспечивается ОС, «нижний» уровень которой, непосредственно управляющий аппаратными средствами, представлен базовой системой ввода/вывода, (BIOS) – набором программ, хранящимся на системной плате в микросхеме ПЗУ с соответствующим именем.

Все программы, размещённые в микросхеме BIOS, подразделяются на группы в соответствии с их функциями:

- выполнения собственно функций ввода-вывода (драйверов) до момента загрузки драйверов операционной системы, прежде всего – для связи с манипуляторами, вывода информации на экран и взаимодействия с дисковыми накопителями, а также для обслуживания аппаратных прерываний (например, от клавиатуры и таймера) и обработки программных прерываний BIOS, предназначенных для управления обменом данными между программными средствами компьютера и подключенными к нему периферийными устройствами;
- настройки и конфигурирования узлов системной платы и устройств, подключенных к ней, что выполняется с помощью программы BIOS Setup;
- инициализации и начального тестирования всех основных (стандартных) узлов компьютера – расположенных на системной плате, подключенных к интерфейсам, а также установленных в слотах расширения. Для этого используется программа POST (Power On Self Test). Отметим, что некоторые устройства, драйвера которых устанавливаются после загрузки ОС, могут не проверяться;
- загрузки операционной системы с внешнего устройства – винчестера, флешки или ПЗУ сетевой карты. В некоторых BIOS существует возможность загрузки операционной системы, хранящейся непосредственно в ней.

Загрузке операционной системы предшествует этап самотестирования системного ядра (процедура POST), в ходе которого определяется его целостность, работоспособность и, в ходе «раскрутки», выявляются его аппаратные ресурсы – состав и параметры подключаемых элементов (некоторые значения задаются настройками BIOS SETUP).

При необходимости на этом этапе подключаются дополнительные модули расширения BIOS.

Собственными BIOS могут обладать некоторые устройства с набором функций, дополняющих системную плату (например, сетевая карта с BootROM, контроллер дисков, видеоадаптер). Как правило, наличие такого BIOS обусловлено необходимостью наличия специальных драйверов непосредственно на этапе запуска системы.

В таких случаях основная BIOS после начальной самопроверки пытается обнаружить подобные устройства, включает хранящиеся в их ROM подпрограммы в состав этапов тестирования и загрузки и при необходимости открывает их экраны настройки.

Экран «дополнительного» BIOS'а обычно становится доступным при необходимости контроля и настройки соответствующего устройства. В иных случаях, а

также при наличии интегрированных устройств, управление осуществляется посредством клавиатурных команд (например, после обнаружения устройство появляется в BOOT Menu) или через меню BIOS системной платы:

При наличии в системе подобных устройств BIOS системного ядра будет активироваться полностью в последнюю очередь.

Стандартная информация, предоставляемая пользователю по итогам тестирования, включает перечень, основные параметры и текущее состояние портов и обнаруженных устройств.

Отметим, что современные компьютеры, поддерживающие режим Fast Boot, как правило, не выводят на экран подобную информацию.

Стандартный цикл самотестирования системной платы обычно включает следующую последовательность шагов:

1. Проверка регистров процессора.
2. Проверка контрольной суммы BIOS.
3. Проверка таймера.
4. Проверка контроллеров DMA.
5. Проверка регенерации памяти и тестирование первых 64 Кбайт.
6. Проверка интерфейса клавиатуры.
7. Инициализация контроллера прерываний и установка векторов.
8. Проверка батареи и контрольной суммы CMOS BIOS.
9. Проверка защищенного режима.
10. BIOS SETUP – обновление по запросу пользователя информации в CMOS.
11. Получение настроек конфигурации из CMOS, коррекция последующих шагов теста.
12. Проверка видеоадаптера.
13. Проверка контроллера прерываний.
14. Проверка клавиатуры.
15. Тест памяти от 64 до 640 Кбайт.
16. Тест памяти свыше 1 Мбайт.
17. Проверка загрузочных устройств.
18. Поиск дополнительных модулей ROM BIOS и их инициализация.
19. Вызов загрузчика операционной системы (INT 19h, Bootstrap), при невозможности загрузки операционной системы – попытка запуска альтернативного загрузчика (INT 18h); при неудаче – останов системы (HALT).

Каждый из шагов реализован соответствующей подпрограммой тестирования, записанной в виде двоичных кодов непосредственно в ПЗУ BIOS (системной платы и устройств) и выполняемой процессором при передаче управления на её первую команду.

При тестировании учитывается специфика организации и работы проверяемого элемента:

- обязательные элементы ядра проверяются в ходе выполнения ими стандартных операций (прямое тестирование). При ошибке или отказе какого-либо диагностируемого элемента дальнейшая проверка останавливается, при возможности, выдаётся сообщение либо на экран либо на динамик.

