



Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новгородский государственный университет имени Ярослава Мудрого»
МНОГОПРОФИЛЬНЫЙ КОЛЛЕДЖ
ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ
Учебно-методическая документация

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
ПО ПРАКТИЧЕСКИМ ЗАНЯТИЯМ И ЛАБОРАТОРНЫМ РАБОТАМ**

ОП.04 ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Специальность:
11.02.11 Сети связи и системы коммутации

Квалификация выпускника: техник

Разработчик: Кривый А. Я., преподаватель

Методические рекомендации приняты на заседании предметной (цикловой) комиссии дисциплин профессионального цикла специальности 11.02.11 Сети связи и системы коммутации колледжа протокол № 1 от 18.09.2017 г.

Председатель предметной (цикловой) комиссии  Г. В. Цыбульская

Содержание

Пояснительная записка	4
Тематический план и содержание учебной дисциплины «Вычислительная техника»	5
Содержание практических занятий и лабораторных работ	11
Практическая работа №1. Перевод чисел из одной системы в другую	11
Практическая работа №2. Выполнение арифметических операций в обратном и дополнительных кодах	18
Практическое занятие №3. Изучение номенклатуры ИМС логических элементов в программе Electronic WorkBench	24
Практическая работа №4. Построение комбинационных схем по переключательным функциям в разных базисах	30
Практическая работа №5. Синтез комбинационного логического устройства с применением карт Карно	33
Лабораторная работа №1. Исследование шифраторов и дешифраторов	39
Лабораторная работа №2. Исследование и построение мультиплексоров. Реализация заданной функции с помощью мультиплексора	45
Лабораторная работа №3. Исследование триггеров	51
Лабораторная работа №4. Исследование сумматоров	58
Лабораторная работа №5. Построение четырёхразрядного сумматора. Вычитание	62
Лабораторная работа №6. Изучение работы цифрового компаратора	65
Лабораторная работа №7. Устройства контроля четности	68
Лабораторная работа №8. Принцип работы кэш-памяти. Алгоритмы замещения строк	71
Лабораторная работа №9. Моделирование работы арифметико-логического устройства	77
Лабораторная работа №10. Архитектура ЭВМ и система команд. Вычисление значения выражения с применением различных способов адресации	81
Лабораторная работа №11. Программирование типовых управляющих структур на языке Ассемблер	78
Лабораторная работа №12. Программирование цикла с переадресацией	89
Лабораторная работа №13. Подпрограмма и стек	93
Лабораторная работа №14. Программирование внешних устройств	90
Лабораторная работа №15. Подсистема прерывания	102
Информационное обеспечение обучения	108
Лист регистрации изменений	109

Пояснительная записка

Методические рекомендации по практическим занятиям и методические указания по лабораторным работам, являющиеся частью учебно-методического комплекса по дисциплине «Вычислительная техника» составлены в соответствии с:

1. Федеральным государственным образовательным стандартом по специальности 11.02.11 Сети связи и системы коммутации;
2. Рабочей программой учебной дисциплины;
3. Положением о планировании, организации и проведении лабораторных работ и практических занятий студентов, осваивающих основные профессиональные образовательные программы среднего профессионального образования в колледжах НовГУ.

Методические рекомендации включают 15 лабораторных занятий, предусмотренных рабочей программой учебной дисциплины в объеме 30 часов и 5 практических работ предусмотренных рабочей программой учебной дисциплины в объеме 10 часов.

В результате выполнения практических заданий обучающийся **должен уметь:**

- использовать типовые средства вычислительной техники и программного обеспечения в профессиональной деятельности;
- осуществлять перевод чисел из одной системы счисления в другую, применять законы алгебры логики;
- строить и использовать таблицы истинности логических функций, элементов и устройств.

В результате освоения дисциплины обучающийся **должен знать:**

- виды информации и способы их представления в ЭВМ;
- логические основы ЭВМ, основы микропроцессорных систем;
- типовые узлы и устройства ЭВМ, взаимодействие аппаратного и программного обеспечения ЭВМ.

Для выполнения практических работ используются следующие технические средства:

- персональные компьютеры, 10 персональных компьютеров Core i3 3.0, 4 Gb, 4 Гб ОЗУ, 250 Gb HDD
- мультимедийный проектор.

а также программные средства:

- ОС Windows 7/XP
- Программа «Учебная ЭВМ»
- Программа для моделирования Electronic Work Bench 5.12 и выше

Критерии оценки выполнения практических и лабораторных работ:

- если выполнена работа в полном объеме и без ошибок, даны ответы на все поставленные вопросы, то ставится оценка «5»;
- если работа выполнена полностью, но есть неточности при выполнении, даны ответы на контрольные вопросы, то ставится оценка «4»;
- если работа выполнена не полностью, есть неточности при выполнении, не даны ответы на контрольные вопросы, то ставится оценка «3»;
- если работа выполнена меньше чем наполовину, то ставится оценка «2».

Тематический план и содержание учебной дисциплины «Вычислительная техника»

Наименование разделов и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающихся, курсовая работа (проект)	Объем часов	Уровень освоения
1	2	3	4
Раздел 1. Информационные основы ЭВМ		22	
Тема 1.1. Введение	Содержание учебного материала		2
	Основные сведения о ВТ. Понятие вычислительного устройства. История развития ВТ. Классификация вычислительных устройств Задачи курса.	2	
	Самостоятельная работа №1: Проработка учебной литературы, составление конспекта по темам «Значение вычислительной техники в современном мире. Этапы развития ЭВМ и тенденции».	2	
Тема 1.2. Кодирование информации	Содержание учебного материала		2
	Коды, применяемые в ЭВМ: двоичные, позиционные, комбинационные, самокорректирующиеся, параллельные, последовательные. Признаки сигнала: амплитудный, фазовый, полярный, частотный, временной, импульсный. Формы представления чисел: с фиксированной и плавающей точкой.	2	
	Самостоятельная работа №2: Подготовка сообщения на тему Современный уровень и перспективы развития элементов и устройств цифровой техники. Форма сигналов; их параметры: низкий и высокий логические уровни, частота повторения, фронт, срез. Сигналы передачи цифровой информации и их параметры.	2	
Тема 1.3. Системы счисления	Содержание учебного материала		2
	Позиционные системы счисления: 10, 2, 8, 16, 2-10. Перевод из одной системы счисления в другую. Формы представления чисел в ЭВМ. Перевод чисел из одной системы в другую.	2	
	Практическое занятие №1. Перевод чисел из одной системы в другую	2	
Тема 1.4. Выполнение арифметических	Самостоятельная работа №3: Перевод чисел из одной системы счисления в другую систему счисления	4	2
	Содержание учебного материала		
	Коды: прямой, обратный, дополнительный. Выполнение операции сложения с использованием кодов. Выполнение арифметических операций в обратном и	2	

операций	дополнительных кодах.		
	Практическое занятие №2. Выполнение арифметических операций в обратном и дополнительных кодах.	2	
	Самостоятельная работа №4 Выполнение арифметических операций в прямом и обратном кодах.	2	
Раздел 2. Логические основы ЭВМ		28	
Тема 2.1. Логические функции и схемы.	Содержание учебного материала	6	2
	Способы представления логических функций: словарное, табличное, формульное, графическое. Условные графические обозначения (УГО) логических элементов. Основные законы и тождества алгебры логики. Понятие логического базиса. Базисы И, НЕ; ИЛИ, НЕ; И, ИЛИ, НЕ. Логические элементы: схемы, параметры. Изучение номенклатуры ИМС логических элементов.		
	Практическое занятие №3. Изучение номенклатуры ИМС логических элементов в программе Electronic WorkBench. Практическое занятие №4. Построение комбинационных схем по переключательным функциям в разных базисах	4	
	Самостоятельная работа №5: Составление таблиц истинности для булевых функций.	2	
Тема 2.2. Синтез логических устройств	Содержание учебного материала	8	3
	Совершенная дизъюнктивная и конъюнктивная нормальные формы (СДНФ и СКНФ) представления функций. Комбинационные и последовательностные логические устройства. Минимизация логических функций с помощью законов алгебры логики. Карты Карно. Диаграммы Вейча. Минимизация с помощью карт Карно (диаграмм Вейча). Анализ и синтез комбинационных логических устройств. Задачи анализа и синтеза. Переход от табличной формы представления логической функции к формульной. Этапы синтеза логических комбинационных устройств. Синтез комбинационного логического устройства с применением карт Карно.		
	Практическое занятие №5. Синтез комбинационного логического устройства с применением карт Карно.	2	
	Самостоятельная работа №6. Анализ комбинационных схем	2	

Тема 2.3. Программы компьютерного моделирования	Содержание учебного материала	2	2
	Программы компьютерного моделирования. Основы моделирования. Освоение программы моделирования цифровых устройств.		
	Самостоятельная работа №7: Составить перечень основных элементов из библиотеки EWB, необходимый для моделирования логических схем и цифровых устройств ЭВМ.	2	
Раздел 3. Типовые узлы ЭВМ		52	
Тема 3.1. Назначение шифраторов	Содержание учебного материала	2	3
	Синтез шифраторов на интегральных схемах. Назначение и классификация дешифраторов. Синтез линейного дешифратора. Каскадные, матричные дешифраторы. Дешифраторы в интегральном исполнении. УГО. Сравнительный анализ различных дешифраторов, области использования.		
	Лабораторная работа №1. Исследование шифраторов и дешифраторов	2	
	Самостоятельная работа №8 Сравнительный анализ различных дешифраторов, области использования.	2	
Тема 3.2. Мультиплексоры и демультиплексоры	Содержание учебного материала	2	3
	Назначение мультиплексоров. Схема мультиплексора, УГО. Мультиплексоры в интегральном исполнении, способы наращивания. Универсальность мультиплексоров. Назначение демультиплексоров, схема демультиплексора, УГО. Демультиплексоры в интегральном исполнении. Электронный коммутатор.		
	Лабораторная работа №2. Исследование и построение мультиплексоров. Реализация заданной функции с помощью мультиплексора.	2	
	Самостоятельная работа №9. Изучение номенклатуры мультиплексоров по справочнику	2	
Тема 3.3. Триггеры	Содержание учебного материала	2	2
	Типы триггеров. Таблицы переключения R-S триггера, D-триггера, T-триггера, J-K-триггера. Временные диаграммы.		
	Лабораторная работа №3. Исследование триггеров	2	
	Самостоятельная работа №10: Сфера применения различных типов триггеров. Изучение номенклатуры триггеров по справочнику	4	

Тема 3.4. Регистры	Содержание учебного материала	1	2
	Назначение, классификация, характеристики регистров. Схемы параллельного, сдвига, реверсивного регистров. УГО. Схема выдачи информации. Регистры в интегральном исполнении.		
	Самостоятельная работа №11. Изучение номенклатуры регистров по справочнику	2	
Тема 3.5. Счётчики	Содержание учебного материала	1	3
	Назначение, классификация, характеристики счетчиков. УГО. Схемы асинхронных и синхронных счетчиков; с последовательным, сквозным, параллельным переносом. Счетчики с произвольным коэффициентом пересчета. Счетчики в интегральном исполнении и их использование в качестве делителей частоты. Кольцевые счетчики.		
	Самостоятельная работа №12. Изучение номенклатуры счетчиков по справочнику	3	
	Зачётное занятие I семестр	2	
Тема 3.6. Сумматоры	Содержание учебного материала	4	2
	Сумматоры. Назначение, классификация и характеристики. УГО. Математическое описание и схема одноразрядного полного комбинационного сумматора. Многоразрядные сумматоры: параллельные и последовательные. Организация ускоренного переноса. Сумматоры в интегральном исполнении.		
	Лабораторная работа №4. Исследование сумматоров.	4	
	Лабораторная работа №5. Построение четырёхразрядного сумматора. Вычитание.	4	
	Самостоятельная работа №14 Выполнение домашнего практического задания по теме: Построение схемы четырехразрядного сумматора с параллельным переносом	2	
Тема 3.7. Преобразователи кодов	Содержание учебного материала	2	2
	Классификация и назначение преобразователей кодов. УГО. Синтез преобразователей прямого кода в дополнительный и обратный; кода 8421 в 7-сегментный; двоичного в двоично-десятичный. ПЛИМ как стандартизованный кодопреобразователь. Структура ПЛИМ. Понятие ПЛИС и их применение		
	Самостоятельная работа №15. Изучение номенклатуры преобразователей двоичного кода по справочнику	2	
Тема 3.8. Распределители, цифровые компараторы	Содержание учебного материала	2	2
	Распределители: назначение, принципы построения на базе регистров, счетчиков и дешифраторов. Синхрогенераторы: назначение, основные схемы. Цифровые компараторы: назначение, схемы.		

	Лабораторная работа №6. Цифровые компараторы. Лабораторная работа №7. Устройства контроля чётности	4	
	Самостоятельная работа №16. Проработка учебной литературы, конспекта, подготовка сообщения Аналого-цифровые и цифро-аналоговые преобразователи. Принцип действия, характеристики. Типы, схемы.	2	
	Тест по разделу Типовые узлы ЭВМ	1	
Раздел 4. Устройства ЭВМ		21	
Тема 4.1. Запоминающие устройства ЭВМ	Содержание учебного материала	4	2, 3
	Классификация, параметры запоминающих устройств. ПЗУ, ОЗУ, кэш-память, буферное ЗУ, ВЗУ. Организация ЗУ: адресная, словарная, двухкоординатная, стековая, с произвольной выборкой, ассоциативная. Назначение, классификация ПЗУ: масочные, программируемые, репрограммируемые. УГО. Схемотехника постоянных запоминающих устройств (ПЗУ).		
	Назначение ОЗУ. Статические и динамические микросхемы ОЗУ. Построение модулей памяти на базе интегральных микросхем. Кэш-память: принцип организации, назначение. Микросхемотехника оперативных запоминающих устройств		
	Внешние запоминающие устройства: принцип работы, конструкции. Основные характеристики.		
	Лабораторная работа №8. Принцип работы кэш-памяти. Алгоритмы замещения строк.	2	
	Самостоятельная работа №17. Организация памяти в ЭВМ. Концепция многоуровневой памяти	2	
Тема 4.2. Процессоры ЭВМ	Содержание учебного материала	4	1, 2, 3
	Процессор: состав, назначение. АЛУ: структура, типы, выполняемые операции. Устройства управления: схемно-логического и микропрограммного типов. Понятие о прерываниях, виды прерываний. Синтез простейшего устройства управления.		
	Лабораторная работа №9. Моделирование работы арифметико-логического устройства	2	
	Самостоятельная работа №18 Подготовка сообщения на тему: «Процессор. Основные характеристики процессора».	3	
Тема 4.3. Интерфейсы в ВТ	Содержание учебного материала	2	2
	Состав, назначение, требования, типы интерфейсов в ВТ. Параллельные и последовательные. Режимы работы симплексный, дуплексный и полудуплексный.		

	Интерфейсы ЭВМ "общая шина" и "мультишина". Основные характеристики.		
	Самостоятельная работа №19. Проработка учебной литературы, конспекта, подготовка сообщения на тему: «Виды интерфейсов в вычислительной технике».	2	
Раздел 5. Микропроцессоры (МП) и микропроцессорные системы (МПС)		21	
Тема 5.1. Архитектура МП	Содержание учебного материала	2	2
	Классификация. Структурная схема МП. Состав, назначение и взаимодействие отдельных блоков МП. Основные характеристики. СКС и RISC процессоры.		
Тема 5.2 Архитектура МПС	Содержание учебного материала	1	1
	Особенности построения МПС. Понятие о вычислительных системах и их видах. Микропроцессорные комплекты: состав, назначение отдельных БИС. Ресурсы МПС: система прерываний, система прямого доступа в память.		
	Лабораторная работа №10. Архитектура ЭВМ и система команд. Вычисление значения выражения с применением различных способов адресации. Лабораторная работа №11. Программирование типовых управляющих структур на языке Ассемблер. Лабораторная работа №12. Программирование цикла с переадресацией. Лабораторная работа №13. Подпрограмма и стек. Лабораторная работа №14. Программирование внешних устройств. Лабораторная работа №15. Подсистема прерывания.	12	
	Самостоятельная работа №21. Построение алгоритмов: линейных, ветвления, цикла. Составление программ на языке низкого уровня для реализации заданных алгоритмов.	2	
Тема 5.3 Применение СВТ в технике связи	Содержание учебного материала	1	1
	Процессоры, применяемые в ПК. Особенности однокристалльных микро-ЭВМ. PRC-контроллеры и области их применения в технике связи.		
	Самостоятельная работа №22. Подготовка доклада на тему: Компьютеры будущего	1	
	Зачётное занятие II семестр	2	
	Всего	144	

Содержание практических занятий и лабораторных работ

Раздел 1. Информационные основы ЭВМ

Тема 1.3. Системы счисления

Практическая работа №1. Перевод чисел из одной системы в другую

Время, отводимое на выполнение практической работы - 2 часа.

1. **Цель занятия:** Получение и закрепление умений перевода целых чисел из одной системы счисления в другую, получение и закрепление умений перевода дробных чисел из одной системы счисления в другую.

2. **Основные теоретические положения:**

- 2.1 Понятие системы счисления

Системой счисления называется способ представления и изображения чисел с использованием строго ограниченного набора символов, каждый из которых имеет определённые количественные значения. Числа в системах счисления представляются с помощью некоторого набора знаков – **цифр**, а их количество зависит от используемой системы. Позиция цифры внутри изображаемого числа называется **разрядом**.

Для представления чисел в ЭВМ применяется **двоичная**¹ система счисления (используется две цифры «1» и «0»). Каждый следующий разряд больше предыдущего в два раза. Широкое распространение также получили восьмеричная (используются цифры 0,1,...,7) и шестнадцатеричная (используются цифры 0,1,...,9,A,B,C,D,E,F²) системы счисления.

Представление десятичного числа

Пример 1.1 $C_{10} = 414.547 = 4 \cdot 10^2 + 1 \cdot 10^1 + 4 \cdot 10^0 + 5 \cdot 10^{-1} + 4 \cdot 10^{-2} + 7 \cdot 10^{-3}$

В данном примере количество разрядов равно 6. Единица каждого разряда в десять раз больше единицы предыдущего разряда. Приведённый пример иллюстрирует, что десятичная система счисления является позиционной, т.к. имеются три цифры «четыре», которые расположены в разных разрядах числа. Все три четвёрки имеют различный вес, первая указывает сотни, вторая единицы, а третья сотые доли числа.

- 2.2 Перевод в различные системы счисления

Для того чтобы перевести целое число C_{N1} в систему счисления с основанием $N2$, необходимо последовательно делить его на $N2$, выписывая остатки от деления, до тех пор, пока остаток от деления станет не больше $N2$.

Пример 1.2 Представление десятичного числа в двоичной, восьмеричной и шестнадцатеричной системах счисления

$$C_{10} = 214,625$$

$$C_2 = 1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3}$$

$$C_8 = 3 \cdot 8^2 + 2 \cdot 8^1 + 6 \cdot 8^0 + 5 \cdot 8^{-1}$$

$$C_{16} = 13 \cdot 16^1 + 6 \cdot 16^0 + 10 \cdot 16^{-1}$$

Рассмотрим подробно перевод числа 41 в двоичную систему счисления. На первом шаге 41 делится на 2, частное от деления равно 20 записывается над чертой справа от делимого, а остаток от деления равный 1 под делимым. На втором шаге 20 делится на 2,

¹ В двоичной системе наиболее просто реализуются арифметические операции, что наряду с экономичностью реализации аппаратной части устройств вычислительных машин (все выходы схем принимают значение 0 или 1) сделали эту систему основной в современных ЭВМ.

² A = 10, B = 11, C = 12, D = 13, E = 14, F = 15.

частное от деления равное 10 записывается над чертой справа от делимого, а остаток от деления равный 0 под делимым. Так продолжается до тех пор, пока полученное на очередном шаге делимое не станет ≤ 1 . И, наконец, полученные остатки от деления и последнее частное записываются в порядке обратном направлению деления.

Пример 1.3 Перевод числа $C_{10}=41$ в двоичную, восьмеричную и шестнадцатеричную системы счисления.

$$\frac{41}{1} \frac{20}{0} \frac{10}{0} \frac{5}{1} \frac{2}{0} \frac{1}{0} \rightarrow C_2 = 101001 \quad \frac{41}{1} \frac{5}{1} \rightarrow C_8 = 51 \quad \frac{41}{9} \frac{2}{9} \rightarrow C_{16} = 29$$

ЭВМ работает с двоичными числами, человеку проще воспринимать числа в десятичной системе счисления, а шестнадцатеричная форма записи наиболее компактна. Поэтому повсеместно требуется переводить числа из одной системы счисления в другую.

Так как двоичная, восьмеричная и шестнадцатеричная системы счисления связаны между собой через степени числа 2, то прямое и обратное преобразования между этими системами можно производить гораздо проще.

Для перевода целого числа из двоичной системы счисления в восьмеричную (шестнадцатеричную) необходимо разбить число справа от младшего разряда на триады (тетрады), т.е. группы, состоящие из трех (четырёх) цифр. Если в последней триаде (тетраде) остается менее трех (четырёх) цифр, то вместо недостающих цифр слева записываются нули.

Заменив каждую триаду (тетраду) соответствующей восьмеричной (шестнадцатеричной) цифрой, получают число, записанное в восьмеричной (шестнадцатеричной) системе счисления.

Пример 1.4 Упрощенный перевод целого числа из двоичной системы счисления в восьмеричную (шестнадцатеричную)

$$C_2 = 101111 = 101|111 \rightarrow C_8 = 57$$

$$C_2 = 10101110 = 1010|1110 \rightarrow C_{16} = AE$$

Пример 1.5 Исходные данные: A=1533 и B=4374

Перевод в двоичную систему:

$$1533/2= \quad 766/2=383/2=191/2=95/2= \quad 47/2= \quad 23/2= \quad 11/2= \quad 5/2= \quad 2/2= 1$$

Остатки: 1 0 1 1 1 1 1 1 1 0

Ответ: $1533 = 1011111101_2$

$$\begin{array}{ccccccc} 7374/2= & 3687/2= & 1843/2= & 921/2= & 460/2= & 230/2= & 115/2= & 57/2= \\ 28/2= & 14/2= & 7/2= & 3/2= & 1 & & & \end{array}$$

Остатки: 0 1 1 1 0 0 1 1 0 0 1 1

Ответ: $7347 = 1110010110011_2$

Таблица 1 Перевод в восьмеричную систему:

8	0	1	2	3	4	5	6	7
2	000	001	010	011	100	101	110	111

Разбиваем на группы двоичное число на группы по три разряда, начиная с первого разряда, при необходимости дописываем нужное количество нулей перед числом:

1533= 010 111 111 101 - в двоичной системе счисления

Заменяем: 2 7 7 5 - в восьмеричной системе счисления

Ответ: $1533=2775_8$

7347= 001 110 010 110 011 - в двоичной системе счисления
 Заменяем: 1 6 2 6 3 - в восьмеричной системе счисления

Ответ: 7347=16263₈

Таблица 2 Перевод в шестнадцатеричную систему:

16	0	1	2	3	4	5	6	7
2	0000	0001	0010	0011	0100	0101	0110	0111
16	8	9	A	B	C	D	E	F
2	1000	1001	1010	1011	1100	1101	1110	1111

Разбиваем на группы двоичное число на группы по четыре разряда, начиная с первого разряда, при необходимости дописываем нужное количество нулей перед числом:

1533= 0101 1111 1101 - в двоичной системе счисления
 Заменяем: 5 F D - в шестнадцатеричной системе счисления

Ответ: 1533=5FD₁₆

7347= 0001 1100 1011 0011 - в двоичной системе счисления
 Заменяем: 1 C B 3 - в шестнадцатеричной системе счисления

Ответ: 7347=1CB3₁₆

2.3 Перевод дробного числа в систему счисления с основанием N₂

Для того чтобы перевести дробное число C_{N₁} в систему счисления с основанием N₂ необходимо последовательно умножать его на N₂, выписывая целые части, полученные в результате умножения, до тех пор, пока количество разрядов полученного числа не достигнет заданного³.

Пример 1.6 Перевод числа C₁₀=0,375 в двоичную, восьмеричную и шестнадцатеричную системы счисления

$$\frac{0,375}{0} \frac{0,75}{1} \frac{0,5}{1} \frac{0}{1} \rightarrow C_2 = 0,011 \quad \frac{0,375}{3} \frac{0}{3} \rightarrow C_8 = 0,3 \quad \frac{0,375}{6} \frac{0}{6} \rightarrow C_{16} = 0,6$$

Перевод дробного числа, также удобно представлять в виде последовательности цифр, разделённых чертой. Рассмотрим подробно перевод числа 0,375 в двоичную систему счисления. На первом шаге 0,375 умножается на 2, полученное множимое равное 0,75 записывается над чертой справа от 0,375, а целая часть равная 0 под множимым 0,375. На втором шаге 0,75 умножается на 2, полученное множимое равное 0,5 записывается над чертой справа от 0,75, а целая часть равная 1 под множимым 0,75. Так продолжается до тех пор, пока полученное на очередном шаге множимое не станет равно 0, или не заполняться все разряды, отведённые для этого числа. И, наконец, полученные целые части записываются в прямом порядке.

Пример 1.7 Обратный перевод чисел из системы с основанием N₂ в систему с основанием N₁

$$C_2 = 101001 \rightarrow C_{10} = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 32 + 8 + 1 = 41$$

³ Т.е. количество умножений фактически определяет разрядность получаемого дробного числа.

$$C_2 = 0,011 \rightarrow C_{10} = 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} = 0,25 + 0,125 = 0,375$$

Для перевода дробного числа из двоичной системы счисления в восьмеричную (шестнадцатеричную) необходимо разбить число слева от старшего разряда на триады (тетрады). Если в последней триаде (тетраде) остается менее трех (четырех) цифр, то вместо недостающих цифр слева записываются нули. Заменяв каждую триаду (тетраду) соответствующей восьмеричной (шестнадцатеричной) цифрой, получают число, записанное в восьмеричной (шестнадцатеричной) системе счисления.

Пример 1.8 Упрощенный перевод дробного числа из двоичной системы счисления в восьмеричную (шестнадцатеричную)

$$C_2 = 0,101111 = 0,101|111 \rightarrow C_8 = 0,57$$

$$C_2 = 0,10101110 = 0,1010|1110 \rightarrow C_{16} = 0,AE$$

Обратный перевод осуществляется аналогично, т.е. восьмеричные (шестнадцатеричные) числа заменяются триадами (тетрадами) двоичных чисел.

2.4 Представление дробных чисел в ЭВМ

В ЭВМ используется два вида представления чисел: **с фиксированной точкой** или **естественная форма** и **с плавающей точкой** или **нормальная форма**. Точка определяет границу раздела целой и дробной частей числа. Любая программа в ЭВМ реализуется в виде простых операций, выполняемых **центральный процессором** (ЦП). Одной из важнейших характеристик любого процессора является его **разрядность**, т.е. количество разрядов, используемых для представления чисел. Эта величина в силу определенных причин ограничена⁴. При использовании формы с фиксированной точкой для изображения некоторого числа, точка имеет строго определенное место для всех чисел.

Пример 1.9 Естественная форма десятичного числа

$$+14814.00142 \quad +00351.84300 \quad +00000.08414 \quad -00643.13843 \quad -54712.12101$$

В ЭВМ у двоичных чисел с фиксированной точкой для точки выделяют место либо в начале (дробное число), либо в конце (целое число). Для изображения знака числа отводится специальный знаковый разряд в начале числа, в который записывает 1, если число отрицательное и 0, если число положительное.

Другой формой представления чисел в ЭВМ является нормальная форма (числа с плавающей точкой). Для представления чисел в форме с плавающей точкой в ЭВМ используется две группы цифр - мантисса М и порядок Р⁵. В общем виде это представление может быть выражено, как: $C = \pm M \cdot N^{\pm P}$, где С – представляемое число, а N – основание системы счисления.

В разных ЭВМ применяются различные варианты представления чисел в форме с плавающей точкой. Для примера рассмотрим внутреннее представление вещественного числа в 4-х байтовой ячейке памяти.

В ячейке должна содержаться следующая информация о числе: знак числа, порядок и значащие числа мантииссы.

Пример 1.10 Представление десятичных чисел с плавающей точкой

Числа, указанные в примере 2.5 могут быть представлены в следующем виде.

$$+0.1481400142 \cdot 10^5 \quad +0.351843 \cdot 10^3 \quad +0.08414 \cdot 10^{-1} \quad -0.64313843 \cdot 10^3$$

⁴ Электрические устройства памяти, в которых хранятся числа, имеют конечное число разрядов.

⁵ Абсолютное значение мантииссы всегда меньше единицы, а порядок является целым числом.

Таблица 3 Внутреннее представление вещественного числа в 4-х байтовой ячейке

± машинный порядок	М А	Н Т И С	С А
1-й байт	2-й байт	3-й байт	4-й байт

В старшем бите 1-го байта хранится знак числа: 0 означает плюс, 1 минус. Оставшиеся 7 бит байта содержат машинный порядок. В следующих трёх байтах хранятся значащие цифры мантиисы (24 разряда).

Машинный порядок в диапазоне от 0000000 до 1111111 (т.е. от 0 до 127 десятичной системе счисления). Всего 128 значений. Порядок может быть как положительным, так и отрицательным. Разделим 128 значений поровну от -64 до 63.

Машинный порядок смещён относительно математического и имеет только положительное значение. Смещение выражается формулой: $M_p = p + 64$. В двоичной системе счисления: $M_{p2} = p_2 + 100\ 0000_2$.

Алгоритм записи внутреннего представления числа:

1. Переводим модуль данного числа в двоичную систему счисления с 24-мя значащими числами.
2. Нормализуем двоичное число (первый разряд мантиисы после запятой – значащий).
3. Находим машинный порядок в 2-й системе счисления.
4. Учитывая знак числа записываем его представление в 4-х байтовом машинном слове.

3. Задание к работе

3.1 Перевести заданные числа А и В из десятичной системы в двоичную, восьмеричную и шестнадцатеричную системы счисления.

Таблица 3 Варианты заданий (вариант определяется по номеру в журнале)

Вариант	А	В	Вариант	А	В
1	174	6936	16	342	5456
2	192	6893	17	353	5345
3	319	5789	18	364	5234
4	224	6678	19	375	5123
5	231	6567	20	386	5012
6	242	6456	21	397	4901
7	253	6345	22	408	4890
8	264	6234	23	419	4789
9	275	6123	24	430	4678
10	286	6012	25	441	4665
11	297	5901	26	452	4657
12	308	5890	27	463	4639
13	207	6789	28	474	4626
14	320	5678	29	485	4613
15	331	5567	30	496	4600

3.2 Перевод дробного десятичного числа в 2-ю, 8-ю, 16-ю системы счисления

3.2.1 Произвести перевод дробного числа из десятичной системы счисления в системы счисления с основанием, заданными степенями числа 2. При иррациональном представлении числа использовать первые десять разрядов.

3.2.2 Произвести перевод дробного числа из десятичной системы счисления в систему счисления с основанием 8, 16.

Таблица 5 Варианты заданий (вариант определяется по номеру в журнале)

Вариант	Дробное число	Вариант	Дробное число	Вариант	Дробное число
---------	---------------	---------	---------------	---------	---------------

Вариант	Дробное число	Вариант	Дробное число	Вариант	Дробное число
1	0,34341	11	0,90212	21	0,78091
2	0,66611	12	0,87985	22	0,23121
3	0,98834	13	0,34430	23	0,47896
4	0,68510	14	0,65620	24	0,12345
5	0,22287	15	0,35305	25	0,84771
6	0,80740	16	0,51145	26	0,12512
7	0,87979	17	0,58419	27	0,25252
8	0,78824	18	0,38867	28	0,14315
9	0,49300	19	0,22670	29	0,12874
10	0,43331	20	0,76804	30	0,36987

3.3 Представление чисел в форме с фиксированной и плавающей запятой

3.3.1 Переведите заданные числа в 2-ю систему счисления.

3.3.2 Запишите мантиссу и порядок чисел А и В, представленных в таблице 6

Таблица 6 Варианты заданий для чисел с фиксированной и плавающей запятой

№ в.	А	В	№ в.	А	В	№ в.	А	В
1	+23 411.207	-2 345.22	11	+0.221	-0.00104	21	+0.2321	-0.0404
2	+33 813.011	-3 732.515	12	+0.8754	-0.0123	22	+0.8701	-0.0223
3	+18 622.192	-4 345.01	13	+0.9901	-0.0091	23	+0.1901	-0.0191
4	+42 543.656	-2 764.75	14	+0.1908	-0.05491	24	+0.1208	-0.04491
5	+58 932.873	-6 431.76	15	+0.6554	-0.04012	25	+0.6544	-0.04013
6	+8 832.553	-24 385.985	16	+0.0221	-0.4351	26	+0.0121	-0.1151
7	+3 246.432	-35 893.0093	17	+0.05021	-0.982	27	+0.03021	-0.582
8	+4 098.9083	-87 632.726	18	+0.0982	-0.7461	28	+0.03982	-0.74361
9	+1 641.763	-64 634.983	19	+0.0931	-0.6523	29	+0.20931	-0.62523
10	+5 364.623	-87 726.872	20	+0.09142	-0.8734	30	+0.03142	-0.8434

4. Содержание отчёта:

4.1 Перевод чисел А и В своего варианта в 2-ю, 8-ю, 16-ю системы счисления.

4.2 Перевод дробных чисел своего варианта в 2-ю, 8-ю, 16-ю системы счисления.

4.3 Представление данных чисел в форме с фиксированной и плавающей запятой.

5. Контрольные вопросы:

5.1 Понятие системы счисления. Представление чисел в различных системах счисления.

5.2 Перевод в различные системы счисления

5.3 Перевод дробного числа в 2, 8, 16-ю системы счисления.

5.4 Представление чисел с фиксированной запятой.

6. Список литературы

Основные источники:

1. Максимов Н. В. Архитектура ЭВМ и вычислительные системы: учебник / Н. В. Максимов, Т. Л. Партыка, И. И. Попов. - 4-е изд., перераб. и доп. - М.: ФОРУМ, 2012. - 512
2. Просветов Г.И. Дискретная математика: задачи и решения. – М.: Бином. Лаборатория знаний, 2010. – 222с.

3. Партыка Т. Л. Периферийные устройства вычислительной техники : учеб. пособие / Т. Л. Партыка, И. И. Попов. - 3-е изд., испр. и доп. - М. : ФОРУМ, 2012. - 432 с.
4. Юров В.И. Assembler. Учебник для вузов, 2-е издание. – СПб.: Питер, 2010. – 637с.

Дополнительная литература

1. Келим Ю.М. Вычислительная техника.- М.:Академия,2007. - 384с.
2. Мышляева И.М.Цифровая схемотехника.- М.:Академия,2005. - 400с.
3. Вычислительная техника: метод. пособие /авт.-сост. Б.В. Цыбаков.- М., 2004. - 112с.

Тема 1.4. Выполнение арифметических операций

Практическая работа №2. Выполнение арифметических операций в обратном и дополнительных кодах.

Время, отводимое на выполнение практической работы - 2 часа.

1. **Цель занятия:** Получение и закрепление умений и навыков выполнения арифметических операций сложения и вычитания в двоичной системе счисления, используя шестнадцатиразрядную сетку и представление отрицательных чисел в дополнительном коде, выполнения арифметических операций в двоичной системе счисления с действительными числами в нормальной форме.

2. Основные теоретические положения:

2.1 Машинные коды ЭВМ

Для выполнения операций над двоичными числами в ЭВМ предусмотрены специальные машинные коды. В первую очередь, их применение обусловлено необходимостью выполнения арифметических операций над числами, имеющими разные знаки. Кроме того, реализовать, например, операцию вычитания двоичных чисел значительно проще, если заменить ее операцией сложения с отрицательным числом, а операнды (слагаемые) преобразовать в соответствующий код. Различают несколько вариантов машинных кодов, самые популярные из них – это **прямой код**, **обратный код** и **дополнительный код**. Однако наряду с ними применяются также **модифицированные обратный** и **дополнительный коды**.

Прямой код двоичного числа образуется, если перед старшим разрядом абсолютного значения этого числа добавить знаковый разряд, единица в котором означает, что это отрицательное число, нуль, что положительное.

Пример 2.1 Прямой код положительного (отрицательного) числа

$C_{10} = +13$	$C_2 = +1101$	$[C_2]_{пр} = 0 1101$
$C_{10} = -11$	$C_2 = -1011$	$[C_2]_{пр} = 1 1011$

Для получения **обратного кода** чисел существует два правила. Обратный код положительного числа совпадает с его прямым кодом и содержит нуль в знаковом разряде. Обратный код отрицательного числа содержит единицу в знаковом разряде, а у значащих цифр нули заменяются единицами, а единицы нулями, именно поэтому этот код и получил своё название.

Пример 2.2 Обратный код положительного (отрицательного) числа

$C_{10} = +14$	$C_2 = +1110$	$[C_2]_{обр} = 0 1110$
$C_{10} = -10$	$C_2 = -1010$	$[C_2]_{обр} = 1 0101$

Обратный код имеет несколько важных свойств:

- сложение положительного числа с его отрицательным значением в обратном коде даёт **машинную единицу обратного кода** $ME_{обр} = 1|111...111$;
- нуль в обратном коде может быть как положительным - $0|000...000$, так и отрицательным - $1|111...111$ числом.

Неоднозначность в определении нуля в обратном коде является основной причиной того, что он не используется в современных ЭВМ.

Дополнительный код числа получают по следующим правилам. Дополнительный код положительного числа совпадает с его прямым кодом. Дополнительный код отрицательного числа получается прибавлением к младшему разряду обратного кода числа единицы⁶.

⁶ Как для целых, так и для дробных чисел.

Пример 2.3 Дополнительный код положительного (отрицательного) числа

$C_{10} = +18$	$C_2 = +10010$	$[C_2]_{доп} = 0 10010$
$C_{10} = -13$	$C_2 = -1101$	$[C_2]_{доп} = 1 0010 + 0001 = 1 0011$
$C_{10} = -0.625$	$C_2 = -0.101$	$[C_2]_{доп} = 1 0.101 + 0.001 = 1 0.110$

Дополнительный код имеет несколько важных свойств:

- сложение дополнительных кодов положительного и отрицательного чисел дает машинную единицу дополнительного кода $ME_{доп} = ME_{обр} + 2^0 = 10|000...000$;
- нуль в дополнительном коде имеет единственное представление.

Свое название дополнительный код получил, потому что для представления отрицательного числа в этом коде требуется дополнение прямого кода этого числа до $ME_{доп}$.

2.2. Правила десятичной арифметики

Как уже отмечалось, операция вычитания в двоичном коде заменяется операцией сложения с отрицательным числом, поэтому рассмотрим на примерах правила сложения двух положительных, положительного и отрицательного, отрицательного и положительного и двух отрицательных чисел. Вообще, операция сложения, наряду с операцией сдвига, являются основными, т.к. помимо вычитания, к ним сводятся и операции умножения и операции деления двоичных чисел.

Таблица 7 Правила поразрядного сложения двоичных чисел

a_i	b_i	p_{i-1}	S_i	P_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Сложение двоичных чисел происходит поразрядно в соответствии с таблице 2.1. Особенностью сложения разрядов a_i и b_i двух двоичных чисел является наличие переноса из предыдущего разряда p_{i-1} . Результат сложения помещается в S_i , а возникший перенос в P_i .

Пример 2.4 Сложение двух положительных чисел $A_{10} = +18$ и $B_{10} = +7$ в прямом, обратном и дополнительном кодах

Представляем числа в двоичном коде $A_2 = +10010$ и $B_2 = +111$.

Выравниваем разрядную сетку чисел, т.е. приписываем перед старшим разрядом B_2 (так как оно имеет всего три разряда, а A_2 - пять) два нуля $A_2 = 0|10010$ и $B_2 = 0|00111$.

Запишем прямой, обратный и дополнительный коды обоих чисел:

$$[A_2]_{пр} = [A_2]_{обр} = [A_2]_{доп} = 0|10010 \quad [B_2]_{пр} = [B_2]_{обр} = [B_2]_{доп} = 0|00111$$

Сложение в прямом, обратном и дополнительном кодах даст одинаковый результат

R_2 :

$$\begin{array}{r} 0|10010 \\ + \\ 0|00111 \\ \hline 0|11001 \end{array}$$

$$R_2 = 0|11001$$

$$R_{10} = +25$$

Пример 2.5 Сложение положительного числа $A_{10} = +18$ и отрицательного $B_{10} = -7$ в прямом, обратном дополнительном кодах

Представляем числа в двоичном коде $A_2 = +10010$ и $B_2 = -111$.

Выравниваем разрядную сетку чисел $A_2 = 0|10010$ и $B_2 = 1|00111$.

Запишем прямой, обратный и дополнительный коды обоих чисел:

$$\begin{array}{lll} [A_2]_{пр} = 0|10010 & [A_2]_{обр} = 0|10010 & [A_2]_{дон} = 0|10010 \\ [B_2]_{пр} = 1|00111 & [B_2]_{обр} = 1|11000 & [B_2]_{дон} = 1|11001 \end{array}$$

Сложение в обратном коде даст результат R_2 , а в дополнительном T_2 :

$$\begin{array}{r} 0|10010 \\ + \\ 1|11000 \\ \hline 0|01010 \\ + \\ 0|00001 \\ \hline 0|01011 \end{array} \quad \begin{array}{l} R_2 = 0|01011 \\ R_{10} = +11 \end{array}$$

Обратим внимание на сложение в обратном коде, где получен перенос в знаковый разряд, это означает, что необходимо прибавить единицу в младший разряд суммы.

В дополнительном коде также получен перенос в знаковый разряд, однако в этом случае единица теряется. Это обстоятельство является различием операций сложения в обратном и дополнительном кодах.

$$\begin{array}{r} 0|10010 \\ + \\ 1|11001 \\ \hline 0|01011 \end{array} \quad \begin{array}{l} T_2 = 0|01011 \\ T_{10} = +11 \end{array}$$

Сложение отрицательного числа с положительным производится аналогично процедуре описанной в примере 1.6.

Сложение двух отрицательных чисел производится поразрядно, согласно процедуре, описанной в примере 1.7, знак суммы в этом случае будет «-», т.е. в знаковом разряде получается 1.

Пример 2.6 Сложение и вычитание двоичных чисел в 16-ти разрядной сетке, используя представление отрицательных чисел в дополнительном коде.

$$\begin{array}{lll} 1533 = & 00000101 & 11111101 \\ +7347 = & 00011101 & 00010111 \\ 8880 & 00100010 & 10110000 \end{array}$$

$$\begin{array}{lll} 1533 = & 00000101 & 11111101 \\ - 7347 = & 11100011 & 01001101 \\ -5814 = & 11101001 & 01001010 \end{array}$$

$$\begin{array}{lll} - 1533 = & 11111010 & 00000011 \\ + 7347 = & 00011100 & 10110011 \\ 6014 = & 00010111 & 01111110 \end{array}$$

$$\begin{array}{lll} - 1533 = & 11111010 & 00000011 \\ - 7347 = & 11100011 & 01001101 \\ - 8880 = & 11011101 & 01010000 \end{array}$$

2.3. Правила десятичной арифметики. Числа с плавающей точкой

Арифметические операции над числами с плавающей точкой в ЭВМ обрабатываются в специальных блоках и хранятся в памяти парами – мантисса/порядок в нормализованном виде, причём мантиссы и порядки обрабатываются отдельно. Проведение операций **сложения (вычитания)** рассмотрим на примерах.

Пример 2.7 Сложение (вычитание) чисел $A_{10} = +2.5$ и $B_{10} = -1.25$

Представляем числа в двоичном коде $A_2 = 0|10.1$ и $B_2 = 1|1.01$.

Порядок A_2 равен $P_{A2} = 0|1$. Мантисса A_2 равна $M_{A2} = 0|01$.

Порядок B_2 равен $P_{B2} = 0|1$. Мантисса B_2 равна $M_{B2} = 1|001$.

При сложении (вычитании) чисел, которые имеют одинаковые порядки исходных чисел, как в данном примере, их мантиссы складываются (вычитаются), а результату присваивается их общий порядок. Для сложения необходимо произвести выравнивание порядков мантисс, т.е. $M_{A2} = 0|010$. Затем получим дополнительный код второго слагаемого $[B_2]_{\text{доп}} = 1|111$. Произведём операцию сложения мантисс $[M_{A2}]_{\text{доп}} + [M_{B2}]_{\text{доп}}$:

$$\begin{array}{r} 0|010 \\ + \\ 1|111 \\ \hline 0|001 = [M_{R2}] \end{array}$$

Порядок результата $P_{R2} = 0|1$, а мантисса $M_{R2} = 0|001$. Т.е. $R_{10} = +0.125 \cdot 10^1 = 1.25$.

Пример 2.8 Сложение (вычитание) чисел $A_{10} = +1.25$ и $B_{10} = -0.75$

Представляем числа в двоичном коде $A_2 = 0|1.01$ и $B_2 = 1|0.11$.

Порядок A_2 равен $P_{A2} = 0|1$. Мантисса A_2 равна $M_{A2} = 0|101$.

Порядок B_2 равен $P_{B2} = 0|0$. Мантисса B_2 равна $M_{B2} = 1|11$.

При сложении (вычитании) чисел, которые имеют не одинаковые порядки исходных чисел, как в данном примере, то порядки чисел сначала выравниваются, для этого число с меньшим порядком сдвигается вправо на разницу порядков. Поэтому сдвигаем второе число вправо на единицу $M_{B2}' = 1|011$. Порядок результата берётся равным порядку большего числа, т.е. в данном случае $P_{R2} = 0|1$. Для сложения необходимо получить дополнительный код второго слагаемого $[B_2']_{\text{доп}} = 1|101$. Произведём операцию сложения мантисс $[M_{A2}]_{\text{доп}} + [M_{B2}']_{\text{доп}}$:

$$\begin{array}{r} 0|101 \\ + \\ 1|101 \\ \hline 0|010 = [M_{R2}] \end{array}$$

Мантисса результат $M_{R2} = 0|010$, т.к. мантисса числа не нормализована⁷, сдвигаем его на один разряд влево и вычитаем из порядка единицу. $P_{R2} = 0|0$, а мантисса $M_{R2} = 0|10$ $R_{10} = +0.5 \cdot 10^0 = 0.5$.

Операции **умножения (деления)** чисел с плавающей точкой производятся аналогично сложению и вычитанию раздельно над порядками и мантиссами. Имеется несколько общих правил умножения (деления) чисел с плавающей точкой. Во-первых, при умножении (делении) порядки складываются (вычитаются), как мы и привыкли в десятичной системе счисления. Во-вторых, мантиссы перемножаются (делятся). И, наконец, знак произведения (частного) формируется сложением знаковых разрядов сомножителей (делимого и делителя) по модулю два⁸. Рассмотрим примеры операций умножения (вычитания).

⁷ Старший разряд мантиссы равен нулю.

⁸ Возникающие переносы просто игнорируются.

Пример 2.9 Умножение чисел $A_{10} = +1.5$ и $B_{10} = -2$

Представляем числа в двоичном коде $A_2 = 0|1.1$ и $B_2 = 1|10$. Порядок A_2 равен $P_{A2} = 0|1$. Мантисса A_2 равна $M_{A2} = 0|11$. Порядок B_2 равен $P_{B2} = 0|1$. Мантисса B_2 равна $M_{B2} = 1|10$. Порядок произведения $P_{R2} = P_{A2} + P_{B2} = 1 + 1 = 2$. Мантисса произведения $M_{R2} = M_{A2} \cdot M_{B2} = 11 \cdot 10 = 110$. Определим знак произведения $0 + 1 = 1$. Тогда $R_2 = 1|110 \cdot 10^2 = 1|11.0$ или $R_{10} = -3$.

Результат, переведённый в десятичную систему счисления, совпадает с результатом, который можно получить, если перемножить A_{10} и B_{10} .

Пример 2.10 Деление чисел $A_{10} = +15$ и $B_{10} = -5$

Представляем числа в двоичном коде $A_2 = 0|1111$ и $B_2 = 1|101$. Порядок A_2 равен $P_{A2} = 0|2$. Мантисса A_2 равна $M_{A2} = 0|1111$. Порядок B_2 равен $P_{B2} = 0|1$. Мантисса B_2 равна $M_{B2} = 1|101$. Порядок частного $P_{R2} = P_{A2} - P_{B2} = 2 - 1 = 1$. Мантисса частного $M_{R2} = M_{A2} : M_{B2} = 1111 / 1010 = 1.1$. Определим знак частного $0 + 1 = 1$. Тогда $R_2 = 1|1.1 \cdot 10^1 = 1|11$ или $R_{10} = -3$. Результат, переведённый в десятичную систему счисления, совпадает с результатом, который можно получить, если разделить A_{10} на B_{10} .