- подключаемые необязательные элементы, обычно приспособленные к самотестированию, опрашиваются на заключительных шагах раскрутки. При их обнаружении определяется их состояние, при необходимости – выводятся на экран их параметры, выполняется инициализация и настройка.

Перед началом каждого из тестов процедура POST генерирует так называемый POST код, который выводится по определенному адресу в пространстве адресов

устройств ввода/вывода компьютера. В случае обнаружения неисправности в тестируемом устройстве процедура POST просто «зависает», а предварительно выведенный POST код однозначно определяет, на каком из тестов произошло «зависание».

Адреса портов для вывода POST кодов зависят от типа компьютера, а собственно таблицы POST кодов различны для различных производителей BIOS.

Для отображения POST кодов в удобном для человека виде служат устройства под названием POST Card.

Хотя ход и результаты процедуры POST способны предоставить полезную информацию и пользователю, непосредственное обращение к хранящимся в ПЗУ подпрограммам обычно затруднено и чревато непредсказуемыми последствиями.

Познакомиться с проведением POST и оценить аппаратные ресурсы компьютера можно, реализовав аналогичные процедуры с использованием программы-отладчика, запускаемой в среде операционной системы.:

10.3. Задание к лабораторной работе

1. Ознакомьтесь с методическими указаниями. Попытайтесь понять назначение различных команд отладчика.
2. Создайте на диске собственный рабочий каталог и скопируйте в него файл DEBUG.COM.
3. Запустите с дискеты или из созданного каталога отладчик и все дальнейшие задания выполняйте с его помощью.

10.4. Методические указания и порядок выполнения работы

DEBUG является отладчиком программ на уровне мнемонических ассемблерных кодов. Этот уровень представляет наибольший интерес для системных программистов и специалистов по техническому обслуживанию ЭВМ.

Так как данный отладчик – это одна из утилит DOS, для его запуска достаточно в командной строке DOS набрать его имя – DEBUG, указав имя редактируемого исполнимого файла с расширением .COM, например:

```
C:\DEBUG NAME.COM
```

Если поле NAME.COM (или иное, содержащее имя редактируемого файла) отсутствует, DEBUG позволит создать новый исполнимый файл.

После загрузки DEBUG все сегментные регистры процессора CS, DS, ES, SS устанавливаются на начало свободной памяти, то есть на меньший сегментный адрес за концом программы отладчика. Указатель команды IP получает значение 100H, а остальные регистры (AX, BX, CX, DX, SI, DI, BP) устанавливаются в 00H. Если, однако, в команде вызова DEBUG было указано имя файла, то регистр CX будет содержать длину файла, причем если длина превысит 64 К, старшие разряды числа будут размещены в регистре BX.

10.4.1. Работа с отладчиком и его основные команды

Подсказкой и приглашением ко вводу команды DEBUG является дефис - , выводимый компьютером на экран. Ввод своей команды пользователь подтверждает нажатием клавиши ENTER.

Формат команды
Комментарии к выполнению

- A [XXXX] Ассемблирование, начиная с адреса XXXX, или с текущего адреса; позволяет транслировать прямо в память с адреса XXXX инструкции языка Ассемблер. Значение XXXX по умолчанию принимается равным 0100.

Примечание: Адреса в DEBUG задаются и выводятся в формате СЕГМЕНТ:СМЕЩЕНИЕ, например, для 0175:0000 физический адрес имеет значение 1750. После ввода команды "A" указанный адрес отображается в строке экрана, в которую можно вводить команду Ассемблера в стандартном памяти виде. Ввод команды должен быть завершен нажатием клавиши ENTER, после чего на экран выводится следующий адрес, и так далее. Для выхода из режима ассемблирования в режим собственных команд отладчика клавиша ENTER должна быть нажата дважды.

- D XXXX [LYYY] Показ содержимого памяти с адреса XXXX длиной YYY. Без указания параметров выводит 128 байт текущего сегмента.

- U [XXXX] Дизассемблирование с адреса XXXX. Транслирует содержимое памяти в ассемблерных кодах. Длина дизассемблируемой области памяти определяется объемом экрана. Для последовательного просмотра всей программы можно вводить команду U, опуская адрес.

- G[[=XXXX]YYYY] Запуск программы на выполнение. Начало исполняемой программы располагается по адресу XXXX. По умолчанию, если =XXXX опущено, используются текущие значения CS и IP. Адрес YYYY определяет точку прекращения выполнения программы. Если значение YYYY опущено, программа исполняется полностью.

- T [=XXXX YYY] Трассировка программы, начиная с адреса XXXX. Выполняется трассировка YYY команд. После выполнения каждой команды на экран выводится содержимое регистров и значения флагов (в формате, аналогичном используемому в команде R), а также адрес и мнемоника следующей команды. Ввод - T без параметров дает выполнение одного шага.

- P [=XXXX YYY] Аналогичная указанной выше команда, не производящая, в отличие от нее, трассировки подпрограмм.

- R [XX] Команда позволяет отображать и изменять содержимое регистров (XX) и флагов процессора. По команде R без параметров на экран выводится содержимое всех регистров, а также регистра флагов, адрес и коды следующей (еще не выполненной) команды. Состояние регистра флагов отображается мнемонически:

OV/NV - переполнение (есть/нет) - разряд OF
DN/UP - направление (ниже/выше) - разряд DF
EI/DI - прерывания (разр./запр.) - разряд IF
NG/PL - знак (минус/плюс) - разряд SF
ZR/NZ - нуль (да/нет) - разряд ZF
AC/NA - Доп. перенос (да/нет) - разряд AF
PE/PO - Паритет (чет/нечет) - разряд PF
CY/NC - Перенос (да/нет) - разряд CF

Для того, чтобы изменить значения соответствующих флагов, надо ввести необходимые обозначения после команды - RF в той же строке.

- M XXXX LYYY ZZZZ Перемещение области памяти, начиная с адреса XXXX длиной YYYY на новое место, начиная с адреса ZZZZ.

- E XXXX[список] Замена содержимого памяти, начиная с адреса XXXX. [Список] может быть задан в шестнадцатеричных кодах и ASCII. В последнем случае [список] заключается в кавычки. Если команда - E введена без списка, то выводится содержимое ячейки памяти, расположенной по адресу XXXX и точка, являющаяся приглашением к его замене. Если после этого нажать клавишу "пробел", то можно перейти к следующему адресу, а нажав клавишу "-" - вернуться на шаг ранее.

- F XXXX[область][список] Команда заполнения области памяти, ограниченной либо в формате YYYY, либо LYYY, данными, указанными в [списке].
- C XXXX LYYY ZZZZ Сравнение содержимого областей памяти длиной YYY, расположенных по адресам XXXX и ZZZZ.
- N [пусть][имя файла] Определяет путь и имя файла, который будет записан на диск по команде W или считан с диска по команде L.
- L XXXX Загрузка файла, определенного командой N, в память, начиная с адреса XXXX.
- L XXXX [диск][сектор]
[количество секторов] Аналогична ранее рассмотренной, загружает в память указанные сектора диска. По умолчанию XXXX принимается равным CS:0100.
- W XXXX Команда записи на диск в файл, идентифицированный командой N, данных из памяти, начиная с адреса XXXX. Длину файла указывают регистры CX и BX.
- W XXXX [диск][сектор]
[количество секторов] Запись данных из памяти непосредственно на диск в указанный сектор.
- S XXXX LYYY [строка] Команда поиска, отыскивающая наличие указанной цепочки знаков в области, определенной начальным адресом XXXX и YYY.
- I [порт] Выводит содержимое порта на экран.
- O [порт] XX Записывает байт XX в указанный порт.
- Q Выход из отладчика в среду MS DOS.

10.4.2. Ход работы

1. Выберите подходящую команду и просмотрите с ее помощью ПЗУ по адресу: F000:FFF5 - для определения даты версии BIOS и по адресу F000:8000 - для определения авторских прав на него.

2. Введите, используя ассемблер, программу проверки регистров МП (XXXX – выбранный вами диапазон адресов в шестнадцатеричном коде):

```
- A 100
XXXX:0100 MOV AX, 1
XXXX:0103 MOV BX, AX
XXXX:0105 CMP BX, AX
XXXX:0107 JNZ 010F
XXXX:0109 ADD AX, AX
XXXX:010B JNC 0103
XXXX:010D JMP 0110
XXXX:010F NOP
XXXX:0110 INT 20
XXXX:0112
```

Прокомментируйте команды программы. Проверьте ее выполнение по шагам с использованием команд трассировки -N и -P. Прокомментируйте действия.

3. Создайте и сохраните полученную программу в своем каталоге на диске C: под именем, например, test.com:

- N C: \...\test.com - задание имени рабочего диска, пути к рабочему каталогу и имени создаваемого файла
- R BX - обращение к регистру BX
- XXXX - текущее содержимое регистра

: 00	- задание нового содержимого (старшей части длины записываемого файла)
- R CX	- обращение к регистру CX
XXXX	- текущее содержимое регистра
: 0012	- младшая часть байтов длины записываемого файла; определяется как разность смещения: 0112 0100=0012
- W	- команда записи файла на диск

4. Введите, используя ассемблерные коды, программу звучания динамика:

```

- A100
..... IN AX, 61
..... OR AX, 03
..... OUT 61, AX
..... MOV CX, FFFF
.... * LOOP *      Вместо * следует вставить адрес
..... IN AX, 61
..... AND AX, FC
..... OUT 61, AX
..... INT 20

```

Прокомментируйте команды программы.