3. Задание к работе

3.1 Используя результаты перевода чисел A и B из десятичной системы в двоичную, восьмеричную и шестнадцатеричную системы счисления предыдущей практической работы вычислить в двоичной системе счисления $A+B$, $A-B$, $B-A$, $-A-B$, используя шестнадцатиразрядную сетку и представление отрицательных чисел в дополнительном коде. Результаты перевести в десятичную систему счисления.

Таблица 8 Варианты заданий (вариант определяется по номеру в журнале)

Вариант	A	B	Вариант	A	B
1	174	6936	16	342	5456
2	192	6893	17	353	5345
3	319	5789	18	364	5234
4	224	6678	19	375	5123
5	231	6567	20	386	5012
6	242	6456	21	397	4901
7	253	6345	22	408	4890
8	264	6234	23	419	4789
9	275	6123	24	430	4678
10	286	6012	25	441	4665
11	297	5901	26	452	4657
12	308	5890	27	463	4639
13	207	6789	28	474	4626
14	320	5678	29	485	4613
15	331	5567	30	496	4600

3.2. Правила недесятичной арифметики (с числами с плавающей запятой)

- 3.2.1. Выполнить сложение двух чисел A и B . Результат переведите в 2-ю систему счисления.
- 3.2.2. Выполнить умножение двух чисел A и B . Результат переведите в 2-ю систему счисления.
- 3.2.3. Выполнить деление двух чисел A и B (большее на меньшее). Результат переведите в 2-ю систему счисления.

Таблица 9 Варианты заданий для чисел с фиксированной и плавающей запятой

№ в.	A	B	№ в.	A	B	№ в.	A	B
1	+23 411.207	-2 345.22	11	+0.221	-0.00104	21	+0.2321	-0.0404
2	+33 813.011	-3 732.515	12	+0.8754	-0.0123	22	+0.8701	-0.0223
3	+18 622.192	-4 345.01	13	+0.9901	-0.0091	23	+0.1901	-0.0191
4	+42 543.656	-2 764.75	14	+0.1908	-0.05491	24	+0.1208	-0.04491
5	+58 932.873	-6 431.76	15	+0.6554	-0.04012	25	+0.6544	-0.04013
6	+8 832.553	-24 385.985	16	+0.0221	-0.4351	26	+0.0121	-0.1151
7	+3 246.432	-35 893.0093	17	+0.05021	-0.982	27	+0.03021	-0.582
8	+4 098.9083	-87 632.726	18	+0.0982	-0.7461	28	+0.03982	-0.74361
9	+1 641.763	-64 634.983	19	+0.0931	-0.6523	29	+0.20931	-0.62523
10	+5 364.623	-87 726.872	20	+0.09142	-0.8734	30	+0.03142	-0.8434

4. Содержание отчёта:

4.1 Выполненные операции в двоичной системе счисления $A+B$, $A-B$, $B-A$, $-A-B$, используя шестнадцатиразрядную сетку и представление отрицательных чисел в дополнительном коде.

4.2 Выполненные операции в двоичной системе счисления $A+B$, $A*B$, B/A или A/B чисел с плавающей точкой.

5. Контрольные вопросы:

5.1 Машинные коды ЭВМ.

5.2 Правила десятичной арифметики

5.3 Правила десятичной арифметики с числами с плавающей запятой

6. Список литературы

Основная литература

1. Максимов Н. В. Архитектура ЭВМ и вычислительные системы: учебник / Н. В. Максимов, Т. Л. Партыка, И. И. Попов. - 4-е изд., перераб. и доп. - М.: ФОРУМ, 2012. - 512
2. Просветов Г.И. Дискретная математика: задачи и решения. – М.: Бином. Лаборатория знаний, 2010. – 222с.
3. Партыка Т. Л. Периферийные устройства вычислительной техники : учеб. пособие / Т. Л. Партыка, И. И. Попов. - 3-е изд., испр. и доп. - М. : ФОРУМ, 2012. - 432 с.
4. Юров В.И. Assembler. Учебник для вузов, 2-е издание. – СПб.: Питер, 2010. – 637с.

Дополнительная литература

1. Горнец Н.Н. Организация ЭВМ и систем. - М.: Академия, 2008. - 320с.
2. Келим Ю.М. Вычислительная техника. - М.: Академия, 2007. - 384с.
3. Мышляева И.М. Цифровая схемотехника. - М.: Академия, 2005. - 400с.
4. Вычислительная техника: метод. пособие / авт.-сост. Б.В. Цыбаков. - М., 2004. - 112с.

Раздел 2. Логические основы ЭВМ

Тема 2.1 Логические функции и схемы

Практическое занятие №3. Изучение номенклатуры ИМС логических элементов в программе Electronic WorkBench

Время, отводимое на выполнение практической работы - 2 часа.

Перечень необходимых технических средств обучения:

- ✓ персональные компьютеры Core i3 3.0, 4 Gb, 4 Гб ОЗУ, 250 Gb HDD;
- ✓ локальная сеть;
- ✓ коммутатор для подключения в сети Internet.

Перечень необходимых программных средств обучения:

- ✓ ОС Windows XP (7);
- ✓ Программа для моделирования Electronic Work Bench 5.12 и выше

1. **Цель занятия:** Ознакомление с возможностями программы анализа электронных и логических схем. Изучение интерфейса пользователя. Ознакомление с терминами и понятиями. Построение простейших электронных и логических схем.

2. Основные теоретические положения:

2.1 Структура окна и система меню

Окно содержит строку команд меню, строку основных типовых электронных устройств, поле для составления исследуемой схемы и полосы управления прокруткой. Первые команды меню типовые и пояснений не требуют.

Таблица 10 Основные команды меню

Меню Circuit	
Rotate	вращение выделенного компонента (Ctrl + R)
Flip Horizontal	перевернуть по горизонтали
Flip Vertical	перевернуть по вертикали
Components Properties	свойства компонента
Create SubCircuit	(Ctrl + B) - объединяет выделенные элементы схемы в подсхему и помещает ее(полученную подсхему) под выбранным именованием в окно Custom
Меню Analysis	
Activate	запуск моделирования (Ctrl + Q)
Pause	прерывание моделирования (F9)
Stop	остановка моделирования (Ctrl + T)

2.2 Технология создания схем

Для открытия нужной библиотеки компонентов нужно подвести курсор мыши к соответствующей иконке и нажать один раз её левую кнопку. После размещения компонентов производится соединение их выводов проводниками. Для выполнения подключения курсор мыши подводится к выводу компонента и нажимается левая кнопка и протягивается к выводу другого компонента, после чего кнопка мыши отпускается и соединение готово.

2.3 Основные компоненты

- ✓ группа SOURCES

—  заземление (метка) **Ground**

-  источник фиксированного напряжения **Battery**
✓ **группа BASIC**
-  точка соединения проводников, используется также для введения на схему надписей длиной не более 14 символов
-  переключатель **Switch**, управляемый нажатием задаваемой клавишей клавиатуры (в квадратных скобках), по умолчанию - клавиша пробела
✓ **группа INDICATORS**
-  **светоиндикатор** (свет свечения может быть настроен красным, зелёным и синим)
✓ **группа LOGIC GATES**
-  логический элемент "И"
-  логический элемент "ИЛИ"
-  логический элемент "НЕ"
-  логический элемент "ИЛИ-НЕ"
-  – логический элемент "И-НЕ"
-  – логический элемент
-  – логический элемент "ИЛИ-НЕ"
-  – буфер

3. Задание к работе

3.1 Собрать схему логического элемента "ИЛИ" для четырёх входных сигналов.

- 3.1.1 В группе **Logic Gates** требуется выбрать логический элемент **ИЛИ**.
- 3.1.2 К выходу логического элемента присоединяем из группы **INDICATORS** красный светодиод.
- 3.1.3 Для получения логического сигнала (0 или 1) воспользуемся источником напряжения **Battery** и переключателем **Switch**
- 3.1.4 Затем набираем 4 источника и 4 переключателя, при этом присваиваем каждому переключателю клавишу переключения (**q, w, e, r**).
- 3.1.5 Затем соединяем входы логической схемы "ИЛИ" с каждым из переключателей.
- 3.1.6 Проверьте работу схемы. Проверка состоит в подаче различных кодовых комбинаций на вход логической схемы. На выходе логической схемы "ИЛИ" появляется логическая 1 (горит светодиод) при подаче логических 1 (потенциал 5 вольт) хотя бы от одного из четырех входов логической схемы "ИЛИ"

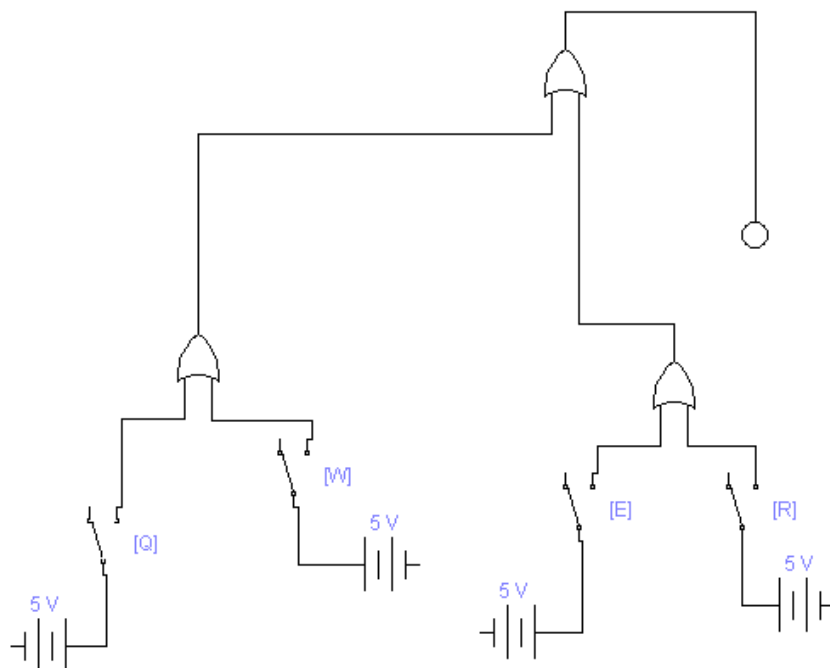


Рисунок 1 Схема 4-ИЛИ

- 3.1.7 Добавьте заземление к каждому элементу питания.
- 3.1.8 Создайте подсхему, которую назовите **OR** (для этого выделите схему и выберите в меню **Circuit - Create SubCircuit**).

3.2 Собрать схему логического элемента "И" для четырёх входных сигналов.

- 3.2.1 При выборе элемента «И». Двумя щелчками мыши на изображении логического элемента переходим к настройкам параметров логического элемента "И". Выбираем количество входов = 4.
- 3.2.2 Присвойте название логическому элементу **AND_4**.
- 3.2.3 Присвойте каждому переключателю клавишу переключения (1, 2, 3, 4)
- 3.2.4 Добавьте индикаторы A, B, C и D.
- 3.2.5 Постройте схему и проверьте её работу:

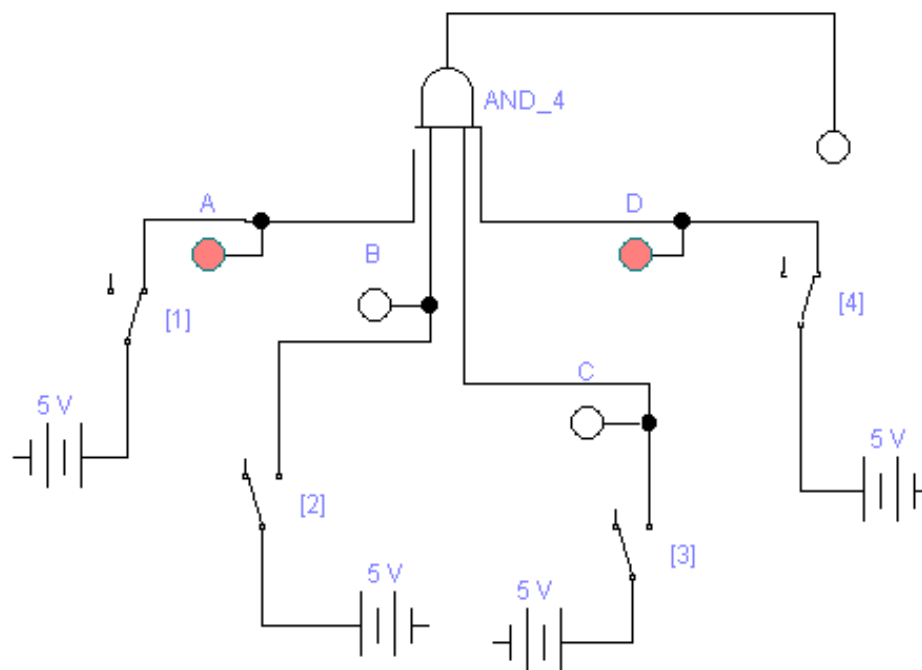


Рисунок 2 Схема 4-И

- 3.2.6 На основании созданной схемы создайте подсхему, которую назовите **And**.
- 3.3 Собрать схему логического элемента "ИЛИ-НЕ" для четырёх входных сигналов. Проверьте работу построенной схемы. Создайте подсхему **Or_No**.
- 3.4 Собрать схему логического элемента "И-НЕ" для четырёх входных сигналов. Проверьте работу построенной схемы. Создайте подсхему **And_No**. Сохраните документ.
- 3.5 Собрать схему логического элемента "XOR" – исключающее ИЛИ ($x \oplus y$) для четырёх входных сигналов. Проверьте работу построенной схемы. Создайте подсхему **XOR**. Сохраните документ.
- 3.6 Создайте новый документ. Собрать схемы, предложенные ниже. С помощью ключей (управляются клавишами) подать на вход схемы различные комбинации переменных A, B и C. Значения функции F занести в таблицу 9:

Таблица 11 Результаты эксперимента

A	B	C	F1	F2	F3	F4
0	0	0				
1	0	0				
0	1	0				
0	0	1				
1	1	0				
1	0	1				
0	1	1				
1	1	1				

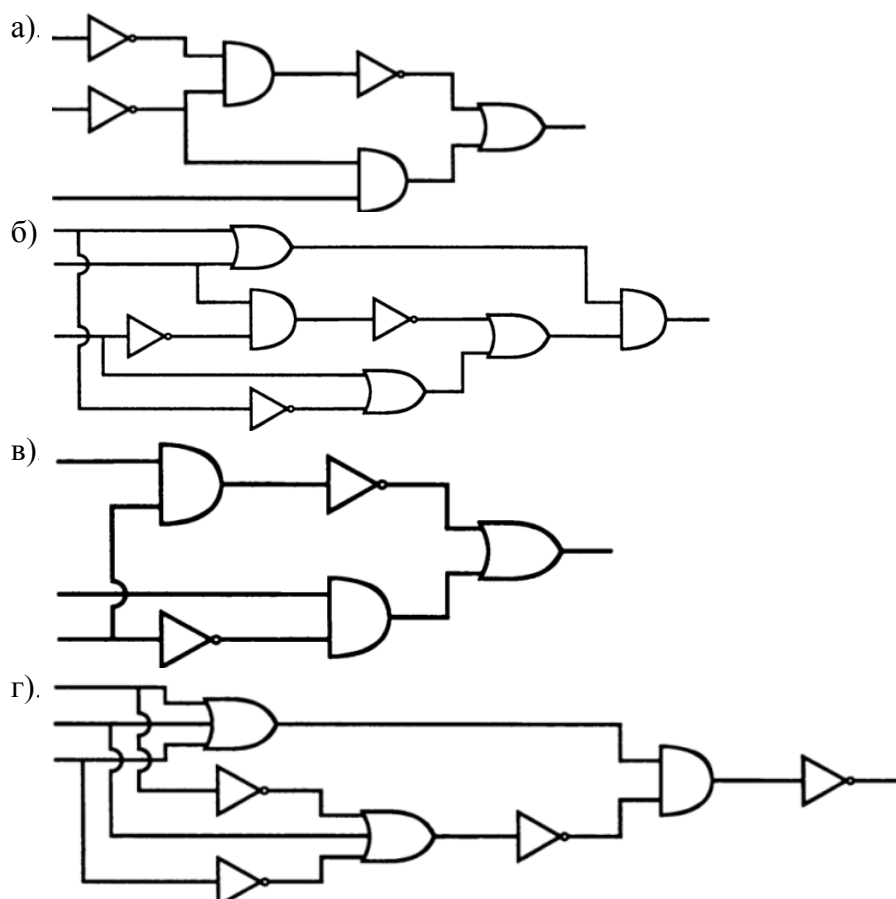


Рисунок 3. Схемы для построения

4. Содержание отчёта:

4.1 Собранные схемы элементов: 4-ИЛИ, 4-И, 4-ИЛИ-НЕ, 4-И-НЕ, 4-XOR

4.2 Заполненная таблица 9 как результат работы собранных схем 3.6 а) б) в) г).

5. Контрольные вопросы:

5.1 Какие логические элементы имеются в библиотеке EWB?

5.2 Перечень основных элементов из библиотеки EWB, необходимый для моделирования логических схем и цифровых устройств ЭВМ.

5.3 Как производится соединение более двух входов или выходов между собой?

5.4 Какой командой можно скопировать изображение схемы в отчёт по лабораторной работе, подготавливаемой в текстовом редакторе WORD.

5.5 С помощью каких элементов можно смоделировать подачу логической 1 и логического 0?

6. Список литературы

Основная литература

1. Максимов Н. В. Архитектура ЭВМ и вычислительные системы: учебник / Н. В. Максимов, Т. Л. Партыка, И. И. Попов. - 4-е изд., перераб. и доп. - М.: ФОРУМ, 2012. - 512
2. Просветов Г.И. Дискретная математика: задачи и решения. – М.: Бином. Лаборатория знаний, 2010. – 222с.
3. Партыка Т. Л. Периферийные устройства вычислительной техники : учеб. пособие / Т. Л. Партыка, И. И. Попов. - 3-е изд., испр. и доп. - М. : ФОРУМ, 2012. - 432 с.
4. Юров В.И. Assembler. Учебник для вузов, 2-е издание. – СПб.: Питер, 2010. – 637с.

Дополнительная литература

1. Ижевский А.С., Калиниченко Б.Б. Методические указания для лабораторных работ «Моделирование и анализ электрических схем на основе логических элементов Electronic Workbench», 2009. - 38с.
2. Келим Ю.М. Вычислительная техника.- М.:Академия,2007. - 384с.
3. Вычислительная техника: метод. пособие /авт.-сост. Б.В. Цыбаков.- М., 2004. - 112с.

Практическая работа №4. Построение комбинационных схем по переключательным функциям в разных базисах

Время, отводимое на выполнение практической работы - 2 часа.

Перечень необходимых технических средств обучения:

- ✓ персональные компьютеры Core i3 3.0, 4 Gb, 4 Гб ОЗУ, 250 Gb HDD;
- ✓ локальная сеть;
- ✓ коммутатор для подключения в сети Internet.

Перечень необходимых программных средств обучения:

- ✓ ОС Windows XP (7);
- ✓ Программа для моделирования Electronic Work Bench 5.12 и выше

1. **Цель занятия:** Научиться строить комбинационные схемы по переключательным функциям в разных базисах.

2. **Основные теоретические положения:**

Комбинационная схема (КС) – это схема, в которой выходные сигналы $Y=(y_1, \dots, y_m)$ в определённый момент времени однозначно определяются совокупностью входных сигналов $X=(x_1, \dots, x_n)$, поданных в тот же момент времени t .

В данном случае название **комбинационный** обусловлено тем, что выход КС определяется комбинацией входных сигналов.

Особенность КС заключается в том, что выход формируется в момент поступления входных сигналов. Это обстоятельство обуславливает высокое быстродействие данного типа схем.

Пример построения схемы переключательной функции $f=ab'+c$ в различных базисах:

а. В базисе **И, ИЛИ, НЕ**:

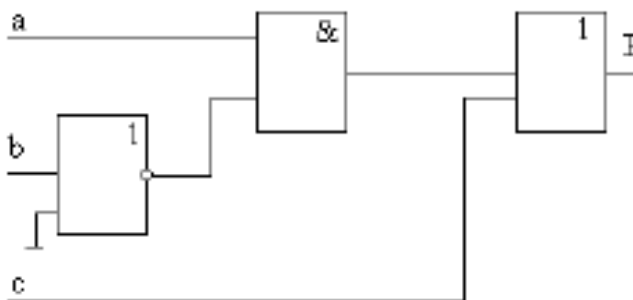


Рисунок 8. Схема функции $f=ab'+c$ в базисе И, ИЛИ, НЕ

б. В базисе **ИЛИ, НЕ**: $f=\overline{\overline{a}b} \vee c = \overline{\overline{a} \vee b} \vee c$

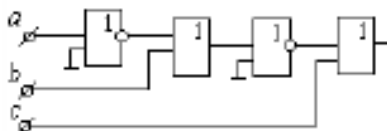


Рисунок 9. Схема функции $f=ab'+c$ в базисе ИЛИ, НЕ

с. В базисе **И, НЕ**: $f=\overline{\overline{a}b \cdot c}$

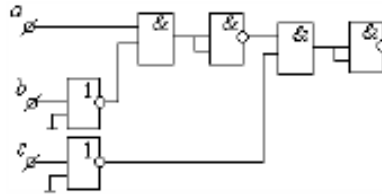


Рисунок 10. Схема функции $f=ab'+c$ в базисе И, НЕ

d. В базисе **И-НЕ**: $f = \overline{\overline{ab} \vee c}$, $f = \overline{a\overline{b} \cdot c}$

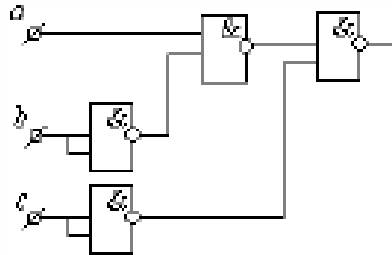


Рисунок 11. Схема функции $f=ab'+c$ в базисе И-НЕ

e. В базисе **ИЛИ-НЕ**: $f = \overline{a \vee b \vee c}$

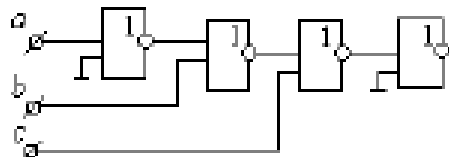


Рисунок 12. Схема функции $f=ab'+c$ в базисе ИЛИ-НЕ

3. Задание к работе

Постройте комбинационные схемы соответствующие булевым выражениям в различных базисах (номер варианта определяется по последней цифре номера в журнале):

- а) И, ИЛИ, НЕ
- б) И, НЕ
- в) ИЛИ, НЕ
- г) И-НЕ
- д) ИЛИ-НЕ

- 1) $f=(x'+y)(x+yz)$
- 2) $f=(xy')+((yz')+(x'z))$
- 3) $f=(y'z')+((xy)'(y'z))$
- 4) $f=(xy)+((xy'z')+z)'$
- 5) $f=(xy')'+(yz)'$
- 6) $f=((x+y)z)'$
- 7) $f=(x'(y+x'y'))$
- 8) $f=xy'+zx$
- 9) $f=(x'y+(y'z))'$
- 10) $f=((xy')'+(x'z))$

4. Содержание отчёта:

Комбинационные схемы переключательной функции в различных базисах.

5. Контрольные вопросы:

5.1 Аксиомы и теоремы алгебры логики.

- 5.2 Функционально-полные системы элементов.
- 5.3 Построение схемы в базисах И,НЕ; ИЛИ,НЕ. Закон де-Моргана.
- 5.4 Построение схемы в базисах И-НЕ; ИЛИ-НЕ. Закон де-Моргана, закон двойного отрицания

6. Список литературы

Основная литература

1. Максимов Н. В. Архитектура ЭВМ и вычислительные системы: учебник / Н. В. Максимов, Т. Л. Партыка, И. И. Попов. - 4-е изд., перераб. и доп. - М.: ФОРУМ, 2012. - 512
2. Просветов Г.И. Дискретная математика: задачи и решения. – М.: Бином. Лаборатория знаний, 2010. – 222с.
3. Партыка Т. Л. Периферийные устройства вычислительной техники : учеб. пособие / Т. Л. Партыка, И. И. Попов. - 3-е изд., испр. и доп. - М. : ФОРУМ, 2012. - 432 с.
4. Юров В.И. Assembler. Учебник для вузов, 2-е издание. – СПб.: Питер, 2010. – 637с.