5. По адресу F000:FFFE в ROM BIOS находится признак типа машины. Например: FF - PC; FE - XT; FC - AT.

- определите тип компьютера командой - D;
- введите и разберите программу:

```

DB 0
PUSH DS
MOV AX, F000
MOV ES, AX
MOV DS, AX
MOV AL, [FFFE]
POP DS
MOV [100], AL
INT 20

```

Просмотрите содержимое регистра DS с помощью команды D.

6. ATX-совместимые компьютеры хранят информацию о периферии в регистре 14 CMOS-микросхемы конфигурации. При этом применяется следующее кодирование информации:

- 7-6 биты: количество НГМД: 00 - 1 НГМД, 01 - 2 НГМД;
- 5-4 биты: тип монитора: MDA/CGA40/CGA80/none (то есть иные);
- 3-2 биты: не используются;
- 1 бит: наличие (1) сопроцессора;
- 0 бит: наличие (1) НГМД.

Введите программу:

```
MOV AL, 14
OUT 70, AL
IN AL, 71
INT 20
```

Просмотрите содержимое регистра AX с помощью команды - RAX.

10.5. Требования к отчету и защите

1. Описание хода выполнения лабораторной работы.
2. Тексты тестов из методички и блок-схемы алгоритмов тестирования
3. Результаты исполнения тестов.
4. Письменно ответить на вопросы:

В какой момент на этапе загрузки компьютера выполняется самотестирование? Почему некоторые тесты POST не проходят, а после загрузки ОС вполне выполняются аналогичные программы из папки TESTS? А может ли быть такое, что не обнаруживаемое в процедуре POST, неработоспособное под DOS'ом устройство будет работать под Windows? А как можно обеспечить обнаружение нового устройства средствами BIOS?

Что такое прерывания INT (и в чём их практическая польза при тестировании)?

- Какое назначение и какие конкретные действия выполняют прерывания:
 - a. INT 20h ?
 - b. INT 11h ?
 - c. INT 16h ?
 - d. INT 19h ?
 - e. INT 18h ?
 - f. INT 13h и её функция 42h ?

11. ЛАБОРАТОРНАЯ РАБОТА № 10. УСТАНОВКА И ЗАГРУЗКА ОПЕРАЦИОННОЙ СИСТЕМЫ

11.1. Общие сведения

Цель:

Изучение процесса загрузки ОС, особенностей и проблем запуска системы. Практическое знакомство с методами установки операционной системы Windows на жесткий диск компьютера и её восстановления при ошибках загрузки.

Материалы, оборудование, программное обеспечение:

Комплект плат и устройств IBM PC-совместимого персонального компьютера, инструменты, CD/DVD привод, загрузочный и установочный диск, (виртуальный привод и образы дисков).

Условия допуска к выполнению:

Изучение правил техники безопасности, предварительное знакомство с УМП, отсутствие существенных текущих задолженностей по лабораторному практикуму.

Критерии положительной оценки:

Выполнение типового задания, оформление отчета по работе, ответы на вопросы для самопроверки.

Планируемое время выполнения:

Аудиторное время выполнения (под руководством преподавателя): 4 ч.

Время самостоятельной подготовки: 4 ч.

11.2. Теоретическое введение

Литература:

Горнец, Н.Н. ЭВМ и периферийные устройства. Компьютерные и вычислительные системы : учеб. / Н. Н. Горнец, А. Г. Рощин . - Москва : Академия, 2012. - 235 с. Стр. 169-198.

Загрузчики и установка Windows:

<https://hww.ru/2017/07/vse-cto-vy-xoteli-znat-pro-zagruzku-windows-zagruzchiki-i-vostanovlenie-zagruzki-os/>

<https://phystechsoft.ru/tekhnicheskaya-dokumentatsiya>

<http://www.codenet.ru/progr/dos/>

Контрольные вопросы для самопроверки:

1. В чем конкретно заключаются преимущества и проблемы установки из дистрибутива?
2. С какой целью установка проводится в несколько этапов? Какие именно это этапы?
3. Можно ли совместить или исключить некоторые этапы установки, к чему это может привести?
4. Каким образом можно радикально ускорить инсталляцию ОС? В каком случае, каковы риски?
5. Что из себя должен представлять «образ жесткого диска»?

6. Каков состав загрузочных записей и системных файлов, используемых при загрузке операционной системы, в какой последовательности передаётся управление между ними для следующих ОС:

- MS DOS, Windows 9x;
- Windows XP;
- Windows 7 - Windows 10?