Дополнительная литература

1. Ижевский А.С., Калининченко Б.Б. Методические указания для лабораторных работ «Моделирование и анализ электрических схем на основе логических элементов Electronic Workbench», 2009. - 38с.
2. Келим Ю.М. Вычислительная техника.- М.:Академия,2007. - 384с.
3. Вычислительная техника: метод. пособие /авт.-сост. Б.В. Цыбаков.- М., 2004. - 112с.

Тема 2.2. Синтез логических устройств

Практическая работа №5. Синтез комбинационного логического устройства с применением карт Карно

Время, отводимое на выполнение практической работы - 2 часа.

Перечень необходимых технических средств обучения:

- ✓ персональные компьютеры Core i3 3.0, 4 Gb, 4 Гб ОЗУ, 250 Gb HDD;
- ✓ локальная сеть;
- ✓ коммутатор для подключения в сети Internet.

Перечень необходимых программных средств обучения:

- ✓ ОС Windows XP (7);
- ✓ Программа для моделирования Electronic Work Bench 5.12 и выше

1. **Цель занятия:** Получение навыков построения аналитической формы БФ по заданной таблице истинности, упрощать переключательную функцию с помощью карт Карно, закрепление навыков построения функции в базисах И,ИЛИ,НЕ; И-НЕ; ИЛИ-НЕ.

2. Основные теоретические положения:

2.1 Минимизация логических функций при помощи карт Карно

Карта Карно — графический способ минимизации переключательных (булевых) функций, обеспечивающий относительную простоту работы с большими выражениями и устранение потенциальных гонок. Представляет собой операции попарного неполного склеивания и элементарного поглощения.

Основным методом минимизации логических функций, представленных в виде СДНФ или СКНФ является операция попарного неполного склеивания и элементарного поглощения. Операция попарного склеивания осуществляется между двумя термами (членами), содержащими одинаковые переменные, входящие которых (прямые и инверсные) совпадают для всех переменных, кроме одной. В этом случае все переменные, кроме одной, можно вынести за скобки, а оставшиеся в скобках прямое и инверсное входение одной переменной подвергнуть склейке.

Например:

$$\overline{X_1}X_2X_3X_4 \vee \overline{X_1}X_2\overline{X_3}X_4 = \overline{X_1}X_2X_4(X_3 \vee \overline{X_3}) = \overline{X_1}X_2X_4.$$

Возможность поглощения следует из очевидных равенств

$$A \vee \overline{A} = 1; A\overline{A} = 0.$$

Таким образом, главной задачей при минимизации СДНФ и СКНФ является поиск термов, пригодных к склейке с последующим поглощением, что для больших форм может оказаться достаточно сложной задачей. Карты Карно предоставляют наглядный способ отыскания таких термов.

На рисунке 4.1 изображена простая таблица истинности для функции из двух переменных, соответствующий этой таблице 2-мерный куб (квадрат), а также 2-мерный куб с обозначением членов СДНФ и эквивалентная таблица для группировки термов:

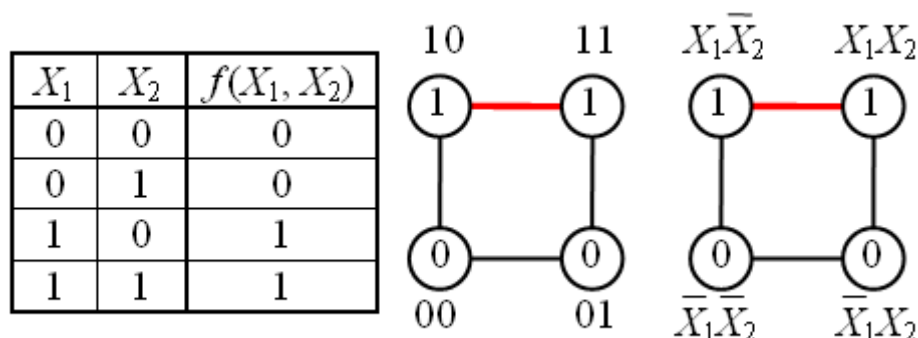


Рисунок 4 Карта Карно для функции от двух переменных

В случае функции трёх переменных приходится иметь дело с трёхмерным кубом. Это сложнее и менее наглядно, но технически возможно. На рисунке 4.2 в качестве примера показана таблица истинности для булевой функции трёх переменных и соответствующий ей куб.

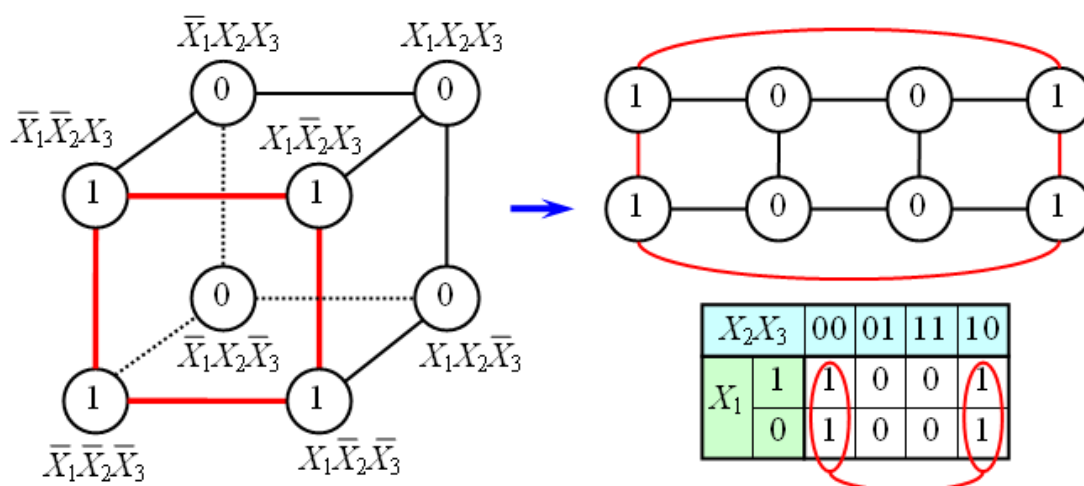


Рисунок 5. Карта Карно для функции от 3-х переменных

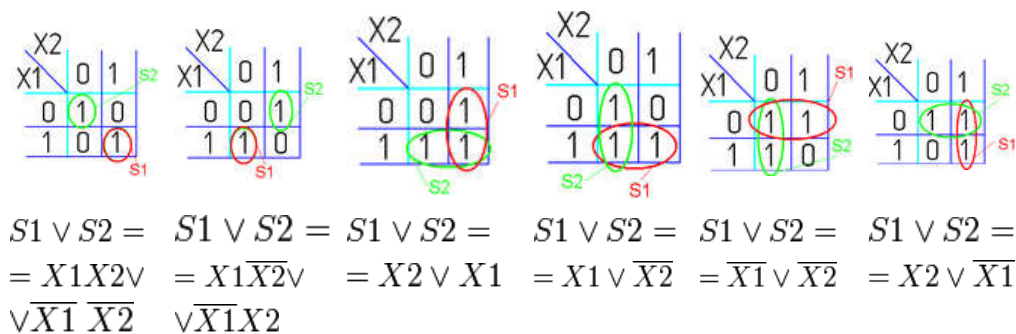
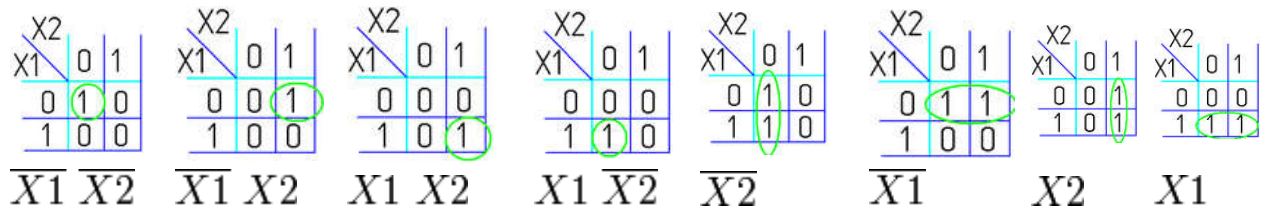
Как видно из рисунка, для трёхмерного случая возможны более сложные конфигурации термов. Например, четыре терма, принадлежащие одной грани куба, объединяются в один терм с поглощением двух переменных:
 $\bar{X}_1\bar{X}_2\bar{X}_3 \vee X_1\bar{X}_2\bar{X}_3 \vee \bar{X}_1\bar{X}_2X_3 \vee X_1\bar{X}_2X_3 =$

Если необходимо получить минимальную ДНФ, то в Карте рассматриваем только те клетки которые содержат единицы, если нужна КНФ, то рассматриваем те клетки которые содержат нули. Сама минимизация производится по следующим правилам (на примере ДНФ):

1. Объединяем смежные клетки содержащие единицы в область, так чтобы одна область содержала 2^n (n целое число $= 0 \dots \infty$) клеток (помним про то что крайние строки и столбцы являются соседними между собой), в области не должно находиться клеток содержащих нули;
2. Область должна располагаться симметрично оси(ей) (оси располагаются через каждые четыре клетки);

3. Не смежные области расположенные симметрично оси(ей) могут объединяться в одну;
4. Область должна быть как можно больше, а кол-во областей как можно меньше;
5. Области могут пересекаться;
6. Возможно несколько вариантов накрытия.

Далее берём первую область и смотрим какие переменные не меняются в пределах этой области, выписываем конъюнкцию этих переменных, если неменяющаяся переменная нулевая, проставляем над ней инверсию. Берём следующую область, выполняем то же самое что и для первой, и т. д. для всех областей. Конъюнкции областей объединяем дизъюнкцией. Например(для Карт на 2-ве переменные):



Для КНФ всё то же самое, только рассматриваем клетки с нулями, не меняющиеся переменные в пределах одной области объединяем в дизъюнкции (инверсии проставляем над единичными переменными), а дизъюнкции областей объединяем в конъюнкцию.

1.2 Пример составления схемы по таблице истинности

Задано $Y=00110101$

1. Строим таблицу истинности логической функции:

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

- а. Запишем функцию в дизъюнктивной нормальной форме:
 $Y=A'BC'+A'BC+AB'C+ABC$
- б. Запишем функцию в виде карты Карно:

AB					
C		00	01	11	10
		0	1	1	0
0	0	0	1	1	0
1	0	0	0	1	1

2. Минимизируем логическую функцию с помощью карты Карно:

- а. На карте Карно выделяем две группы единиц, расположенных в соседних клетках.

AB					
C		00	01	11	10
		0	1	1	0
0	0	0	1	1	0
1	0	0	0	1	1

- б. Минимизированная логическая функция имеет вид: $C = BC' + AC$.

3. Схема на элементах И, ИЛИ, НЕ:

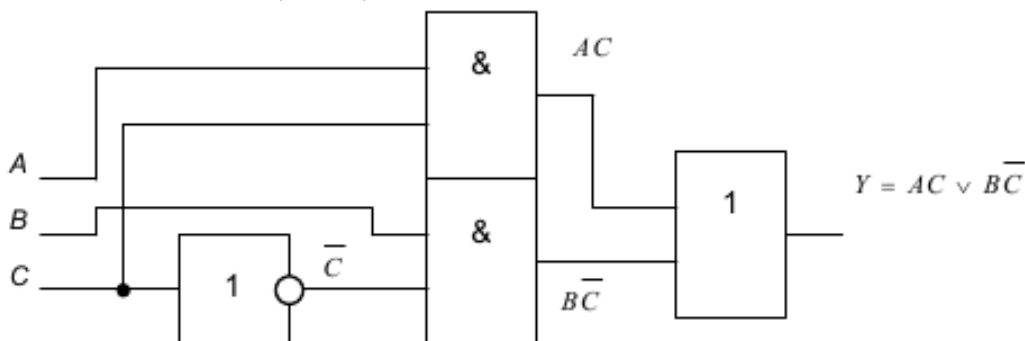


Рисунок 6 Схема на элементах ИЛИ-НЕ

–Схема на элементах И-НЕ:

- а. Для построения схемы преобразуем минимизированное логическое уравнение по закону де Моргана:

$$Y = B \cdot \bar{C} \vee A \cdot C = \overline{(B \cdot \bar{C}) \cdot (A \cdot C)}$$

- б. По полученному уравнению строим схему:

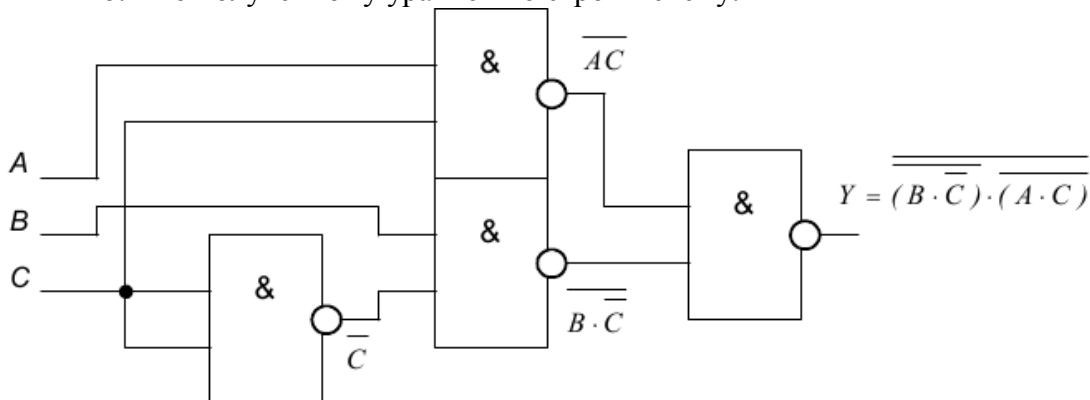


Рисунок 7 Схема на элементах И-НЕ

–Для построения схемы в базисе ИЛИ-НЕ исходное уравнение также требуется преобразовать, используя закон де Моргана, а затем по полученному уравнению построить схему.

3. Задание к работе

- 1.1 В последнюю строку таблицы истинности вписать заданное восьмиразрядное двоичное число Y . Представить логическую функцию, заданную таблицей истинности, в СДНФ (СКНФ) и в виде карты Карно.
- 1.2 Минимизировать рассматриваемую логическую функцию двумя способами: аналитическим и с помощью карты Карно.
- 1.3 Построить схему, реализующую заданную на наборе логических элементов:
 - И (AND), ИЛИ (OR), НЕ (NOT).
 - И-НЕ (NAND).
 - ИЛИ-НЕ (NOR).

Таблица 12 Варианты заданий (вариант определяется по номеру в журнале)

Вариант	Y	Вариант	Y
1	01101110	16	11110001
2	11011100	17	11100011
3	10111001	18	11000111
4	01110011	19	10001111
5	11100110	20	00011111
6	11001101	21	00111110
7	10011011	22	01111100
8	10101110	23	11110100
9	01011101	24	11101001
10	10111010	25	11010011
11	01110101	26	10100111
12	11101010	27	01001111
13	11010100	28	10011110
14	10101001	29	00111101
15	11111000	30	01111010

4. Содержание отчёта:

- 4.1 Представление логической функции, заданной таблицей истинности, в СДНФ (СКНФ) и в виде карты Карно.
- 4.2 Минимизация рассматриваемой логической функции двумя способами: аналитическим и с помощью карты Карно.
- 4.3 Построение схемы, реализующую заданную на наборе логических элементов:
 - И (AND), ИЛИ (OR), НЕ (NOT).
 - И-НЕ (NAND).
 - ИЛИ-НЕ (NOR).

5. Контрольные вопросы:

1. Построение формулы по таблице истинности: СДНФ, СКНФ.
2. Упрощение СДНФ (СКНФ) аналитическим способом.
3. Упрощение СДНФ (СКНФ) с помощью построения Карт Карно.

4. Список литературы

Основная литература

1. Максимов Н. В. Архитектура ЭВМ и вычислительные системы: учебник / Н. В. Максимов, Т. Л. Партыка, И. И. Попов. - 4-е изд., перераб. и доп. - М.: ФОРУМ, 2012. - 512
2. Просветов Г.И. Дискретная математика: задачи и решения. – М.: Бином. Лаборатория знаний, 2010. – 222с.
3. Партыка Т. Л. Периферийные устройства вычислительной техники : учеб. пособие / Т. Л. Партыка, И. И. Попов. - 3-е изд., испр. и доп. - М. : ФОРУМ, 2012. - 432 с.

4. Юров В.И. Assembler. Учебник для вузов, 2-е издание. – СПб.: Питер, 2010. – 637с.

Дополнительная литература

1. Ижевский А.С., Калиниченко Б.Б. Методические указания для лабораторных работ «Моделирование и анализ электрических схем на основе логических элементов Electronic Workbench», 2009. - 38с Келим Ю.М. Вычислительная техника.- М.:Академия,2007. - 384с.
2. Вычислительная техника: метод. пособие /авт.-сост. Б.В. Цыбаков.- М., 2004. - 112с.

Раздел 3. Типовые узлы ЭВМ
Тема 3.1 Назначение шифраторов
Лабораторная работа №1. Исследование шифраторов и дешифраторов

Время, отводимое на выполнение работы - 2 часа.

Перечень необходимых технических средств обучения:

- ✓ персональные компьютеры Core i3 3.0, 4 Gb, 4 Гб ОЗУ, 250 Gb HDD;
- ✓ локальная сеть;
- ✓ коммутатор для подключения в сети Internet.

Перечень необходимых программных средств обучения:

- ✓ ОС Windows XP (7);
- ✓ Программа для моделирования Electronic Work Bench 5.12 и выше

1. Цель занятия: Углубление теоретических знаний по методу логического синтеза комбинационных цифровых устройств. Изучение назначения и принцип работы устройства логического анализатора. Знакомство с базовыми функциями логического анализатора.

2. Основные теоретические положения:

2.1 Логический анализатор

Анализатор предназначен для отображения на экране монитора 16-разрядных кодовых последовательностей одновременно в 16 точках схемы, а также в виде двоичных чисел на входных клеммах-индикаторах.

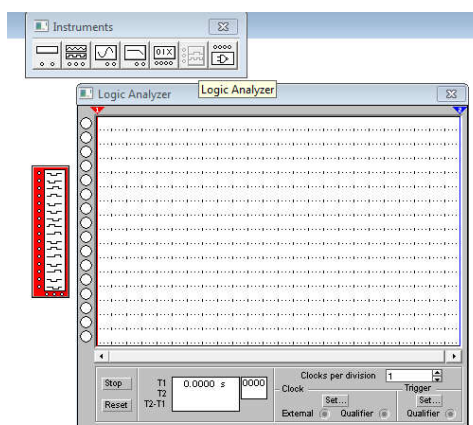


Рисунок 19. Logic Analyzer

Длительность развертки задается в окне TIME BASE. В блоке Clock имеются клеммы как для обычного (Extend), так и избирательного (Qualifier) источника запускающих сигналов, параметры которых могут установлены с помощью меню, вызываемого кнопкой Set. Запуск генератора можно осуществлять по переднему (Positive) или заднему (Negative) фронту запускающего сигнала с использованием внешнего (External) или внутреннего (Internal) источника. В окне Clock qualifier можно установить значение логического сигнала (0, 1 или X), при котором производится запуск анализатора.

Дополнительные условия запуска анализатора могут быть выбраны с помощью меню, которое вызывается кнопкой Set в блоке Trigger. С помощью этого окна в каналах A, B и C можно задать нужные двоичные 16-разрядные комбинации сигналов и затем в строке Trigger combinations установить дополнительные условия отбора:

- A OR B – запуск анализатора от канала A или B;
- A THEN B – запуск анализатора от канала A, если сигнал в канале B равен 1;
- (A OR B) THEN C – запуск анализатора от канала A или B, если сигнал в канале C равен 1.

В окне канала Trigger qualifier можно задать логические сигналы 1, 0 или X, при наличии которых производится запуск анализатора.

2.2 Дешифраторы и шифраторы

Дешифраторы и шифраторы по существу принадлежат к числу преобразователей кодов. С принятием шифрации связано представление о сжатии данных, с понятием дешифрации – обратное преобразование. Комбинационная схема, преобразующая поступающий на входы код в сигнал только на одном из ее выходов, называется **дешифратором**. В условных обозначениях дешифраторов и шифраторов используются буквы **ДС** и **СД** (от слов decoder и coder соответственно).

В общем случае дешифратор с “n” входами может иметь до 2^n выходов, которые обычно нумеруются десятичными цифрами, эквивалентными значениям двоичных кодов. Например, в трехвходовом дешифраторе при подаче на вход двоичного кода 011 единичный сигнал появится на выходе 3, а на остальных выходах сигнал будет равен нулю.

Дешифраторы (ДШ) — это комбинационные схемы с n входами и $m = 2^n$ выходами. Единичный сигнал, формирующийся на одном из m выходов, однозначно соответствует комбинации входных сигналов.

Дешифратор, устройство для расшифровки (декодирования) сообщения и перевода содержащейся в нём информации на язык (в код) воспринимающей системы. В общем случае Д. имеет n входов и m выходов. Поступающая на входы Д. информация преобразуется — дешифрируется, — и на соответствующем выходе (группе выходов) выделяется сигнал, указывающий признак (или содержание) входной информации.

Функционирование дешифратора описывается системой логических уравнений составленных на основе таблицы истинности.

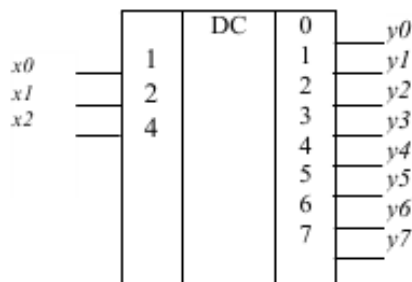


Рисунок 20 Условные обозначения дешифратора

Шифратор (ШР) решает задачу, обратную схемам ДШ, т. е. по номеру входного сигнала формирует однозначную комбинацию выходных сигналов. Номер входного сигнала определяется присутствием логической единицы на соответствующем входе (только одном). Соответствие комбинации выходных сигналов номеру входного можно задать соответствующей таблицей.

Шифратор — логическое устройство, выполняющее преобразование позиционного кода в n-разрядный двоичный код. Таким образом, шифратор - это комбинационное устройство, реализующее обратную дешифратору функцию. Шифраторы как отдельный класс функциональных устройств представлены в наиболее богатой ТТЛ-серии всего двумя ИМС — 74147 и 74148.

Для 4 входных сигналов она может выглядеть следующим образом - таблица 14.

Таблица 44 Таблица соответствия выходных сигналов номеру входного сигнала шифратора

Входы	Выходы
-------	--------

x ₁	x ₂	x ₃	x ₄	y ₁	y ₂
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	0	1
0	0	0	1	1	0

Состояния входов табл. 14 содержат только по одному единичному элементу. Другие произвольные комбинации входов недопустимы.

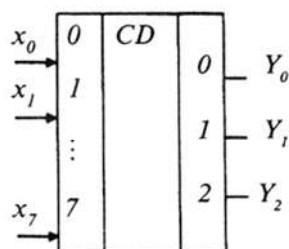


Рисунок 21. Обозначение шифратора на принципиальных электрических схемах

3. Задание к работе

3.1 **Исследование дешифратора** (при n информационных входах до 2^n выходов) с числом входов =3 по таблице истинности:

Таблица 15. Таблица истинности дешифратора

№	x ₂	x ₁	x ₀	y ₀	y ₁	y ₂	y ₃	y ₄	y ₅	y ₆	y ₇
0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
2	0	1	0	0	0	1	0	0	0	0	0
3	0	1	1	0	0	0	1	0	0	0	0
4	1	0	0	0	0	0	0	1	0	0	0
5	1	0	1	0	0	0	0	0	1	0	0
6	1	1	0	0	0	0	0	0	0	1	0
7	1	1	1	0	0	0	0	0	0	0	1

3.1.1 Заметьте, что указанный в первом столбце порядковый номер строки является фактически десятичным эквивалентом подаваемого на входы двоичного числа. Приведите пример.

3.1.2 Синтезируйте дешифратор как комбинационную схему, имеющую 8 (=2³) выходов и 3 входа.

3.1.3 Добавьте на проект **Word Generator** и **Logic Analyzer**. Введите в **Word Generator** слова для проверки правильности работы собранной схемы (от 0 до кол-ва выходов). Соединяем **Logic Analyzer** с выходами собранной схемой.

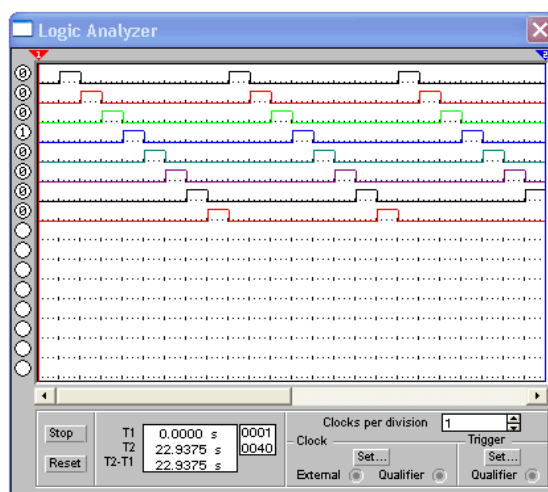


Рисунок 22. Диаграмма работы дешифратора

3.2 Анализ схемы включения дешифратора 74154 (отечественный аналог К155ИДЗ).

- 3.2.1 ИМС 74154 имеет четыре адресных входа А, В, С, D, два входа разрешения G1, G2 и шестнадцать выходов 0...15 (выходы не прямые, а инверсные, т.е. в исходном состоянии на выходах сигнал логической единицы).
- 3.2.2 В режиме дешифратора с генератора слова на входы G1, G2 подается 0, а на адресные входы — код в диапазоне 0000...1111.

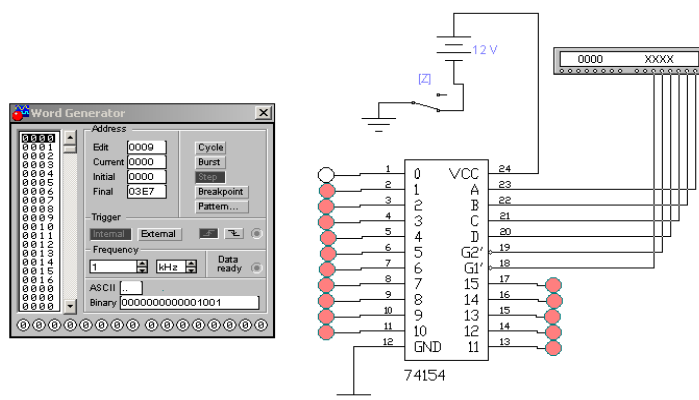


Рисунок 23. Анализ работы дешифратора 74154

3.3 Исследование работы шифратора приоритетного:

- 3.3.1 Создайте предложенную ниже схему шифратора:

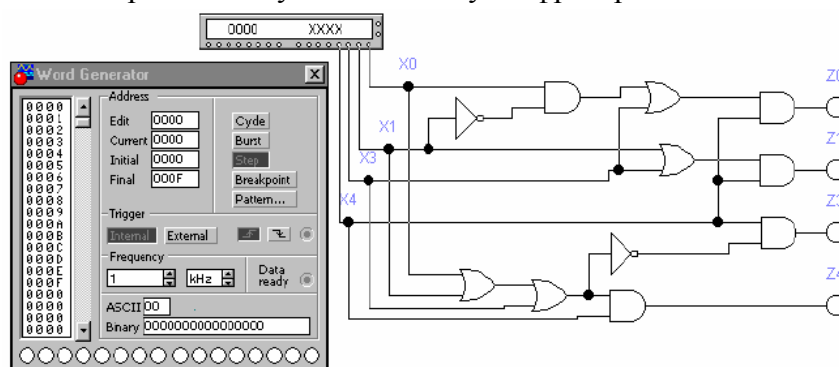


Рисунок 24. Схема дешифратора

3.3.2 Построить таблицу истинности для схемы:

Таблица 16. Таблица истинности схемы работы шифратора приоритетного

x_0	x_1	x_2	x_3	z_0	z_1	z_2	z_3
0	0	0	0				
0	0	0	1				
0	0	1	0				
0	0	1	1				
0	1	0	0				
0	1	0	1				
0	1	1	0				
0	1	1	1				
1	0	0	0				
1	0	0	1				
1	0	1	0				
1	0	1	1				
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				

3.3.3 Постройте СДНФ каждой функции. Упростите с помощью карт Карно.

3.4 Постройте приоритетный шифратор 74148:

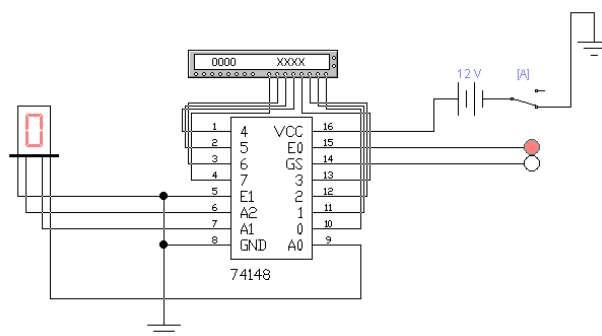


Рисунок 25. Приоритетный шифратор 74148

- 3.4.1 Назначение выводов ИМС 74148: 0...7 — входы; A0, A1, A2 — выходы; E1 — вход разрешения; EO, GS — выходы для каскадирования шифраторов.
- 3.4.2 При моделировании необходимо обратить внимание на реализацию принципа приоритета, при этом следует учесть, что все входы и выходы — инверсные (на функциональной схеме ИМС в программе EWB они показаны прямыми).

4. Содержание отчёта:

4.1 Ответьте на вопросы по заданию 3.3

- 4.1.1 Какие функции реализует данное устройство?
- 4.1.2 Назначение входов X1, X2, X3, X4?
- 4.1.3 Назначение выходов Z1, Z2, Z3, Z4?
- 4.1.4 Какие кодовые слова формируются на выходах Z0 и Z1, их физический смысл?
- 4.1.5 Какой сигнал формируется на выходе Z2, Z3, их физический смысл?

5. Контрольные вопросы:

- 5.1 Для чего предназначен логический анализатор.
- 5.2 Опишите принцип работы логического анализатора.

- 5.3 Что такое шифратор, при решении каких задач он используется?
- 5.4 Меня кодовые комбинации в генераторе слова схемы, покажите, в чем заключается смысл слова "приоритетный" в названии шифратора типа 74148.
- 5.5 При решении каких задач цифровой техники используется дешифратор?

6. Список литературы

Основная литература

1. Максимов Н. В. Архитектура ЭВМ и вычислительные системы: учебник / Н. В. Максимов, Т. Л. Партыка, И. И. Попов. - 4-е изд., перераб. и доп. - М.: ФОРУМ, 2012. - 512
2. Просветов Г.И. Дискретная математика: задачи и решения. – М.: Бином. Лаборатория знаний, 2010. – 222с.
3. Партыка Т. Л. Периферийные устройства вычислительной техники : учеб. пособие / Т. Л. Партыка, И. И. Попов. - 3-е изд., испр. и доп. - М. : ФОРУМ, 2012. - 432 с.
4. Юров В.И. Assembler. Учебник для вузов, 2-е издание. – СПб.: Питер, 2010. – 637с.

Дополнительная литература

1. Ижевский А.С., Калининченко Б.Б. Методические указания для лабораторных работ «Моделирование и анализ электрических схем на основе логических элементов Electronic Workbench», 2009. - 38с.
2. Келим Ю.М. Вычислительная техника.- М.:Академия,2007. - 384с.
3. Вычислительная техника: метод. пособие /авт.-сост. Б.В. Цыбаков.- М., 2004. - 112с.

Тема 3.2. Мультиплексоры и демультиплексоры

Лабораторная работа №2. Исследование и построение мультиплексоров. Реализация заданной функции с помощью мультиплексора

Время, отводимое на выполнение работы - 2 часа.

Перечень необходимых технических средств обучения:

- ✓ персональные компьютеры Core i3 3.0, 4 Gb, 4 ГБ ОЗУ, 250 Gb HDD;
- ✓ локальная сеть;
- ✓ коммутатор для подключения в сети Internet.

Перечень необходимых программных средств обучения:

- ✓ ОС Windows XP (7);
- ✓ Программа для моделирования Electronic Work Bench 5.12 и выше

Методические указания.

1. **Цель занятия:** Изучение назначения и принцип работы устройств мультиплексора и демультиплексора

2. **Основные теоретические положения:**

Мультиплексирование (в английском варианте multiplexing), это процесс объединения нескольких (множества – «multi») сигналов (каналов) в один групповой высокоскоростной сигнал.

Задачи мультиплексирования сигналов решаются на сетях связи с помощью оборудования, которое называется мультиплексорами.

Мультиплексор (от английского multiplex – многократный) это телекоммуникационное устройство, объединяющее несколько потоков данных или каналов в один выходной сигнал (групповой поток). Ранее, мультиплексорное оборудование, используемое на сетях связи, принято было называть другими терминами: «системы передачи» или «аппаратура уплотнения». **Мультиплексор** осуществляет выборку одного из нескольких входов и подключает его к своему выходу, в зависимости от состояния двоичного кода. Другими словами, мультиплексор - переключатель сигналов, управляемый двоичным кодом и имеющий несколько входов и один выход. К выходу подключается тот вход, чей номер соответствует двоичному коду. Или **мультиплексор** - это устройство, преобразующее параллельный код в последовательный.

У мультиплексора может быть, например, 16 входов и один выход. Это означает, что если к этим входам присоединить 16 источников цифровых сигналов – генераторов последовательных цифровых слов, то байты от любого из них можно передавать на единственный выход. Для выбора любого из 16 каналов необходимо иметь 4 входа селекции ($2^4=16$), на которые подается двоичный адрес канала. Так, для передачи данных от канала номер 9 на входах селекции необходимо установить код 1001. В силу этого мультиплексоры часто называют селекторами или селекторами-мультиплексорами.



Рисунок 26. Мультиплексор

Демультимплексоры в функциональном отношении противоположны мультиплексорам. С их помощью сигналы с одного информационного входа распределяются в требуемой последовательности по нескольким выходам. Выбор нужной выходной шины, как и в мультиплексоре, обеспечивается установкой соответствующего кода на адресных входах. При m адресных входах демультимплексор может иметь до 2^m выходов. В виде микросхемы, специально предназначенной для этого, демультимплексор не реализуется. Демультимплексор реализуется на дешифраторе, имеющем вход разрешения работы E . Пример реализации демультимплексора на основе дешифратора предлагается на рисунке.

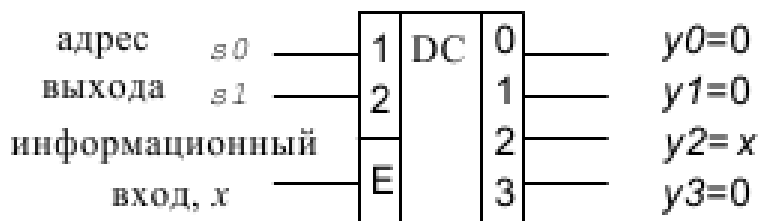


Рисунок 27. Пример реализации демультимплексора на основе дешифратора

3. Задание к работе

3.1 Исследование работы мультиплексора

3.1.1 Нарисуйте схему трёхканального мультиплексора. Проанализируйте работу с использованием Word Generator и Logic Analyzer.

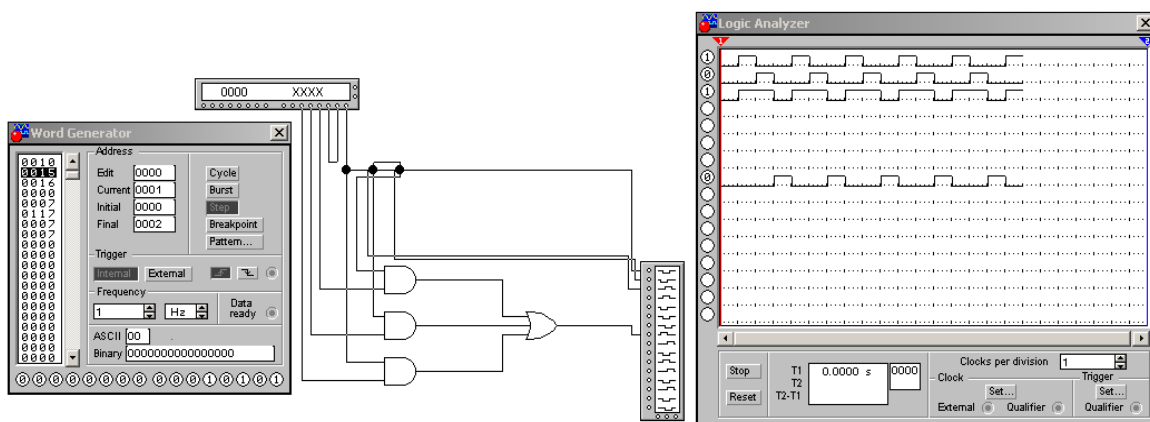


Рисунок 28. Схема трёхканального мультиплексора

3.1.2 Постройте схему двухканального мультиплексора, состоящего из элементов ИЛИ, НЕ и двух элементов И по формуле $BC' + AC$ (самостоятельно) с использованием генератора слов и логического анализатора.

3.2 Исследование мультиплексора 1-of-8 MUX с использованием генератора слов и логического анализатора.

3.2.1 Добавьте мультиплексор 1-of-8 MUX

3.2.2 Постройте схему:

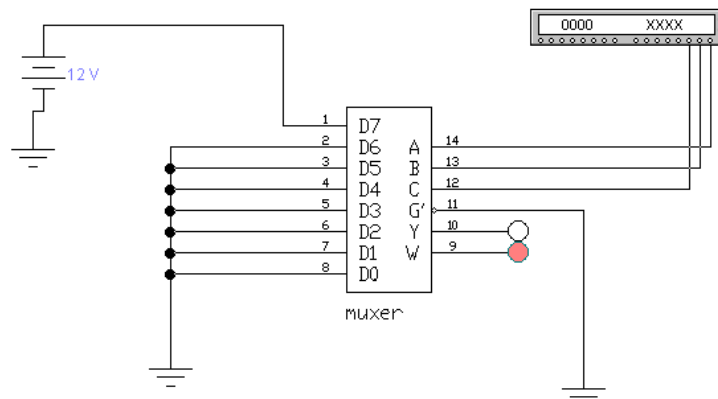


Рисунок 29. Исследование мультиплексора

3.2.3 Подавая в пошаговом режиме слова от генератора слов на входы мультиплексора и наблюдая уровни сигналов на выходах Y и W при помощи логических пробников, заполните таблицу истинности для трёх входов.

3.3 Реализация заданной функции с помощью мультиплексора. Если функция определена только для 8 комбинаций значений переменных, поэтому для её реализации можно использовать мультиплексор 8x1 с тремя управляющими входами.

3.3.1 Определите значение функции $F1 = \overline{C} \cdot \overline{B} \cdot \overline{A} \vee B \cdot A \vee C \cdot B$ для каждой комбинации значений аргументов и заполните графу F1расч.

Таблица 17. Таблица истинности: сравнение расчётного значения и значения, полученного в результате работы схемы

A	B	C	F1расч	F1рез
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

3.3.2 Из таблицы видно, что для реализации функции на мультиплексоре необходимо подать на информационный вход мультиплексора с номером N сигнал, значение которого равно соответствующему значению функции F1, т. е. на входы с номерами 1, 2, 4, 5 следует подать уровень логического нуля, а на остальные - уровень логической единицы. Постройте схему.

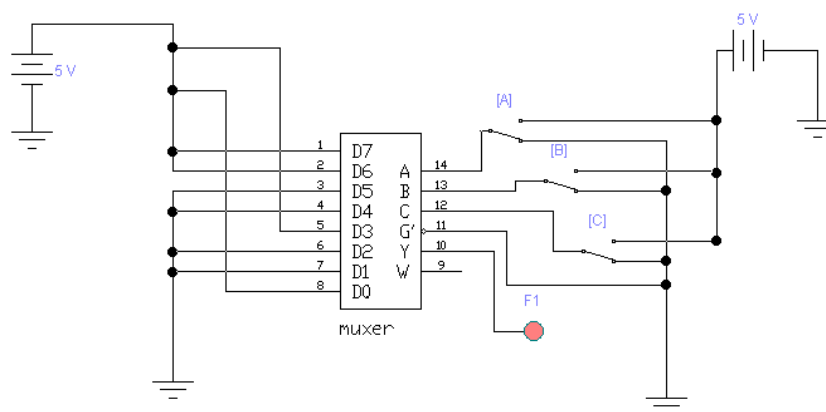


Рисунок 30. Реализация функции на мультиплексоре

3.3.3 Подайте при помощи ключей А, В, С все возможные комбинации логических сигналов на входы мультиплексора и, определяя уровень сигнала на выходе Y логическим пробником F1, заполните графу F1рез в таблице. Убедитесь, что функция, реализуемая мультиплексором, описывается заданным выражением.

3.4 Разработать, собрать и проверить работу схемы на основе мультиплексора 8х1, реализующую заданную логическую функцию Y. Варианты задач (по списку в журнале – последняя цифра):

- 1) $Y = \bar{C} \cdot \bar{B} \vee C \cdot A;$
- 2) $Y = \bar{B} \cdot \bar{A} \vee C \cdot \bar{B} \vee \bar{C} \cdot \bar{A};$
- 3) $Y = \bar{C} \cdot \bar{B} \cdot \bar{A} \vee C \cdot \bar{B} \cdot A \vee \bar{C} \cdot B \cdot A \vee C \cdot B \cdot \bar{A};$
- 4) $Y = C \cdot A \vee \bar{B} \cdot \bar{A} \vee \bar{C} \cdot \bar{A};$
- 5) $Y = D \cdot \bar{B} \cdot \bar{A} \vee C \cdot B \vee D \cdot \bar{B} \cdot A;$
- 6) $Y = \bar{D} \cdot \bar{B} \cdot \bar{C} \cdot \bar{A} \vee C \cdot B \cdot A \vee D \cdot \bar{C} \cdot B;$
- 7) $Y = D \cdot C \vee B \cdot A;$
- 8) $Y = \bar{D} \cdot C \cdot A \vee D \cdot \bar{B} \cdot \bar{A} \vee C \cdot B \cdot \bar{A} \vee D \cdot \bar{C} \cdot B;$
- 9) $Y = D \cdot (\bar{E} \cdot B \cdot A \vee C \cdot \bar{B} \cdot A \vee E \cdot \bar{C} \cdot B);$
- 10) $Y = \bar{B} \cdot (E \cdot D \cdot C \cdot A \vee \bar{E} \cdot \bar{D} \cdot C \vee D \cdot \bar{C} \cdot A);$

3.4.1 Самостоятельно исследуйте работу сдвоенного четырехканального мультиплексора (микросхема 74153). Составьте таблицу функционирования схемы для выходов Y1 и Y2. Для этого установите ключами 1 и 2 уровень логического нуля на входах G1 и G2. Затем в пошаговом режиме последовательно подайте от генератора все слова последовательности для каждой комбинации логических уровней на входах А и В. Для каждого шага определите входы, сигналы с которых проходят на выходы микросхемы.

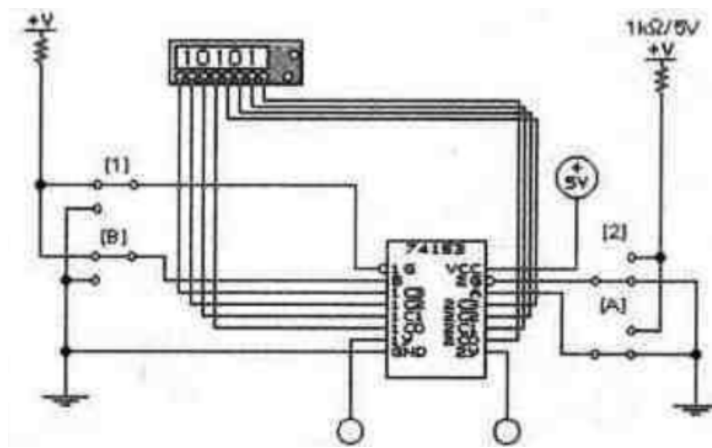


Рисунок 31. Схема четырёхканального мультиплексора

3.5 Исследование демультиплексора

3.5.1 Постройте схему двухканального демультиплексора:

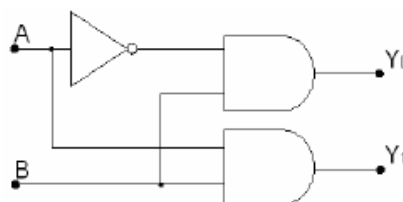


Рисунок 32. Схема двухканального мультиплексора

- Схема содержит два элемента И и один элемент НЕ.
- На схеме: X - информационный вход, A - вход адреса, Y0, Y1 - выходы.
- Если A=0 сигнал информационного входа передается на выход Y0, а при A=1 - на выход Y1.

3.5.2 Проанализируйте работу демультиплексора с помощью генератора слов и логического анализатора.

4. Содержание отчёта:

- 4.1 Схема трёхканального мультиплексора в логическом базисе элементов ИЛИ, НЕ.
- 4.2 Схема мультиплексора 1-of-8 MUX.
- 4.3 Схема реализации заданной функции на базе мультиплексора.
- 4.4 Схема сдвоенного четырёхканального мультиплексора.
- 4.5 Схема двухканального демультиплексора.

5. Контрольные вопросы:

- 5.1 Что такое мультиплексор, каково его назначение?
- 5.2 Что такое демультиплексор, для решения каких задач его можно применить?
- 5.3 Какими логическими уравнениями описывается работа микросхемы сдвоенного мультиплексора 74153?

6. Список литературы

Основная литература

1. Максимов Н. В. Архитектура ЭВМ и вычислительные системы: учебник / Н. В. Максимов, Т. Л. Партыка, И. И. Попов. - 4-е изд., перераб. и доп. - М.: ФОРУМ, 2012. - 512
2. Просветов Г.И. Дискретная математика: задачи и решения. – М.: Бином. Лаборатория знаний, 2010. – 222с.
3. Партыка Т. Л. Периферийные устройства вычислительной техники : учеб. пособие / Т. Л. Партыка, И. И. Попов. - 3-е изд., испр. и доп. - М. : ФОРУМ, 2012. - 432 с.
4. Юров В.И. Assembler. Учебник для вузов, 2-е издание. – СПб.: Питер, 2010. – 637с.

Дополнительная литература

4. Ижевский А.С., Калиниченко Б.Б. Методические указания для лабораторных работ «Моделирование и анализ электрических схем на основе логических элементов Electronic Workbench», 2009. - 38с.
5. Келим Ю.М. Вычислительная техника.- М.:Академия,2007. - 384с.
6. Вычислительная техника: метод. пособие /авт.-сост. Б.В. Цыбаков.- М., 2004. - 112с.

Лабораторная работа №3. Исследование триггеров

Время, отводимое на выполнение работы - 2 часа.

Перечень необходимых технических средств обучения:

- ✓ персональные компьютеры Core i3 3.0, 4 Gb, 4 ГБ ОЗУ, 250 Gb HDD;
- ✓ локальная сеть;
- ✓ коммутатор для подключения в сети Internet.

Перечень необходимых программных средств обучения:

- ✓ ОС Windows XP (7);
- ✓ Программа для моделирования Electronic Work Bench 5.12 и выше

Методические указания.

1. **Цель занятия:** Изучение назначения и принцип работы устройств триггера. Знакомство с базовыми устройствами триггер из библиотеки EWB
2. **Основные теоретические положения:**

Более сложным преобразователем информации являются **схемы с памятью**. Наличие памяти в схеме позволяет запоминать промежуточные состояния обработки и учитывать их значения в дальнейших преобразованиях. Выходные сигналы $Y = (y_1, y_2, \dots, y_m)$ в схемах данного типа формируются не только по совокупности входных сигналов $X = (x_1, x_2, \dots, x_n)$, но и по совокупности состояний схем памяти $Q = (q_1, q_2, \dots, q_n)$. При этом различают текущий дискретный момент времени t и последующий $(t + 1)$ момент времени (рис. 33).