7. С помощью каких утилит (менеджеров загрузки) можно редактировать меню различных ОС

8. Как обеспечивается загрузка ОС средствами NTLDR, BootMGR, LILO, GRUB, PLOP? А как загружаются образы ISO? А как вирусы?

9. В чём сходство и чем различаются назначение и функционал загрузочной записи и менеджера загрузки?

10. Какими средствами удобнее и проще обеспечить мультизагрузку? А в чём тогда заключаются преимущества иных загрузчиков, и есть ли они?

11. Какие, помимо BootIce и EasyBCD, существуют утилиты для работы с загрузочными записями различных ОС?

12. Какие ещё программы используются для создания загрузочных / установочных флешек? В чём состоит их специфика?

13. В чём состоит отличие загрузки из образа ISO от загрузки из образа VHD? Как они реализованы на практике? В чём проявляются различия?

14. Как создаются образы VHD, какие они бывают (дочерние, дифференциальные, динамические ...), для каких целей создаются?

15. В каких случаях может быть целесообразно применение VHD непосредственно для загрузки ОС? В каких случаях – для использования с виртуальными машинами?

11.2.1. Основные сведения

Загрузка операционной системы в общем случае происходит в несколько этапов. На каждом из них возможно возникновение проблем, связанных с ошибками, допущенными при установке ОС.

На первом этапе средствами BIOS/UEFI выполняются необходимые подготовительные операции собственно компьютера.

Загрузке операционной системы предшествует тестирование основных узлов компьютера — расположенных на системной плате, вставленных в слоты расширения компьютера, а также подключенных к интерфейсам, поддерживающим загрузку. В ходе процедуры POST (Power On Self Test) BIOS проверяет корректность их работы и инициализирует работоспособные устройства.

Далее BIOS последовательно опрашивает устройства, отмеченные в заранее созданном списке BIOS Setup, пока не найдёт загрузочное. Это может быть дискета, жёсткий диск, сетевой ресурс, встроенное ПЗУ или какое-либо иное (алгоритм выбора загрузочного устройства может быть различным и зависит от реализации BIOS). Стандартно для ПК наивысшим приоритетом обладает диск A:, далее – жёсткие диски.

При использовании стандартного метода BIOS-MBR, если загрузочное устройство обнаружится, с него должен быть считан начальный загрузчик (запись, расположенная в секторе 1, головки 0, цилиндра 0), который должен быть занесён в оперативную память по адресу 0000h:7C00h. В случае если устройство содержит несколько разделов, каждый из которых потенциально может быть загрузочным (как, например, в случае жёстких дисков), должна быть считана и таблица разделов.

Специфика загрузки ОС с использованием метода UEFI-GPT рассматривается в <https://habr.com/ru/post/327572/>

Если загрузочное устройство не обнаружено, либо загрузочный сектор не может быть прочитан, или содержит ошибки, процесс загрузки останавливается.

Для корректного выполнения первого этапа загрузки необходимо выполнить соответствующую настройку BIOS Setup, указать состав и порядок загрузочных устройств и зафиксировать выбранные значения в CMOS памяти системной платы.

На втором этапе запускается начальный загрузчик ОС, к этому моменту расположенный в памяти по адресу 0000h:7C00h. Его задачей является сканирование таблицы разделов, поиск активного раздела и загрузчика операционной системы

В случае использования жесткого диска начальный загрузчик называется главной загрузочной записью (master boot record – MBR) и может не зависеть от ОС.

Типичная процедура загрузки компьютера IBM/PC совместимой архитектуры с жесткого диска выполняется следующим образом:

- BIOS считывает 512 байт первого сектора диска, в котором располагается MBR в ОЗУ по адресу 0x00007C00 (0x07C0:0x0000 в формате реального режима). MBR содержит небольшой фрагмент исполняемого кода, таблицу разделов (partition table) и специальную сигнатуру («индикатор операционной системы»). Затем управление передается прочитанному коду.

- Исполняемая далее программа читает и анализирует таблицу разделов жесткого диска (обычно ищется активный раздел, с которого загружается загрузочная запись (Partition Boot Record - PBR)) и передает ему управление.

- Если активного раздела не существует, или загрузочный сектор активного раздела некорректен, MBR может загрузить резервный начальный загрузчик и передать управление ему. Резервный начальный загрузчик должен выбрать раздел (зачастую с помощью пользователя), загрузить его загрузочный сектор и передать ему управление.

Для корректного выполнения второго этапа загрузки необходимо соответствующим образом подготовить загрузочное устройство, создав разделы, выполнив форматирование и разместив MBR, поддерживающий загрузчик ОС. Первый сектор диска (или что-то аналогичное у иных загрузочных устройств) должен быть гарантированно свободен от ошибок!

На третьем этапе загрузчик ОС должен загрузить в память ядро операционной системы и передать ему управление.