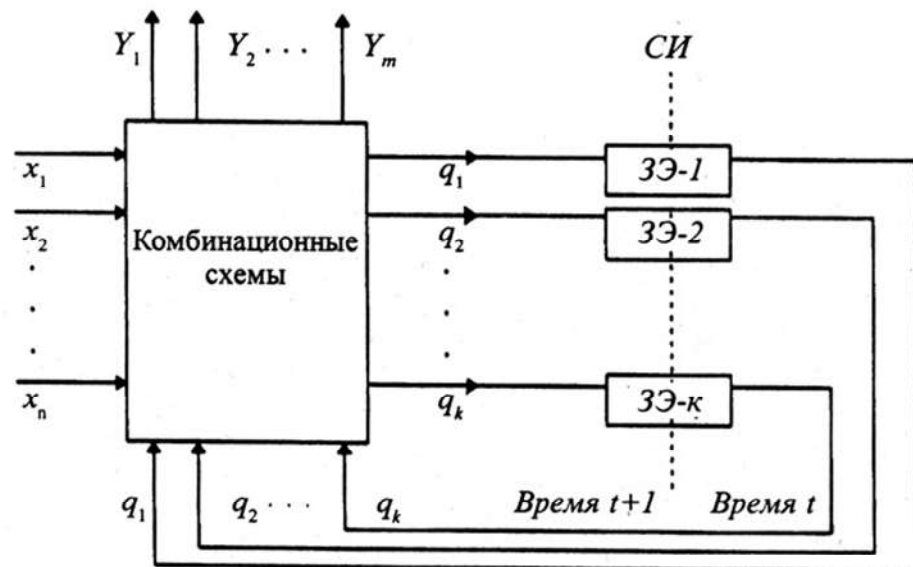


Рисунок 33.4 Обобщенная структура схемы с памятью

Передача значения Q между моментами времени t и $(t+1)$ осуществляется обычно с применением двухступенчатой памяти и синхронизирующих импульсов (СИ).

В качестве простейшего запоминающего элемента в современных ЭВМ используют триггеры. Самая простейшая схема триггера (RS-триггер) может быть синтезирована по общим правилам.

В данном случае для этого необходимо построить автомат памяти - триггер, имеющий вход **R** (Reset - сброс), для установки элемента в «нулевое состояние» и вход **S** (Set - установка) - для установки элемента в «единичное» состояние. При отсутствии сигналов

$R=S=0$ элемент должен сохранять свое состояние до тех пор, пока не будут получены новые сигналы на входе R или S .

Условия работы триггера могут быть показаны в виде таблицы переходов, представляющей собой модификацию таблицы истинности. Наиболее простой вид она имеет для автомата, описываемого системой уравнений автомата Мура.

Триггеры – устройства, имеющие два устойчивых состояния. Под действием управляющих сигналов они переходят из одного состояния в другое и после снятия сигналов хранят это состояние до тех пор, пока не отключено напряжение питания. Таким образом, триггер является ячейкой памяти для одного двоичного разряда, т. е. бита информации. Справа приведена схема асинхронного одноканального RS – триггера на логических элементах И-НЕ.

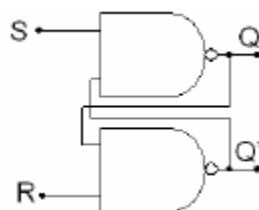


Рисунок 34. Схема асинхронного одноканального RS – триггера

По характеру реакции на входные сигналы триггеры делятся на два типа: асинхронные и синхронные. Асинхронный триггер характеризуется тем, что входные сигналы действуют на состояние триггера непосредственно с момента их подачи на входы, в синхронных триггерах - только при подаче синхронизирующего сигнала на управляющий вход C .

Таблица 18 Условия работы триггера

Входы			Выход	
R	S	q_t	Состояние q_{t+1}	Режим
0	0	0	0	Хранение
0	0	1	1	Хранение
1	0	0	0	Установка 0
1	0	1	0	Установка 0
0	1	0	1	Установка 1
0	1	1	1	Установка 1
1	1	0	?	Запрещенное состояние
1	1	1	?	Запрещенное состояние

Содержание таблицы расшифровывается следующим образом. Элемент памяти может сохранять значение $q_t=0$ или $q_t=1$ в зависимости от установки ранее установленного состояния. При отсутствии входных сигналов на входах R и S ($R=0$ и $S=0$) значения q_{t+1} в первой строке таблицы в точности повторяют значения q_t . При поступлении сигнала $R=1$ (сигнала установки «нуля») элемент независимо от своего состояния принимает значение, равное нулю, $q_{t+1} = 0$. Если же на вход S поступает сигнал установки «единицы» ($S=1$), то $q_{t+1}=1$ независимо от предыдущего состояния q_t . Одновременное поступление сигналов на входы R и S является запрещенной ситуацией, так как она может привести к непредсказуемому состоянию. В схемах формирования сигналов R и S должны быть предусмотрены блокировки, исключающие их совпадения, $S=R=1$.

RS-триггер нашел широкое распространение в схемах ЭВМ. Одиночные триггеры этого типа часто используются в различных блоках управления. В асинхронных RS-триггерах имеется один существенный недостаток, обусловленный самой логикой их построения, т.е. в них сигналы R и S должны быть разнесены во времени. Дополнение этого

триггера комбинационными схемами синхронизации на входе и выходе позволяет получить триггеры с более сложной логикой работы: синхронные RS-, T-, JK-, D-триггеры и целый ряд комбинированных RST-, JKRS-, DRS-триггеров

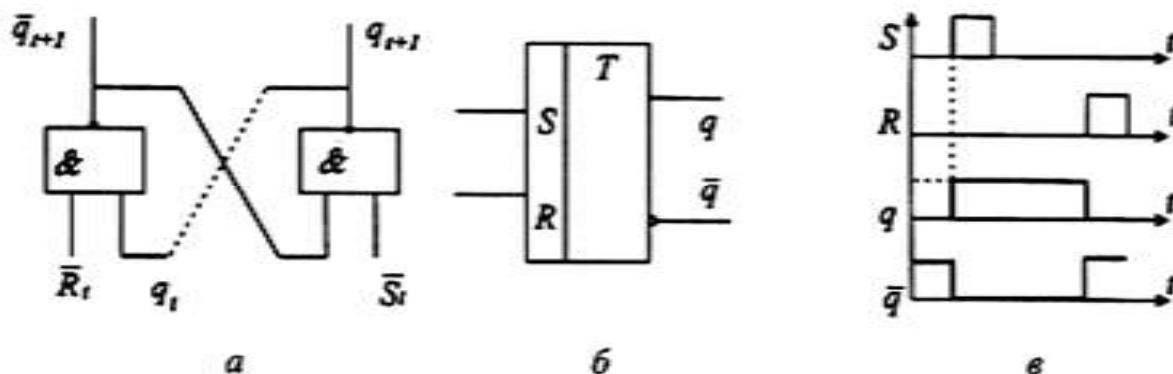


Рисунок 35. Схема асинхронного RS-триггера: а — схема; б — обозначение на принципиальных электрических схемах; в — временная диаграмма

Прописные буквы в названиях триггеров обозначают:

- R (Reset — сброс) — вход установки триггера в нулевое состояние, $Q = 0$
- S (Set — установка) — вход установки триггера в единичное состояние, $Q = 1$
- T (Toggle — релаксатор) — счетный вход триггера
- J (Jerk — внезапное включение) — вход установки JK-триггера в единичное состояние, $Q = 1$
- K (Kill — внезапное выключение) — $Q = 0$
- D (Delay — задержка) — вход установки триггера в единичное или нулевое состояние на время, равное одному такту
- C (Clock — часы) — вход синхронизирующих тактовых импульсов

T-триггер или, иначе говоря, триггер со счетным входом при значении $X_t = 0$ триггер сохраняет свое ранее установленное состояние - режим хранения состояния, при $X_t = 1$ триггер переходит в противоположное состояние. Таблица переходов (табл. 19) и диаграмма работы (рис. 36 в) отражают динамику работы этого элемента.

Таблица 19 Таблица переходов T-триггера

Входные сигналы	Состояние q_t		Режим
X_t	0	1	
0	0	1	Хранение Инверсия
1	1	0	

Используя таблицу переходов, можно получить логическую функцию, реализуемую T-триггером:

$$q_{t+1} = \overline{q_t} \cdot x_t \vee q_t \cdot \overline{x_t} = q_t \oplus x_t$$

Нетрудно видеть, как реализовать полученную зависимость — для этого необходимо в RS-триггере выход q_t соединить с R входом RS-триггера, а выход $\overline{q_t}$ соединить с S входом. На рис. 6.1, а показано, как двухтактный RS-триггер преобразуется в T-триггер

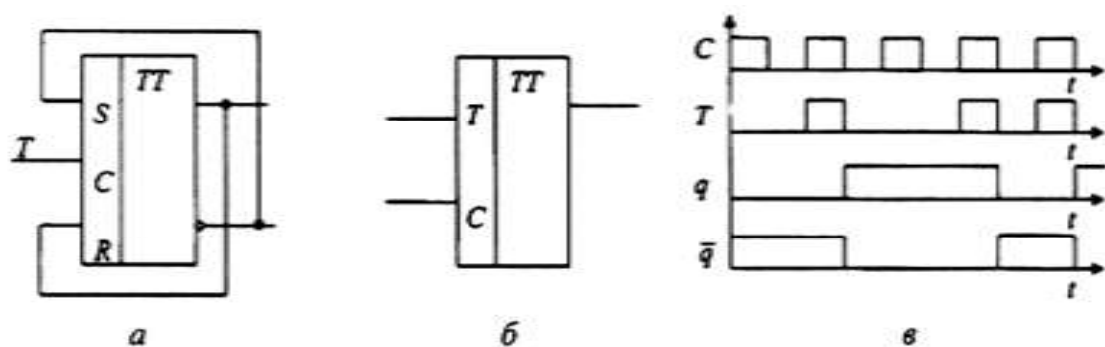


Рисунок 36 Схема триггера со счетным входом: а — функциональная схема; б — условное обозначение; в — временная диаграмма

D-триггер обычно строится на основе двухтактного RS- или JK-триггера. Он предназначен для хранения состояния (1 или 0) на один период тактовых импульсов (с задержкой на 1 такт). Его переходы отражены в табл.20

Таблица 20 Таблица переходов D-триггера

Входные сигналы	Состояние q_t		Режим
	0	1	
D	0	1	
0	0	0	Установка 0
1	1	1	Установка 1

Используя таблицу переходов, можно получить логическую функцию, реализуемую D-триггером:

$$q_{t+1} = q_t x_t \vee \bar{q}_t \bar{x}_t$$

На рис. 37, а и б представлены варианты построения D-триггера, а на рис. 43, в — его условное обозначение.

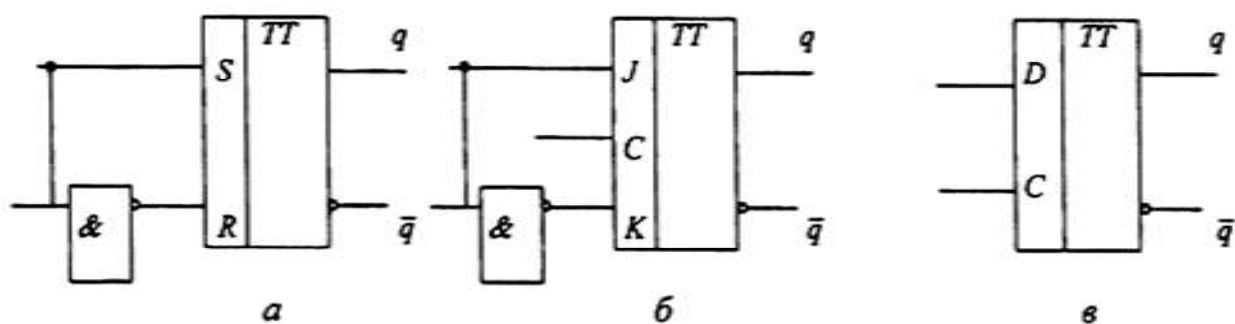


Рисунок 37 D-триггер: а — функциональная схема на основе RS-триггера; б — функциональная схема на основе JK-триггера; в — условное обозначение

Все перечисленные элементы памяти позволяют хранить одну единицу информации бит.

Триггеры используются, в основном, в вычислительной технике для организации компонентов вычислительных систем: регистров, счётчиков, процессоров, ОЗУ.

В библиотеке EWB триггеры представлены тремя типами: RS, JK и D:

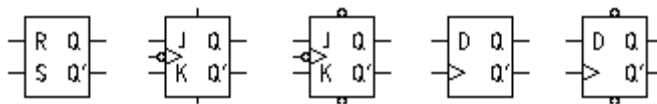


Рисунок 38 Типы триггеров: а) RS триггер, б) JK триггер, в) D триггер

Назначение выводов триггеров следующее:

Для всех триггеров выходы Q – **прямой**, Q' – **инверсный** (обратный).

Для **RS – триггера** R – установка триггера в 0, при сигнале 1 на этом входе $Q=0$, $Q'=1$; S – установка в 1, при сигнале 1 на этом входе $Q=1$, $Q'=0$; комбинация $R=1$, $S=1$ не изменяет состояние выходов и относится к запрещенным.

Для **JK триггера** J , K – информационные входы, $>$ - тактовый вход; вывод сверху – асинхронная предустановка триггера в единичное состояние ($Q=1$) вне зависимости от состояния сигналов на входах (функционально аналогичен входу S RS триггера); вывод внизу – асинхронная предустановка триггера в нулевое состояние (так называемая очистка триггера, после которой $Q'=1$); наличие кружочков на изображениях выводов обозначает, что активными являются сигналы низкого уровня, а для тактового входа – что переключение триггера производится не по переднему фронту тактового импульса, а по его срезу.

Для **D – триггера** вход D – информационный, состояние этого входа после подачи тактового импульса запоминается триггером, т. е. при $D=1$ имеем $Q=1$, при $D=0$ – $Q=0$.

3. Задание к работе

3.1 Постройте схему функционирования RS-триггера:

- 3.1.1 Какую роль играет сигнал синхронизации C ? При каком значении сигнала C происходит установка триггера в 0 и 1?
- 3.1.2 В тетради составьте таблицу истинности RS-триггера:

Таблица 21 Таблица истинности RS-триггера

Вход 1 (S)	Вход 2 (R)	Выход 1 (Q)	Действие триггера
1	0		
0	1		
0	0		
1	1	неустойчивое состояние	запрещено

- 3.1.3 Если на входы подать «0», то как изменится значение на выходах?
- 3.1.4 Каковы должны быть возможные комбинации входных сигналов для имитации работы триггера в различных режимах.
- 3.1.5 Назовите вход установки в 0, в 1?

3.2 Постройте схему функционирования RS-триггера с применением нужного типа триггера из библиотеки EWB. Проверьте работу триггера по построенной таблице.

3.3 Дополните схему триггера инвертором:

Дополнительная литература

1. Ижевский А.С., Калиниченко Б.Б. Методические указания для лабораторных работ «Моделирование и анализ электрических схем на основе логических элементов Electronic Workbench», 2009. - 38с.
2. Келим Ю.М. Вычислительная техника.- М.:Академия,2007. - 384с.
3. Вычислительная техника: метод. пособие /авт.-сост. Б.В. Цыбаков.- М., 2004. - 112с.

Тема 3.6. Сумматоры

Лабораторная работа №4. Исследование сумматоров

Время, отводимое на выполнение работы - 2 часа.

Перечень необходимых технических средств обучения:

- ✓ персональные компьютеры Core i3 3.0, 4 Gb, 4 ГБ ОЗУ, 250 Gb HDD;
- ✓ локальная сеть;
- ✓ коммутатор для подключения в сети Internet.

Перечень необходимых программных средств обучения:

- ✓ ОС Windows XP (7);
- ✓ Программа для моделирования Electronic Work Bench 5.12 и выше

1. **Цель занятия:** Изучение назначения и принцип работы устройств полусумматора и сумматора. Знакомство с базовыми элементами полусумматора и полного сумматора из библиотеки EWB.

2. **Основные теоретические положения:**

Арифметические сумматоры являются составной частью арифметико-логических устройств (АЛУ) микропроцессоров.

В цифровой вычислительной технике используются одноразрядные суммирующие схемы с двумя и тремя входами, причём первые называются полусумматорами, а вторые — полными одноразрядными сумматорами. Полусумматоры могут использоваться только для суммирования младших разрядов чисел. Полные одноразрядные сумматоры имеют дополнительный третий вход, на который подаётся перенос из предыдущего разряда при суммировании многоразрядных чисел.

Комбинационный сумматор. Принципы построения и работы сумматора вытекают из правил сложения двоичных цифр. Схема сумматора также является регулярной и широко используется в ЭВМ. При сложении одноразрядных двоичных цифр можно выявить закономерности в построении и многоразрядных сумматоров.

Сначала рассмотрим сумматор, обеспечивающий сложение двух двоичных цифр a_i и b_i , считая, что переносы из предыдущего разряда не поступают. Этой логике отвечает сложение младших разрядов двоичных чисел. Процесс сложения описывается таблицей истинности (табл. 11) и логическими зависимостями (*), где S_i — функция одноразрядной суммы и P_i — функция формирования переноса. Перенос формируется в том случае, когда $a_i=1$ и $b_i=1$.

$$S_i = a_i \cdot b_i \vee a_i \cdot \bar{b}_i = a_i \oplus b_i \quad (*)$$

$$P_i = a_i \cdot b_i$$

Таблица 13 Таблица истинности комбинационного полусумматора

Входы		Выходы	
a_i	b_i	S_i	P_i
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Зависимости (*) соответствуют логике работы самого младшего разряда любого сумматора. Структурная схема одноразрядного сумматора (полусумматора) представлена на рис.13

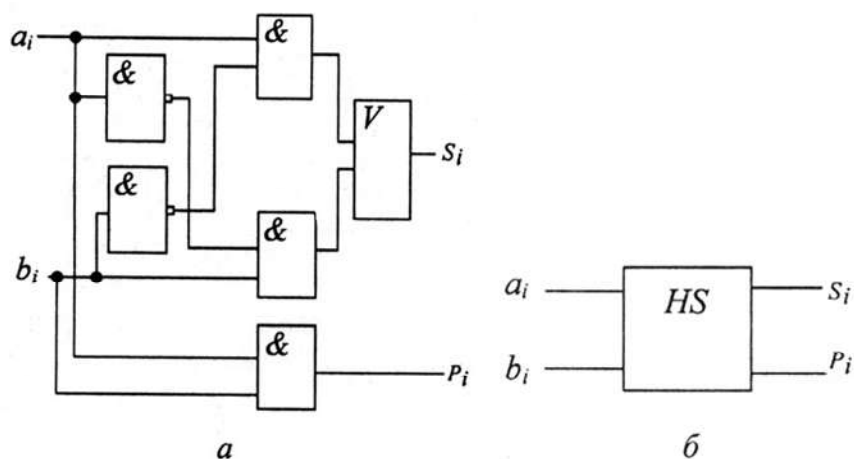


Рисунок 13. Структурная схема полусумматора (а) и обозначение полусумматора (б)

Логические зависимости полусумматора S_i и схемы сложения по модулю 2 совпадают и инверсны по отношению к компаратору.

Таблица истинности сумматора, учитывающего сигналы переноса, отличается от таблицы полусумматора дополнительным входом p - переносом из предыдущих разрядов.

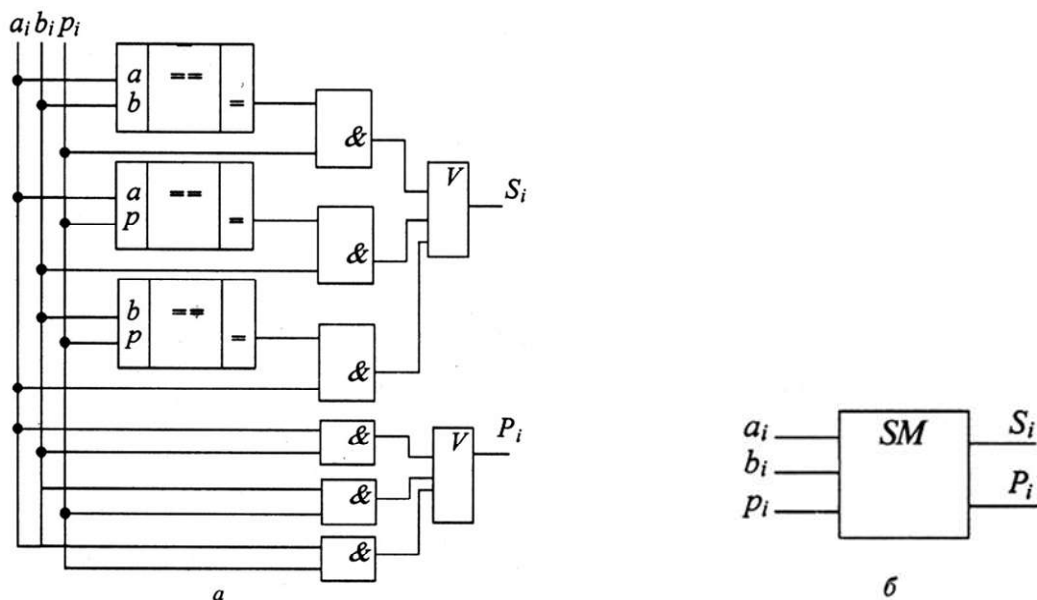


Рисунок 14. Структурная схема одного разряда комбинационного сумматора: а – структурная схема одного разряда; б – условное изображение

Структурная схема многоразрядного комбинационного сумматора на электрических схемах изображена на рис 15.

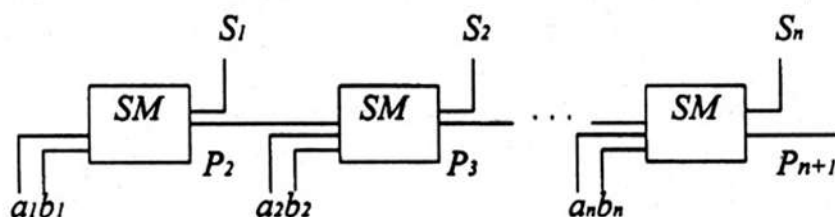


Рисунок 15. Структурная схема многоразрядного комбинационного сумматора

3. Задание к работе

3.1 Разработка схемы полусумматора:

3.1.1 Постройте двоичную таблицу сложения двух чисел 1 и 0:

A	B	A+B
0	0	
0	1	
1	0	
1	1	

3.1.2 При сложении двоичных чисел в каждом разряде образуется сумма и при этом возможен перенос в старший разряд. Введем обозначения слагаемых (A, B), первый разряд суммы (P), и второй разряд суммы (S). Постройте таблицу сложения одноразрядных двоичных чисел с учётом переноса в следующий разряд:

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

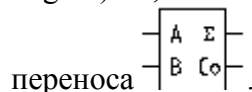
3.1.3 Найдите СДНФ и СКНФ S и P. Постройте логическую схему с входами A, B и выходами P, S для полученных формул на одной схеме.

3.1.4 Найдите СКНФ для S, СДНФ для P. Постройте логическую схему с входами A, B и выходами P, S для полученных формул на одной схеме.

3.1.5 Преобразуйте формулу СКНФ для S, используя закон Де Моргана. Постройте логическую схему для полученной формулы.

3.2 Исследование полусумматора.

3.2.1 Для исследования полусумматора добавьте на схему элемент (библиотека Digital). A, B – входы слагаемых, Σ - результат суммирования, C0 – выход



3.2.2 Добавьте на схему Logic Converter. Соедините два входа и выход Σ .

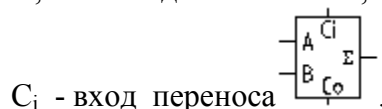
3.2.3 Постройте таблицу истинности для схемы. По таблице истинности получите формулу.

3.3 Построение схемы полного одноразрядного сумматора. Исследование полного сумматора

3.3.1 Постройте таблицу истинности для полного одноразрядного сумматора для трёх входов и двух выходов.

3.3.2 Постройте формулы СДНФ для P и S.

3.3.3 Для исследования сумматора добавьте на схему элемент (библиотека Digital). A, B – входы слагаемых, Σ - результат суммирования, C0 – выход переноса,



3.3.4 По аналогии со 2-м заданием постройте схему функционирования полного одноразрядного сумматора.

3.4 Исследуйте приведенный трехразрядный сумматор, последовательно подключая выходы к логическому конвертору:

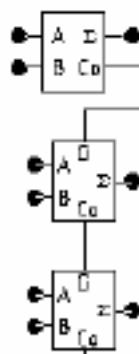


Рисунок 5 Трёхразрядный сумматор

4. Содержание отчёта:

- 4.1 Схемы полусумматора, полного одноразрядного сумматора.
- 4.2 Схема полного одноразрядного сумматора.

5. Контрольные вопросы:

- 5.1 Чем отличается полусумматор от полного сумматора?
- 5.2 Исследуйте выходы переноса полусумматоров и полного сумматора.

6. Список литературы

Основная литература

1. Максимов Н. В. Архитектура ЭВМ и вычислительные системы: учебник / Н. В. Максимов, Т. Л. Партыка, И. И. Попов. - 4-е изд., перераб. и доп. - М.: ФОРУМ, 2012. - 512
2. Просветов Г.И. Дискретная математика: задачи и решения. – М.: Бином. Лаборатория знаний, 2010. – 222с.
3. Партыка Т. Л. Периферийные устройства вычислительной техники : учеб. пособие / Т. Л. Партыка, И. И. Попов. - 3-е изд., испр. и доп. - М. : ФОРУМ, 2012. - 432 с.
4. Юров В.И. Assembler. Учебник для вузов, 2-е издание. – СПб.: Питер, 2010. – 637с.

Дополнительная литература

1. Ижевский А.С., Калининченко Б.Б. Методические указания для лабораторных работ «Моделирование и анализ электрических схем на основе логических элементов Electronic Workbench», 2009. - 38с.
2. Келим Ю.М. Вычислительная техника.- М.:Академия,2007. - 384с.
3. Вычислительная техника: метод. пособие /авт.-сост. Б.В. Цыбаков.- М., 2004. - 112с.

Лабораторная работа №5. Построение четырёхразрядного сумматора. Вычитание

Время, отводимое на выполнение работы - 2 часа.

Перечень необходимых технических средств обучения:

- ✓ персональные компьютеры Core i3 3.0, 4 Gb, 4 Гб ОЗУ, 250 Gb HDD;
- ✓ локальная сеть;
- ✓ коммутатор для подключения в сети Internet.

Перечень необходимых программных средств обучения:

- ✓ ОС Windows XP (7);
- ✓ Программа для моделирования Electronic Work Bench 5.12 и выше

1. **Цель занятия:** Изучение назначения и принцип работы виртуального генератора слова. Знакомство с базовыми функциями виртуального генератора слов. Анализ работы сумматора с помощью виртуального генератора слов.

2. Основные теоретические положения:

2.1 Реализация вычитания с помощью сумматора

Вычитание числа В из А выполняется путем суммирования отрицательного числа В в дополнительном коде с числом А. Представление отрицательного двоичного числа можно получить путем инвертирования всех битов числа и добавлением 1. Прибавление этой единицы эффективно реализуется в полном сумматоре путем замены его первого каскада (полусумматора) на полный сумматор, вход переноса которого подключается к напряжению с уровнем логической 1.

2.2 Генератор двоичных слов

Генератор (его еще называют кодовым генератором) предназначен для генерации 16 разрядных двоичных слов. Кодовые комбинации необходимо задавать в шестнадцатеричном коде.

Каждая комбинация заносится с помощью клавиатуры, номер редактируемой ячейки фиксируется в окошке **EDIT** блока **ADRESS**. Всего таких ячеек и следовательно, комбинаций – 2048. В процессе работы генератора в отсеке **ADRESS** индицируется номер текущей ячейки (**CURRENT**), ячейки инициализации или начала работы (**INITIAL**) и конечной ячейки (**FINAL**). Выдаваемые на 16 выходов (В нижней части генератора) кодовые комбинации индицируются в текстовом (**ASCII**) и двоичном коде (**BINARY**).

Сформированные слова выдаются на 16 расположенных в нижней части виртуального прибора клемм-индикаторов:

- В пошаговом (при нажатии кнопки **STEP**), циклическом (при нажатии кнопки **CYCLE**) или с выбранного слова до конца (при нажатии клавиши **BURST**) при заданной частоте посылок (установка – нажатиями кнопок в окнах **FREQUENCY**);

- При внутреннем (при нажатии кнопки **INTERNAL**) или внешнем запуске (при нажатии кнопки **EXTERNAL** по готовности данных (клемма **DATA READY**), рядом расположена клемма для подключения канала синхронизации);

- При запуске по переднему или заднему фронту.

Кнопка **BREAK POINT** – прерывание работы генератора в указанной ячейке.

При нажатии на кнопку **PATTERN** выпадает меню:

- **Clear buffer** – стереть содержимое буфера (содержимое буфера экрана);

- **Open** – загрузить кодовые комбинации (из файла с расширением .dp);

- **Save** – записать все набранные на экране комбинации в файл (.dp);

- **Up counter** - заполнить буфер экрана кодовыми комбинациями, начиная с 0 в нулевой ячейке и далее с прибавлением 1 в каждой последующей ячейке;

- **Down counter** – заполнить буфер кодовыми комбинациями, начиная с FFFF в нулевой ячейке и далее с уменьшением на 1 в каждой последующей ячейке;

- **Shift right** – заполнить каждые четыре ячейки комбинациями 1-2-4-8 со смещением их в следующих четырех ячейках вправо;

- **Shift left** – тоже самое, но со смещением влево.

3. Задание к работе

3.1 Исследование полусумматора.

3.1.1 Постройте схему работы полусумматора (рис.16).

3.1.2 Проанализируйте работу полусумматора, используя генератор слов.

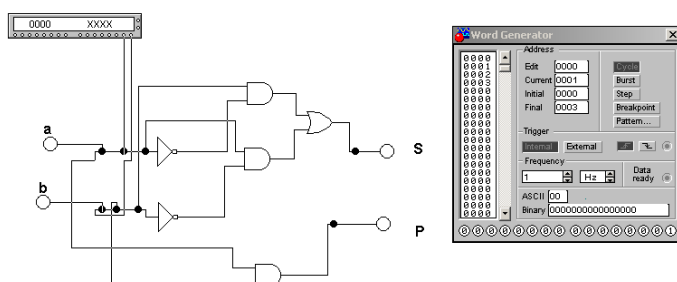


Рисунок 166 Схема работы полусумматора

3.2 Построение схемы полного одноразрядного сумматора. Исследование полного сумматора

3.2.1 Постройте схему функционирования полного одноразрядного сумматора.

3.2.2 Проанализируйте работу одноразрядного сумматора с использованием генератора слов.

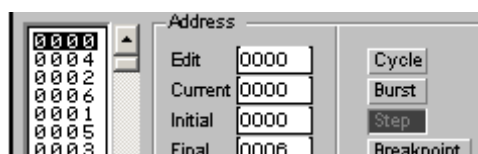


Рисунок 17 Генератор слов

3.3 Требуется исследовать четырехразрядный сумматор с помощью виртуального генератора слов

3.3.1 Откройте панель виртуального генератора слова.

3.3.2 Собираем четырехразрядный сумматор, состоящий из одного полусумматора и трех полных сумматоров.

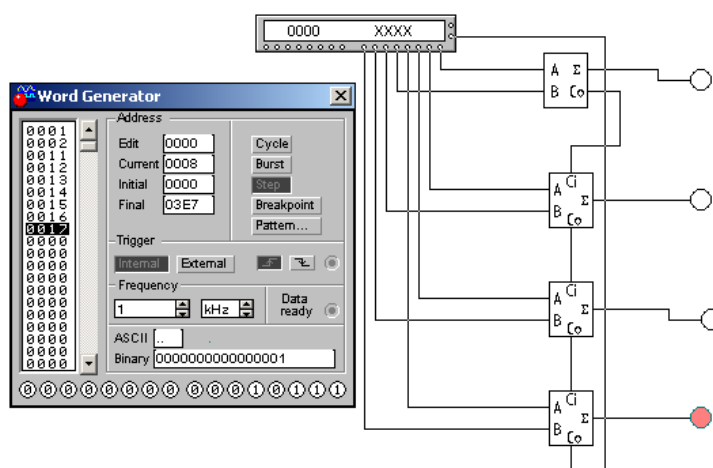


Рисунок 18. Четырёхразрядный сумматор

3.3.3 Из панели Indicators выбираем 4 индикатора и подключаем к выходам сумматора (при сигнале 1 – зажигаются).

3.3.4 Выходы виртуального генератора слов подключаем к входам сумматора; четыре первых разряда соответствуют первому слагаемому, четыре последующих разряда – второму слагаемому.

- 3.3.5 В лицевой панели генератора вводим шестнадцатиразрядные числа 0, 1, 11, 12, 13, 14, 15, 16, 17.
- 3.3.6 Запустить генератор слов в режиме Step. Проанализировать работу сумматора.
- 3.3.7 **Постройте пятиразрядный сумматор и проанализируйте с помощью генератора слов.** Какие шестнадцатиразрядные числа необходимо ввести для анализа работы сумматора?

3.4 Самостоятельно постройте четырёхразрядный вычитатель

4. Содержание отчёта:

- 4.1 Схемы полусумматора, полного одноразрядного сумматора.
- 4.2 Схема четырёхразрядного сумматора, построенная с помощью виртуального генератора слов
- 4.3 Схема четырёхразрядного вычитателя.

5. Контрольные вопросы:

- 5.1 Объяснить предназначение виртуального генератора слов.
- 5.2 Перечислить основные функции генератора слов
- 5.3 Объяснить, почему выбраны шестнадцатиразрядные числа 0, 1, 11, 12, 13, 14, 15, 16, 17 для анализа работы сумматора?
- 5.4 Как обеспечить вычитание двух чисел?

6. Список литературы

Основная литература

- 1. Максимов Н. В. Архитектура ЭВМ и вычислительные системы: учебник / Н. В. Максимов, Т. Л. Партыка, И. И. Попов. - 4-е изд., перераб. и доп. - М.: ФОРУМ, 2012. - 512
- 2. Просветов Г.И. Дискретная математика: задачи и решения. – М.: Бином. Лаборатория знаний, 2010. – 222с.
- 3. Партыка Т. Л. Периферийные устройства вычислительной техники : учеб. пособие / Т. Л. Партыка, И. И. Попов. - 3-е изд., испр. и доп. - М. : ФОРУМ, 2012. - 432 с.
- 4. Юров В.И. Assembler. Учебник для вузов, 2-е издание. – СПб.: Питер, 2010. – 637с.

Дополнительная литература

- 4. Ижевский А.С., Калиниченко Б.Б. Методические указания для лабораторных работ «Моделирование и анализ электрических схем на основе логических элементов Electronic Workbench», 2009. - 38с.
- 5. Келим Ю.М. Вычислительная техника.- М.:Академия,2007. - 384с.
- 6. Вычислительная техника: метод. пособие /авт.-сост. Б.В. Цыбаков.- М., 2004. - 112с.

Тема 3.8. Распределители, цифровые компараторы

Лабораторная работа №6. Изучение работы цифрового компаратора

Время, отводимое на выполнение работы - 2 часа.

Перечень необходимых технических средств обучения:

- ✓ персональные компьютеры Core i3 3.0, 4 Gb, 4 ГБ ОЗУ, 250 Gb HDD;
- ✓ локальная сеть;
- ✓ коммутатор для подключения в сети Internet.

Перечень необходимых программных средств обучения:

- ✓ ОС Windows XP (7);
- ✓ Программа для моделирования Electronic Work Bench 5.12 и выше

1. **Цель занятия:** Изучение назначения устройства и принцип работы цифрового компаратора.

2. **Основные теоретические положения:**

Цифровой компаратор или **компаратор амплитуд** является электронным устройством берущим два числа в двоичном виде и определяющим является ли первое число меньшим, большим или равным второму числу. Компараторы используются в центральных процессорах и микроконтроллерах. Примерами цифровых компараторов являются КМОП — 4063 и 4585, ТТЛ — 7485 и 74682-89. Аналоговым эквивалентом цифрового компаратора является компаратор напряжений. Некоторые микроконтроллеры имеют аналоговые компараторы на некоторых своих входах, которые могут быть считаны или включать прерывание.

Цифровые компараторы (от английского compare – сравнивать) выполняют сравнение двух чисел A , B одинаковой разрядности, заданных в двоичном или двоично-десятичном коде. В зависимости от схемного исполнения компараторы могут определять равенство $A=B$ или неравенства $A<B$, $A>B$. Результат сравнения отображается в виде логического сигнала на одноименных выходах, в случае выполнения условия на выходе 1.

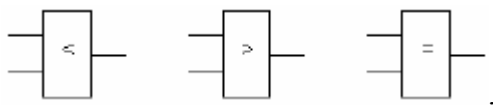
Цифровые компараторы применяются для выявления нужного числа (слова) в цифровых последовательностях, для выполнения условных переходов.

Схемы сравнения, или компаратор, обычно строятся как поразрядные. Они широко используются и автономно, и в составе более сложных схем, например при построении сумматоров.

Таблица 23. Таблица истинности компаратора

Входы		Выходы
a_i	b_i	Y_i
0	0	1
0	1	0
1	0	0
1	1	1

Условно обозначим логические схемы компараторов:



Операциям сравнения ($A<B$, $A=B$, $A>B$) соответствуют структурные формулы ($A'\cdot B$, $A'\cdot B'+A\cdot B$, $A\cdot B'$).

Тогда компараторы $\Rightarrow$, $\Leftarrow$, $\Diamond$ будут выглядеть следующим образом:

, где - элемент отрицания.

Логические функции компараторов ($\Leftarrow$, $\Rightarrow$, $\Diamond$) выглядят: $(A' \cdot B)'$, $(A \cdot B')$, $(A' \cdot B' + A \cdot B)'$.

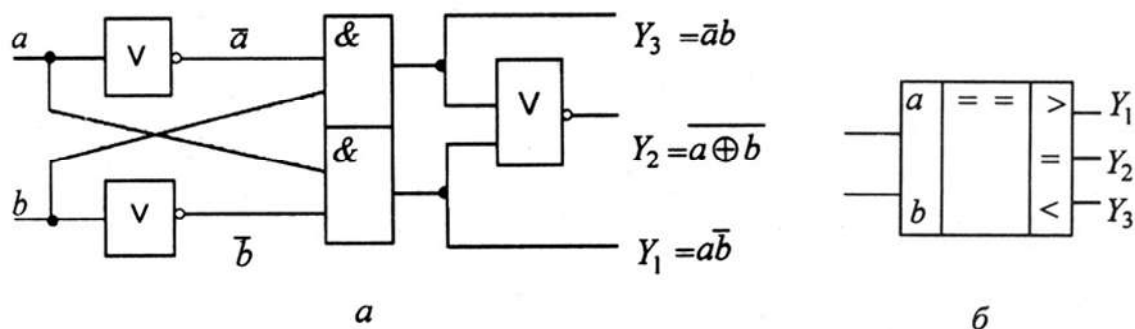


Рисунок 48 Структурная схема компаратора (а) и обозначение компаратора на принципиальных электрических схемах (б)

3. Задание к работе

3.1 Построение схем одноразрядных компараторов.

3.1.1 Постройте логическую схему для функции $f = A' \cdot B$. Результат выполнения задания представлен ниже:

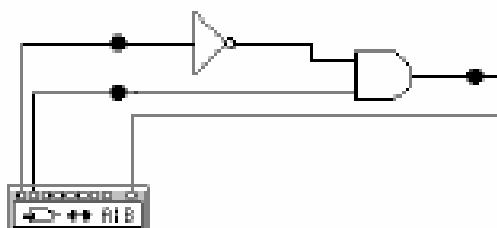


Рисунок 49. Логическая схема операции $<$

3.1.2 Постройте логическую схему для функции $f = A \cdot B'$. Результат выполнения задания представлен ниже:

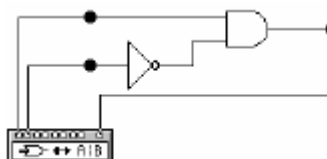


Рисунок 50. Логическая схема операции $>$

3.1.3 Постройте логическую схему для функции $f = A' \cdot B' + A \cdot B$. Результат выполнения задания представлен ниже:

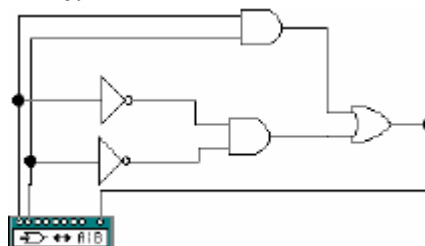


Рисунок 51. Логическая схема операции $=$

3.1.4 Найдите логические функции компараторов $\Rightarrow$, $\Leftarrow$, $\Diamond$, используя закон де Моргана и закон двойного отрицания.

- 3.1.5 Составьте схемы устройств, удовлетворяющие условиям $A \leq B$, $A < B$, $A \geq B$.
- 3.1.6 Составьте таблицы истинности (заполните таблицу 17) для созданных схем цифровых компараторов.

Таблица 24 Таблица истинности для формул работы схемы компаратора

A	B	B'	$A+B'$ ($A \Rightarrow B$)	A'	$A'+B$ ($A \leq B$)	$A+B$	$A'+B'$	$(A+B) \cdot (A'+B')$ ($A < B$)
---	---	----	---------------------------------	----	--------------------------	-------	---------	--------------------------------------

4. Содержание отчёта:

- 4.1 Построенные логические схемы, реализующие все функции компаратора.
- 4.2 Составленная таблица истинности для анализа работы компаратора.

5. Контрольные вопросы:

- 5.1 Какие функции выполняет цифровой компаратор, в каких устройствах он может быть использован?
- 5.2 Составьте схему устройства, объединяющую все три компаратора.
- 5.3 Исследуйте составленные схемы устройств.
- 5.4 Операция сложения по модулю 2. Таблица истинности. Обозначение на схеме.

6. Список литературы

Основная литература

1. Максимов Н. В. Архитектура ЭВМ и вычислительные системы: учебник / Н. В. Максимов, Т. Л. Партыка, И. И. Попов. - 4-е изд., перераб. и доп. - М.: ФОРУМ, 2012. - 512
2. Просветов Г.И. Дискретная математика: задачи и решения. – М.: Бином. Лаборатория знаний, 2010. – 222с.
3. Партыка Т. Л. Периферийные устройства вычислительной техники : учеб. пособие / Т. Л. Партыка, И. И. Попов. - 3-е изд., испр. и доп. - М. : ФОРУМ, 2012. - 432 с.
4. Юров В.И. Assembler. Учебник для вузов, 2-е издание. – СПб.: Питер, 2010. – 637с.

Дополнительная литература

1. Ижевский А.С., Калиниченко Б.Б. Методические указания для лабораторных работ «Моделирование и анализ электрических схем на основе логических элементов Electronic Workbench», 2009. - 38с.
2. Келим Ю.М. Вычислительная техника.- М.:Академия,2007. - 384с.
3. Вычислительная техника: метод. пособие /авт.-сост. Б.В. Цыбаков.- М., 2004. - 112с.

Лабораторная работа №7. Устройства контроля четности

Время, отводимое на выполнение работы - 2 часа.

Перечень необходимых технических средств обучения:

- ✓ персональные компьютеры Core i3 3.0, 4 Gb, 4 ГБ ОЗУ, 250 Gb HDD;
- ✓ локальная сеть;
- ✓ коммутатор для подключения в сети Internet.

Перечень необходимых программных средств обучения:

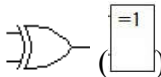
- ✓ ОС Windows XP (7);
- ✓ Программа для моделирования Electronic Work Bench 5.12 и выше

1. **Цель занятия:** Изучение назначения и принцип работы устройства контроля четности

2. **Основные теоретические положения:**

Устройства контроля на четность находят широкое применение в АСУ, системах управления, связи, вычислительных системах, системах передачи данных для оперативного обнаружения возникающих ошибок. Задачей устройства контроля на четность является выдача сигнала ошибок при поступлении на его входы комбинации сигналов, содержащей нечетное число единиц.

Сложение по модулю 2, логическое сложение, исключáющее íли, строгая дизъюнкция, (англ. *Xor*) — булева функция (логическая операция), результат выполнения которой является истинным, если только один из аргументов является истинным. На схеме

обозначается , т. е. сумматоров, сигналом переноса которых пренебрегают.

В вычислительной технике и сетях передачи данных **битом чётности** (англ. Parity bit) называют контрольный бит, служащий для проверки общей чётности двоичного числа (чётности количества единичных битов в числе).

Контроль некой двоичной последовательности (например, машинного слова) с помощью бита чётности также называют **контролем по паритету**.

Бит чётности или контрольный разряд формируется при выполнении операции «Исключающее-ИЛИ» поразрядно. Рассмотрим схему, использующую девятибитные кодовые слова, состоящие из восьми бит данных, за которыми следует бит чётности.

Число 10111101 содержит 6 '1' битов. Бит чётности будет 0, получаем кодовое слово 101111010.

Число 01110011 содержит 5 '1' битов. Бит чётности будет 1, получаем кодовое слово 011100111.

Число 00000000 не содержит '1' битов. Бит чётности будет 0, получаем кодовое слово 000000000.

Пустой или несуществующий поток битов также имеет ноль единичных битов, поэтому бит чётности будет 0.

Операция контроля четности двоичных чисел позволяет повысить надежность передачи и обработки информации. Ее сущность заключается в суммировании по модулю 2 всех разрядов с целью выяснения четности числа, что позволяет выявить наиболее вероятную ошибку в одном из разрядов двоичной последовательности.

Например, если при передаче кода 1001 произойдет сбой во втором разряде, то на приемном пункте получим код 1101 – такую ошибку определить в общем случае затруднительно. Если же код относится к двоично-десятичному (способ кодирования десятичных чисел, при котором каждая цифра представляется четырьмя двоичными разрядами – двоичной тетрадой), обнаружение ошибок путем введения дополнительного бита четности происходит следующим образом:

На передающей стороне передаваемый код анализируется и дополняется контрольным битом до четного или нечетного числа единиц в суммарном коде.

Соответственно суммарный код называется четным или нечетным. В случае нечетного кода дополнительный бит формируется таким образом, чтобы сумма всех единиц в передаваемом коде, включая контрольный бит, была нечетной. При контроле четности все наоборот. Например, в числе 0111 число единиц нечетно. Поэтому при контроле нечетности дополнительный код должен быть нулем, а при контроле четности – единицей. На практике чаще всего используется контроль нечетности, поскольку он позволяет фиксировать полное пропадание информации (случай нулевого кода во всех информационных разрядах). На приемной стороне производится проверка кода четности. Если он правильный, то прием разрешается, в противном случае включается сигнализация ошибки или посылается передатчику запрос на повторную передачу.

Схема формирования бита четности для четырехразрядного кода приведена на рисунке ниже, она содержит четыре элемента исключающие ИЛИ, выполняющие функции сумматоров по модулю 2 (без переноса) и состоит из трех ступеней:

1. На первой ступени попарно суммируются все биты исходного кода на входах А, В, С, D.
2. На второй ступени анализируются сигналы первой ступени, и устанавливается четность или нечетность суммы входного кода.
3. На третьей ступени полученный результат сравнивается с контрольным сигналом на входе Е, задающим *вид используемого контроля*, в результате чего на выходе F формируется дополнительный пятый бит четности, сопровождающий информационный сигнал в канале передачи.

3. Задание к работе

3.1 Построение схемы контроля чётности для четырёхразрядного кода

3.1.1 Постройте схему контроля чётности (см. рисунок 52).

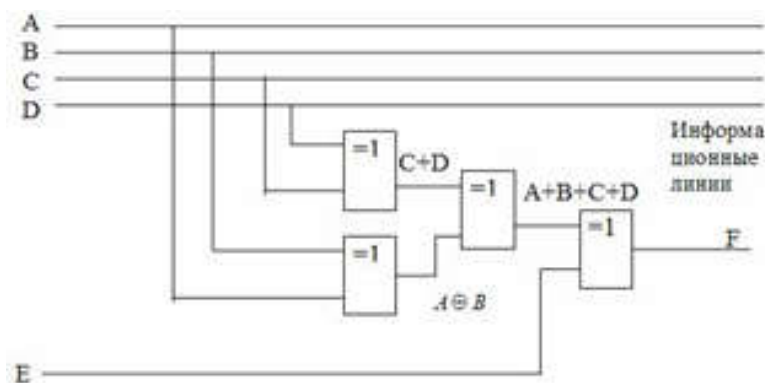


Рисунок 52 Схема контроля чётности

- 3.1.2 Добавьте Logic Converter. Подключите входы к клеммам А, В, С, D, Е.
- 3.1.3 Получите таблицу истинности по построенной схеме. Проанализируйте каждую строку таблицы. Сколько всего возможных двоичных комбинаций?
- 3.1.4 Получите СДНФ по таблице. Упростите формулу.
- 3.1.5 Самостоятельно постройте схему формирователя бита четности трехразрядного (пятиразрядного) кода

3.2 Построение схемы формирования паритетного бита:

- 3.2.1 Добавьте ИМС 74280 (аналог — К555ИП15). ИМС 74280 имеет 9 входов (А, В...I) и два выхода (EVEN, ODD), один из которых — инверсный (even – бит при четном, odd – нечетном числе единиц). Вход I используется для управления видом контроля (0 — контроль четности, 1 — контроль нечетности) и управляется переключателем Z (управляется с клавиатуры одноименной клавишей). Вывод NC — not connection — пустой, т.е. внутри ИМС к нему ничего не подключено.