Следует различать загрузочные записи (PBR, загрузочные области) и загрузчики (менеджеры загрузки).

Загрузочные записи располагаются в недоступной для проводника ОС области диска (обычно начиная с сектора 63 начального раздела диска) и содержат исполняемый код для процессора, который не является собственно файлом, а содержит указание на файл менеджера загрузки.:

Для корректного выполнения третьего этапа загрузки следует подготовить диск, разместив загрузочную запись PBR и скопировав в корневую директорию файл (файлы) загрузчика – NTLDR, BOOTMGR или GRUB и сконфигурированные соответствующие им файлы меню.

На четвертом этапе происходит загрузка операционной системы. Предварительные настройки могут производиться с помощью пунктов меню загрузчика ОС (для Windows на ядре NT5 файлом меню конфигурации загрузки является текстовый файл BOOT.INI, начиная с ядра Windows NT6 используется бинарный файл данных конфигурации загрузки (BCD), загрузчик GRUB руководствуется файлом MENU.LST). Загрузчик ОС выполняет, в соответствии с меню конфигурации загрузки, поиск и запуск файлов установленной системы.

Для корректного выполнения четвертого этапа загрузки на диске следует разместить файлы ОС, включая все необходимые для загрузки драйвера.

Чаще всего этапы 2-4 выполняются в ходе единого цикла, характерного для традиционной процедуры установки ОС с дистрибутива, поставляемого на CD или DVD дисках. Вместе с тем, понимание процессов, протекающих на «дооперационных» этапах и умение вручную выполнять определенные действия, позволят как избежать некоторых ошибок, так и упростить и ускорить установку и восстановление ОС.

Далее рассмотрим различные способы выполнения этапов 2-4 установки ОС

11.3. Задание к лабораторной работе

1. Ознакомьтесь с методическими указаниями, выполните последовательно пункты 11.4.1. – 11.4.4. методических указаний.
2. Аккуратно оформите отчет в электронной форме.

11.4. Методические указания и порядок выполнения работы

11.4.1. Установка ОС с дистрибутива на CD или DVD

Данный способ является наиболее распространенным, особенно при установке лицензионных копий ОС.

Преимуществом является минимальное вмешательство оператора в процесс установки и оптимальная информативность процесса (Во многой мудрости многие печали; и кто умножает познания, умножает скорбь © Соломон).

Установка выполняется в несколько шагов:

1. Вставляем в устройство для чтения дисков (Windows 7 устанавливается только с DVD!) дистрибутив ОС и перегружаем компьютер. На экране должна появиться надпись: «Press any key...». Находим на клавиатуре и жмем any key.
2. Появляется окно установки Windows, в котором следует выбрать необходимые языковые параметры или оставить параметры по умолчанию. Жмем кнопку «Далее».
3. Теперь должен появиться экран запуска установки. Жмем кнопку «Установить» и переходим к окну выбора версии операционной системы. Выбираем нужную Вам версию и жмем кнопку «Далее».
4. Соглашаемся с лицензионным соглашением.
5. Выбираем пункт «Полная установка», после чего появится окно выбора и управления разделами жесткого диска. Выбираем нужный раздел и снова жмем кнопку «Далее». (Также можно нажать на «Настройка диска» и выполнить необходимые подготовительные действия, такие, как изменение раздела и форматирование. Все же удобнее выполнить подготовку заранее из-под уже установленной ОС или LiveCD. Не рекомендуется устанавливать ОС на существующий раздел, если там уже присутствует другая операционная система Windows, так как это может привести к перезаписи или пропуску файлов!).
6. Ждем, пока установится операционная система. После установки перед нами появится окно ввода имени пользователя и имени компьютера. Вводим требуемое, жмем «Далее» и видим экран установки пароля учетной записи. Вводим или оставляем поля пустыми и жмем «Далее».
7. Теперь должно появиться окно ввода ключа продукта данной копии Windows. Вводим ключ (или оставляем поле ввода пустым), жмем кнопку «Далее».
8. Указываем необходимые настройки безопасности, часовой пояс и тип сети (если она обнаружилась).

9. Далее доустанавливаем необходимые драйверы.

Несмотря на распространенность, способ достаточно продолжителен, ограничивает выбор пользователя и нереализуем при отсутствии соответствующих приводов. Наличие в дистрибутиве полного пакета совместимых с оборудованием ядра драйверов должно гарантировать успешность хотя бы первоначального старта установленной ОС.

11.4.2. Установка ОС с образа диска

В случаях, когда необходимо установить копии ОС на несколько идентичных компьютеров, способ 2.1. является крайне неудобным. Оптимальным представляется разовая установка и настройка ОС на первом компьютере, после чего его жесткий диск или разделы клонируются и переносятся на диски остальных компьютеров. Этот подход также позволяет хранить созданные ранее образы для восстановления системы и программного обеспечения после любых сбоев или для переноса уже установленного программного обеспечения на другое «железо». Преимуществом является возможность одновременного переноса MBR и PBR, что не требует выполнения предварительной подготовки целевых дисков.

Существует несколько программных пакетов для клонирования жестких дисков, к наиболее актуальным из которых можно отнести продукты фирмы Acronis. Программа Acronis True Image позволяет создавать различные варианты образов дисков с установленными ОС и прикладными программами, а также восстановление (в том числе - выборочное) из образов. Также существует технология Acronis Universal Restore, которая позволяет осуществлять восстановление образа на отличающееся оборудование.

Тем самым, при наличии подготовленного образа ОС, ее установка сводится к нескольким простым шагам:

1. Подключите к компьютеру накопитель с клоном жесткого диска и загрузочный накопитель USB или привод CD/DVD с загрузочным диском Acronis, и перезагрузите его.
2. В загрузочном меню выберите True Image (или что-то подобное) и нажмите на клавиатуре «Enter».
3. На стартовой странице выберите восстановление, и щелкните на ссылке «Диски».
4. Нажмите кнопку «Обзор», в открывшемся окне укажите путь к файлу образа с расширением *.tib на внешнем накопителе и нажмите кнопку «ОК».
5. Выберите «Восстановить диски или разделы», поставив галочку «использовать Acronis Universal Restore» (опция присутствует в версии Plus Pack), и нажмите кнопку «Далее».
6. При необходимости доустановки драйверов (если образ устанавливается на компьютер с иной архитектурой) – нажмите кнопку «Добавить путь поиска» и в открывшемся окне укажите путь к папке с драйверами, для продолжения нажмите кнопку «ОК».
7. Выберите раздел C: поставив галочку. Пункт MBR оставьте пустым, для продолжения нажмите кнопку «Далее» (Как поступить: если на диске ранее уже была установлена аналогичная Windows, то восстановление MBR можно пропустить. Если образ переносится на чистый жесткий диск, то пункт про MBR следует отметить).
8. В настройках раздела C нажмите ссылку «Новое хранилище», а в открывшемся окне укажите раздел C:, для продолжения нажмите кнопку «Принять». Будьте внимательны при выборе раздела.

9. Оставляем всё по умолчанию в пунктах «Тип раздела и Размер раздела» и нажимаем кнопку «Далее».

10. Проверяем сводные данные, нет ли ошибок. Если уверены, нажимаем кнопку «Приступить».

Учтите, что любое прерывание операции неизбежно приведет к удалению раздела С: и невозможности загрузить Windows. Если это произошло, повторяем все шаги и вместо раздела С: указываем уже свободное пространство.

11. Начнется процесс восстановления образа на выбранный раздел, по окончании чего появится окно об успешном завершении операции восстановления. При появлении ошибки об отсутствующем драйвере сообщение можно пропустить.

12. Восстановление завершено, отключите USB внешний диск, выньте загрузочный диск из привода и перезагрузите компьютер.

11.4.3. Установка ОС из архива

Использование образов позволяет копировать и восстанавливать служебную информацию, однако перенос геометрии диска связан с проблемами при использовании целевого HDD иного, чем оригинал, объема. Поскольку ОС работает на уровне файлов, логично заменить инсталляцию операционной системы копированием соответствующих файлов в подготовленные директории. Вместе с тем в ряде случаев необходима коррекция записи MBR. Дальнейшая установка занимает 1-2 шага:

1. Распаковываем архив на целевой раздел;
2. Перезагружаем компьютер, если он не вывалился в BSOD – доустанавливаем (при необходимости) драйверы.

К сожалению, этот способ тоже не лишен недостатков – поскольку иногда критично физическое размещение некоторых файлов, работоспособность ОС не гарантируется.

11.4.4. Ход работы

1. Ознакомьтесь с методическими указаниями и используемым оборудованием. Примечание: при отсутствии аппаратного CD/DVD привода используйте виртуальный привод Zalman.

Изучите состав и размещение утилит и файлов образов на USB носителе.

2. Установите ОС способом, описанным в 11.4.1., в ходе выполнения пункта «Настройка диска» произведите разметку диска и форматирование. В качестве дистрибутивов используйте (по выбору) файлы образов Win7x64.iso или WinXPx86.iso.

Зафиксируйте время, необходимое для установки ОС без дополнительных драйверов.

3. Выполните поиск в Интернете программ, предназначенных для создания загрузочных / установочных USB-флешек. Изучите одну из них, предназначенную для работы с компьютерами с UEFI.

4. Установите ОС способом, описанным в 11.4.2., используя либо образ Acronis+WinXP.iso, либо запустив иной Acronis, в качестве образа примените WinXP+SP3+USB.tib.

5. Зафиксируйте время, необходимое для установки ОС без дополнительных драйверов.

6. Оформите отчет, описывающий ход выполнения задания и выявленные на разных этапах нюансы установки.

11.5. Требования к отчету и защите

1. Описание хода работы (подключение носителя, выбор режима его работы, выбор из папки ISO образа требуемого дистрибутива и т.п.).
2. Последовательность шагов по установке ОС из ISO образа.
3. Диалог по ходу установки.
4. Ответы на вопросы и выводы по работе:
 - Где располагаются файлы ОС, драйверы и иные системные файлы, и нужны ли они?
 - Каков минимально необходимый набор файлов для загрузочной флешки/дискеты?
 - Чем отличается загрузочный диск от незагрузочного?
 - Что, помимо данных файлов, должно присутствовать на загрузочном устройстве?
 - Как указать необходимость наличия драйверов? Что и как в этом случае используется при загрузке MS DOS?
 - Какие утилиты обеспечивают возможность просмотра или редактирования загрузчиков?

12. ЗАКЛЮЧЕНИЕ

Выполнение лабораторных работ данного практикума должно способствовать формированию у студентов профессиональных знаний по основам построения устройств и элементов вычислительных систем, необходимых для выбора их архитектуры, конфигурации и эксплуатации:

В ходе практического освоения дисциплины студенты выполняют исследования элементов, узлов и устройств, реализующих функции обработки данных и управления в электронных вычислительных машинах, принципы построения запоминающих и внешних устройств и их интерфейсов;

Результатами выполнения работ являются приобретение навыков выбора, комплексирования и эксплуатации программно-аппаратных средств в вычислительных и информационных системах; формирование базовых знаний, умений и навыков для оптимального (в том числе самостоятельного) выбора элементной базы для построения различных архитектур вычислительных средств.

13. ЛИТЕРАТУРА

Основная

1. Горнец, Н.Н. ЭВМ и периферийные устройства. Компьютерные и вычислительные системы : учеб. / Н. Н. Горнец, А. Г. Рошин . - Москва : Академия, 2012. - 235 с.

2. Пятибратов, А.П. Вычислительные системы, сети и телекоммуникации [Электронный ресурс]: учеб. / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко. - Москва : КноРус, 2017. - 372 с. (ЭБС «Book.ru»).

3. Лошаков, С. Периферийные устройства вычислительной техники [Электронный ресурс] / С. Лошаков. - 2-е изд., исправ. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 436 с. (ЭБС «Университетская библиотека онлайн»).

4. Торгонский, Л.А. Проектирование центральных и периферийных устройств ЭВС [Электронный ресурс] : учеб. пособие / Л.А. Торгонский, П.Н. Коваленко ; Министерство образования и науки Российской Федерации, Томский Государственный Университет Систем Управления и Радиоэлектроники (ТУСУР). - Томск : Эль Контент, 2012. - Ч. II. Микропроцессорные ЭВС. - 176 с. (ЭБС «Университетская библиотека онлайн»).

Дополнительная

1. Гусев, В.Г. Электроника и микропроцессорная техника : учеб. / В. Г. Гусев, Ю. М. Гусев ; авт. Гусев Ю.М. - 5-е изд., стереотип. - Москва : Высшая школа, 2008. - 798 с.

2. Кардашев, Г.А. Цифровая электроника на персональном компьютере. Electronics Workbench и Micro-Cap / Г. А. Кардашев. - Москва : Горячая линия-Телеком, 2003. - 311 с.

3. Суханова, Н.В. Основы электроники и цифровой схемотехники [Электронный ресурс] : учеб. пособие / Н.В. Суханова ; Министерство образования и науки РФ, Воронежский государственный университет инженерных технологий. - Воронеж : Воронежский государственный университет инженерных технологий, 2017. - 97 с. (ЭБС «Университетская библиотека онлайн»).

4. Микропроцессорные системы : учеб. пособие / Е. К. Александров [и др.]. - Санкт-Петербург : Политехника, 2002. - 935 с.

5. Сычев, А.Н. ЭВМ и периферийные устройства [Электронный ресурс] : учебное пособие / А.Н. Сычев ; Министерство образования и науки Российской Федерации, Томский Государственный Университет Систем Управления и Радиоэлектроники (ТУСУР). - Томск : ТУСУР,

Локальный электронный методический материал

Владимир Вячеславович Капустин

Вычислительная техника

Редактор Г. А. Смирнова

Уч.-изд. л. 7,0. Печ. л. 5,6

Издательство федерального государственного бюджетного образовательного
учреждения высшего образования
«Калининградский государственный технический университет».
236022, Калининград, Советский проспект, 1