- 3.2.2 Добавьте Word Generator. Проверьте правильность функционирования схемы. Для каких целей используется переключатель Z? на входы рассматриваемого устройства подаются различные двоичные комбинации; состояние выходов ИМС контролируется подключенными к ним светоиндикаторами.

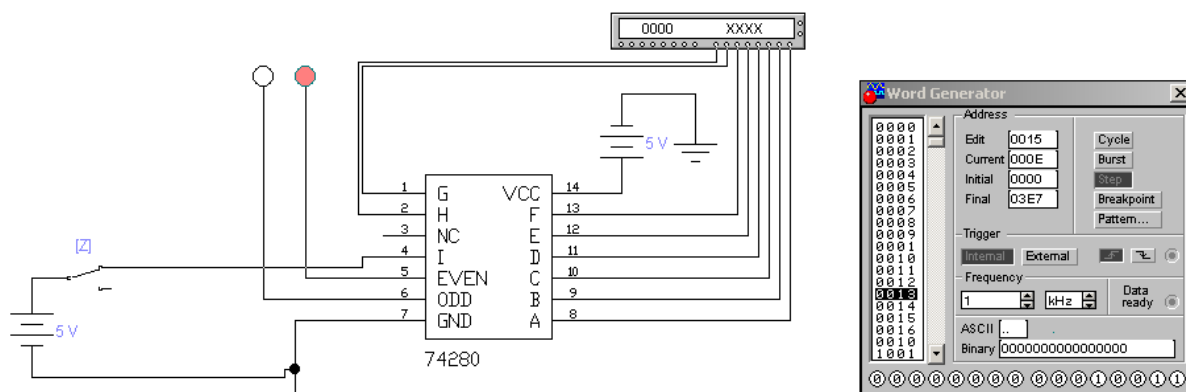


Рисунок 53 Схема формирования паритетного бита

3.3 Самостоятельно постройте устройство контроля на чётность 16-и разрядного кода, с использованием ИМС 74280

4. Содержание отчёта:

- 4.1 Схема контроля чётности для четырёхразрядного кода
- 4.2 Схема формирования паритетного бита
- 4.3 Схема устройства контроля на чётность 16-и разрядного кода, с использованием ИМС 74280

5. Контрольные вопросы:

- 5.1 Что такое бит чётности? Что такое контроль по паритету?
- 5.2 Какое назначение имеют формирователи кода четности, где они могут быть использованы?
- 5.3 Проанализируйте работу составленных схем формирователей битов четности

6. Список литературы

Основная литература

1. Максимов Н. В. Архитектура ЭВМ и вычислительные системы: учебник / Н. В. Максимов, Т. Л. Партыка, И. И. Попов. - 4-е изд., перераб. и доп. - М.: ФОРУМ, 2012. - 512
2. Просветов Г.И. Дискретная математика: задачи и решения. – М.: Бином. Лаборатория знаний, 2010. – 222с.
3. Партыка Т. Л. Периферийные устройства вычислительной техники : учеб. пособие / Т. Л. Партыка, И. И. Попов. - 3-е изд., испр. и доп. - М. : ФОРУМ, 2012. - 432 с.
4. Юров В.И. Assembler. Учебник для вузов, 2-е издание. – СПб.: Питер, 2010. – 637с.

Дополнительная литература

1. Келим Ю.М. Вычислительная техника.- М.:Академия,2007. - 384с.
2. Ижевский А.С., Калиниченко Б.Б. Методические указания для лабораторных работ «Моделирование и анализ электрических схем на основе логических элементов Electronic Workbench», 2009. - 38с.
3. Вычислительная техника: метод. пособие /авт.-сост. Б.В. Цыбаков.- М., 2004. - 112с.

Раздел 4. Устройства ЭВМ

Тема 4.1. Запоминающие устройства ЭВМ

Лабораторная работа №8. Принцип работы кэш-памяти. Алгоритмы замещения строк

Время, отводимое на выполнение работы - 2 часа.

Перечень необходимых технических средств обучения:

- ✓ персональные компьютеры Core i3 3.0, 4 Gb, 4 ГБ ОЗУ, 250 Gb HDD;
- ✓ локальная сеть;
- ✓ коммутатор для подключения в сети Internet.

Перечень необходимых программных средств обучения:

- ✓ ОС Windows XP (7);
- ✓ Программа для моделирования Electronic Work Bench 5.12 и выше
- ✓ Программа «Учебная ЭВМ».

1. **Цель занятия:** знакомство с алгоритмами замещения строк кэш-памяти в учебной ЭВМ, научиться настраивать параметры кэш-памяти, проверить работу различных алгоритмов замещения при различных режимах записи.

2. **Основные теоретические положения:**

1.1 Оперативно-запоминающее устройство

Оперативные запоминающие устройства (ОЗУ) являются неотъемлемой частью микропроцессорных систем различного назначения. ОЗУ делятся на два класса: статические и динамические. В статических ОЗУ запоминание информации производится на триггерах, а в динамических – на конденсаторах емкостью 0,5 пФ. Длительность хранения информации в статических ОЗУ не ограничена, тогда как в динамических ОЗУ она ограничена временем саморазряда конденсатора, что требует специальных средств регенерации и дополнительных затрат времени на этот процесс.

На рисунке 69 показана ячейка статического ОЗУ на D-триггере и вспомогательных логических элементах. Информационный вход ячейки подключен к шине данных D1 одного из разрядов, ее выход – к соответствующей шине D0 через элемент с тремя состояниями U6. Ячейка выбирается сигналами Y=1, X=1, поступающими с дешифратора адреса.

При записи в ячейку памяти на D1 устанавливается 1 или 0, на входе WR/RD' – сигнал 1, в результате чего срабатывают элементы 2 и U1, U2. Положительный перепад сигнала с элемента U2 поступает на тактовый вход D-триггера U4 и в нем записывается 1 или 0 в зависимости от уровня сигнала на его D-входе. При чтении на входе WR/RD' устанавливается 0, при этом срабатывают элементы U1, U3, U5 и на вход РАЗРЕШЕНИЕ ВЫХОДА буферного элемента U6 поступает разрешающий сигнал, в результате чего сигнал с Q-выхода D-триггера передается на разрядную шину D0, состояние которой индицируется логическим пробником IND.

Заметим, что запоминающие устройства статического типа отличаются высоким быстродействием и в компьютерах используются в качестве так называемой кэш-памяти.

1.2 Структура кэш-памяти в учебной ЭВМ

Кэш микропроцессора — кэш (сверхоперативная память), используемый микропроцессором компьютера для уменьшения среднего времени доступа к компьютерной памяти.

Кэш-память содержит N ячеек (в модели N может выбираться из множества $\{4, 8, 16, 32\}$), каждая из которых включает трехразрядное поле тега (адреса ОЗУ), шестизначное поле данных и три однобитовых признака (флага):

- Z — признак занятости ячейки;
- U — признак использования;
- W — признак записи в ячейку.

Таким образом, каждая ячейка кэш-памяти может дублировать одну любую ячейку ОЗУ, причем отмечается ее занятость (в начале работы модели все ячейки кэш-памяти свободны), факт записи информации в ячейку во время пребывания ее в кэш-памяти, а также использование ячейки (т.е. любое обращение к ней).

1.3 Настройка параметров кэш-памяти

Для настройки параметров кэш-памяти можно воспользоваться диалоговым окном **Кэш-память**, вызываемым командой **Вид | Кэш-память**, а затем нажать первую кнопку на панели инструментов открытого окна. После этих действий появится диалоговое окно **Параметры кэш-памяти**, позволяющее выбрать *размер* кэш-памяти, *способ записи* в нее информации и *алгоритм замещения* ячеек.

При доступе процессора в память сначала производится проверка, хранит ли кэш запрашиваемые из памяти данные. Для этого производится сравнение адреса запроса со значениями всех тегов кэша, в которых эти данные могут храниться. Случай совпадения с тегом какой-либо кэш-линии называется **попаданием** в кэш (англ. cache hit), обратный же случай называется кэш **промахом** (англ. cache miss). Попадание в кэш позволяет процессору немедленно произвести чтение или запись данных в кэш-линии с совпавшем тегом. Отношение количества попаданий в кэш к общему количеству запросов к памяти называют **рейтингом попаданий** (англ. hit rate), оно является мерой эффективности кэша для выбранного алгоритма или программы.

При **сквозной записи** при кэш-попадании в процессорных циклах записи осуществляется запись как в ячейку памяти, так и в ячейку ОЗУ, а **при обратной записи** — только в ячейку кэш-памяти, причём эта ячейка отмечается битом записи $W=1$.

При кэш-промахе следует поместить в кэш-память адресуемую процессором ячейку. При наличии свободных ячеек кэш-памяти требуемое слово помещается в одну из них (в порядке очереди). При отсутствии свободных ячеек следует отыскать ячейку кэш-памяти, содержимое которой можно удалить. Поиск такой ячейки осуществляется с использованием алгоритма замещения строк (в учебной модели):

- *случайное замещение* (номер ячейки кэш-памяти выбирается случайным образом);
- *очередь* (номер ячейки выбирается временем пребывания её в кэш-памяти);
- *бит использования* (случайный выбор осуществляется только в тех ячейках, которые имеют нулевое значение флага использования);

Если в параметрах кэш-памяти установлен флаг «с учётом бита записи», то все три алгоритма замещения осуществляют поиск «кандидата на удаление» прежде всего среди тех ячеек, признак записи которых не установлен.

1.4 Оценка эффективности работы системы с кэш-памятью

Оценка эффективности работы системы с кэш-памятью определяется числом «кэш-попаданий» по отношению к общему числу обращений к памяти:

$$K = \frac{S_k - S_{k_{\text{не}}}}{S_0}, \quad K = \frac{S_k - S_{k_{\text{не}}}^i}{S_0}, \quad \text{где}$$

K — коэффициент эффективности работы кэш-памяти;

S_0 — общее число обращений к памяти;

S_k — число кэш-попаданий;

- $S_{K_{WR}}$ – число сквозных записей при кэш-попадании (в режиме сквозной записи);
 $S_{K_{WR}}^i$ – число обратных записей (в режиме обратной записи).

3. Задание к работе

3.1 Смоделируйте и проанализируйте работу ОЗУ, схема которого приведена ниже:

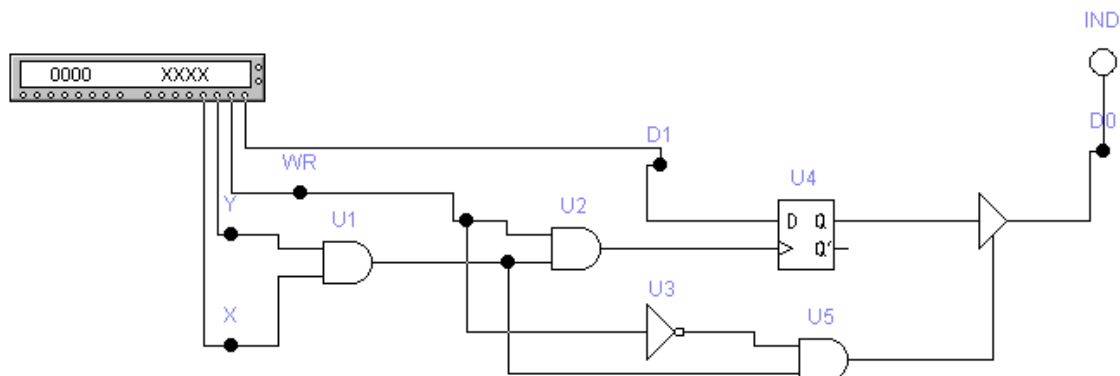


Рисунок 7 Схема ячейки статического ОЗУ на D-триггере и вспомогательных логических элементах

Для проверки функционирования ячейки памяти используется генератор слова, выходной код которого соответствует указанным режимам работы ячейки. Введите слова в генератор слов: 0000, 000F, 000F, 000C, 0000, 000E, 0000, 000C

3.2 Запустите программу учебная ЭВМ. В качестве задания предлагается некоторая короткая "программа", которую необходимо выполнить с подключенной кэш-памятью (размером 4 и 8 ячеек) в шаговом режиме для следующих двух вариантов алгоритмов замещения (см.табл. 45 Варианты задания).

Таблица 45. Варианты задания для работы с кэш-памятью

Номера вариантов	Режим записи	Алгоритм замещения
1,7	Сквозная Обратная	СЗ, без учёта бита записи О, с учётом бита записи
2,5,9	Сквозная Обратная	БИ, без учёта бита записи О, с учётом бита записи
3,6	Сквозная Обратная	О, без учёта бита записи СЗ, с учётом бита записи
4,8,10	Сквозная Обратная	БИ, без учёта бита записи БИ, с учётом бита записи

Таблица 46. Программы для анализа работы с кэш-памятью

№ вариант а	Номера команд программы						
	1.	2.	3.	4.	5.	6.	7.
1.	RD #12	WR 10	WR @10	ADD 12	WR R0	SUB 10	PUSH R0
2.	RD #65	WR R2	MOV R4, R2	WR 14	PUSH R2	POP R3	CALL 002

3.	RD #16	SUB #5	WR 9	WR @9	WR #3	PUSH R3	POP R4
4.	RD #99	WR R6	MOV R7, R6	ADD R7	PUSH R7	CALL 006	POP R8
5.	RD #11	WR R2	WR -@R2	PUSH R2	CALL 005	POP R3	RET
6.	RD #19	SUB #10	WR 9	ADD #3	WR @9	CALL 006	POP R4
7.	RD #6	CALL 006	WR 11	WR R2	PUSH R2	RET	JMP 002
8.	RD #8	WR R2	WR @R2+	PUSH R2	POP R3	WR -@R3	CALL 003
9.	RD #13	WR 14	WR @14	WR @13	ADD 13	CALL 006	RET
10.	RD #42	SUB #54	WR 16	WR @16	WR R1	ADD @R1+	PUSH R1

Не следует рассматривать заданную последовательность команд в таблице 46 как фрагмент программы! Некоторые конструкции введены в задание для того, чтобы обратить внимание на особенности функционирования стека.

- 3.2.1 Ввести в модель учебной ЭВМ текст своего варианта программы.
- 3.2.2 Ассемблировать его и сохранить на диске в виде txt-файл.
- 3.2.3 Установить параметры кэш-памяти размером 4 ячейки, выбрать режим записи и алгоритм замещения в соответствии с первой строкой своего варианта из таблицы с заданием.
- 3.2.4 В шаговом режиме выполнить программу, фиксируя после каждого шага состояние кэш-памяти.
- 3.2.5 Для одной из команд записи (WR) перейти в режим **Такт** и отметить, в каких микрокомандах происходит изменение кэш-памяти.
- 3.2.6 Для кэш-памяти размером 8 ячеек установить параметры в соответствии со второй строкой своего варианта из таблицы и выполнить программу и шаговом режиме еще раз, фиксируя последовательность номеров замещаемых ячеек кэш-памяти.

3.3 Вычисление коэффициента эффективности работы системы с кэш-памятью

- 3.3.1 В меню **Работа** установить режим **Кэш-память**
- 3.3.2 В меню **Вид** выбрать команду **Кэш-память**, открыв тем самым окно **Кэш-память**, в нем нажать первую слева кнопку на панели инструментов, открыв диалоговое окно **Параметры кэш-памяти**, и установить следующие параметры кэш-памяти: размер — 4, режим записи — сквозная, алгоритм замещения — случайное, без учета бита записи (W).
- 3.3.3 Запустить программу в автоматическом режиме; по окончании работы просмотреть результаты работы кэш-памяти в окне **Кэш-память**, вычислить значение коэффициента эффективности K и записать в ячейку табл., помеченную звездочкой.
- 3.3.4 Выключить кэш-память модели (**Работа | Кэш-память**) и изменить один из ее параметров — установить флаг с учетом бита записи (в окне **Параметры кэш-памяти**).
- 3.3.5 Повторить п. 3.3.4, поместив значение полученного коэффициента эффективности в следующую справа ячейку табл.

- 3.3.6 Последовательно меняя параметры кэш-памяти, повторить пп. 3—5, заполняя все ячейки табл. При очередном запуске программы не забывайте устанавливать процессор модели в начальное состояние, нажимая кнопку **R** в окне **Процессор!**
- 3.3.7 Повторить все действия, описанные в пп. 1—7, заполняя вторую таблицу по форме табл.47. **Результаты эксперимента**

Таблица 47. Результаты эксперимента

Способ	Сквозная запись					
Алгоритм	Случайное замещение		Очередь		Бит U	
Размер	Без W	C W	Без W	C W	Без W	C W
4	*					
8						
16						
32						
Способ	Обратная запись					
Алгоритм	Случайное замещение		Очередь		Бит U	
Размер	Без W	C W	Без W	C W	Без W	C W
4						
8						
16						
32						

4. Содержание отчёта:

- 6.1 Вариант задания — текст программы и режимы кэш-памяти.
- 6.2 Последовательность состояний кэш-памяти размером 4 ячейки при однократном выполнении программы (команды 1—7).
- 6.3 Последовательность микрокоманд при выполнении команды wt с отметкой тех микрокоманд, в которых возможна модификация кэш-памяти.
- 6.4 Для варианта кэш-памяти размером 8 ячеек — последовательность номеров замещаемых ячеек кэш-памяти для второго варианта параметров кэш памяти при двукратном выполнении программы (команды 1—7).

5. Контрольные вопросы:

- 5.1 В чем смысл включения кэш-памяти в состав ЭВМ?
- 5.2 Как работает кэш-память в режиме обратной записи? Сквозной записи?
- 5.3 Как зависит эффективность работы ЭВМ от размера кэш-памяти?
- 5.4 Какие алгоритмы замещения ячеек кэш-памяти вам известны?

6. Список литературы

Основная литература

1. Максимов Н. В. Архитектура ЭВМ и вычислительные системы: учебник / Н. В. Максимов, Т. Л. Партыка, И. И. Попов. - 4-е изд., перераб. и доп. - М.: ФОРУМ, 2012. - 512
2. Просветов Г.И. Дискретная математика: задачи и решения. – М.: Бином. Лаборатория знаний, 2010. – 222с.
3. Партыка Т. Л. Периферийные устройства вычислительной техники : учеб. пособие / Т. Л. Партыка, И. И. Попов. - 3-е изд., испр. и доп. - М. : ФОРУМ, 2012. - 432 с.
4. Юров В.И. Assembler. Учебник для вузов, 2-е издание. – СПб.: Питер, 2010. – 637с.

Дополнительная литература

1. Горнец Н.Н. Организация ЭВМ и систем. - М.: Академия, 2008. - 320с. Жмакин А.П. Архитектура ЭВМ. – СПб: БХВ-Петербург, 2006. – 320с.
2. Келим Ю.М. Вычислительная техника.- М.: Академия, 2007. - 384с.
3. Вычислительная техника: метод. пособие /авт.-сост. Б.В. Цыбаков.- М., 2004. - 112с.

Тема 4.2 Процессоры ЭВМ

Лабораторная работа №9. Моделирование работы арифметико-логического устройства

Время, отводимое на выполнение работы - 2 часа.

Перечень необходимых технических средств обучения:

- ✓ персональные компьютеры Core i3 3.0, 4 Gb, 4 ГБ ОЗУ, 250 Gb HDD;
- ✓ локальная сеть;
- ✓ коммутатор для подключения в сети Internet.

Перечень необходимых программных средств обучения:

- ✓ ОС Windows XP (7);
- ✓ Программа для моделирования Electronic Work Bench 5.12 и выше

1. **Цель занятия:** Ознакомление с возможностями моделирования работы арифметико-логического устройства (АЛУ). Исследование работы АЛУ на примере выполнения арифметических и логических операций.
2. **Основные теоретические положения:**

Арифметико-логическое устройство предназначено для выполнения арифметических и логических операций над многоразрядными операндами в зависимости от кодов, подаваемых на управляющие входы.

В вычислительных устройствах АЛУ является базовым узлом и работает в сочетании с ОЗУ, регистрами сдвига, регистрами общего назначения и др. Микросхемы АЛУ, принадлежащие к разным видам логик, функционально во многом совпадают. Аналогом микросхемы K155ИПЗ в программе EWB имеется микросхема 74181. Она представляет собой четырехразрядное АЛУ.

АЛУ работает в режиме выполнения логических операций при значении управляющего сигнала $M=1$ и в режиме выполнения арифметических операций при значении управляющего сигнала $M=0$.

Эта микросхема обеспечивает 32 режима работы АЛУ в зависимости от управляющих сигналов на входах M, S_0-S_3 . Возможные режимы задаются путем подачи сигналов на входы управления S_0, S_1, S_2, S_3 . Если сигнал на входе M равен 0, то выполняются 16 арифметических операций (16 комбинаций сигналов $S_0...S_3$) с учетом переноса по входу CN (если $CN=0$) или без учета переноса (если $CN=1$). При сигнале на входе M равном 1, выполняются 16 логических операций, задаваемых S_0-S_3 .

Четырехразрядные операнды A и B задаются на входах A_0-A_3 и B_0-B_3 соответственно. Результат арифметической или логической операции появляется на выходах F_0-F_3 .

В микросхеме предусмотрены выходы переноса $CN+4$, ускоренного переноса P , ускоренного группового переноса G , равенства операндов $A=B$. Питание микросхемы осуществляется подключением источника питания 5 В ко входу VCC и заземлением входа GND .

В приведенной таблице содержатся выполняемые логические и арифметические операции в зависимости от кодовой комбинации на управляющих входах S_0, S_1, S_2, S_3 .

Таблица 5 Логические и арифметические операции

Выбор функции				Логические функции $M=1$	Арифметические функции, $M=0$	
S_3	S_2	S_1	S_0		С переносом $CN=0$	Без переноса $CN=1$
0	0	0	0	Not A	$A + 1$	A
0	0	0	1	not (A or B)	$A + B + 1$	A or B
0	0	1	0	(Not A) and B	$A + (\text{not } B) + 1$	A or (not B)
0	0	1	1	Логический 0	0	-1
0	1	0	0	Not (A and B)	$A + \text{Not } (A \text{ and } B) + 1$	$A + (A \text{ and } (\text{not } B))$
0	1	0	1	Not B	$(A \text{ or } B) + (A \text{ and } (\text{not } B)) + 1$	$(A+B) + (A \text{ and } (\text{not } B))$

					1	B))
0	1	1	0	A xor B	A – B	A–B–1
0	1	1	1	A and (not B)	Not (A and B)	(A and (not B)) – 1
1	0	0	0	(Not A) or B	A + B +1	A+(A and B)
1	0	0	1	Not (A xor B)	A + B + 1	A + B
1	0	1	0	B	(A or (not B)) + (A and B) + 1	(A or (not B)) + A and B
1	0	1	1	A and B	A and B	A and B – 1
1	1	0	0	Логическая 1	A + A + 1	A + A
1	1	0	1	A or (not B)	(A or B) + A + 1	(A or B) + A
1	1	1	0	A or B	(A or (not B)) + A + 1	(A or (not B)) + A
1	1	1	1	A	A	A – 1

Используемые элементы EWB.

На панели Digital ICs выбирается схема из множества 741xx. Под номером 74181 - Четырёхразрядное АЛУ.

3. Задание к работе

3.1 Построение АЛУ

3.1.1 Подготовить микросхему АЛУ на рабочем столе.

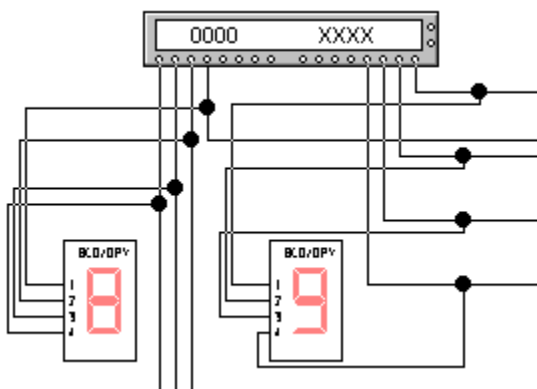


Рисунок 8 Фрагмент АЛУ

- 3.1.2 Подготовить 4 переключателя для задания режимов управления.
- 3.1.3 Сигналы S подаются на входы S0.....S3 АЛУ.
- 3.1.4 Ещё один переключатель используется для задания режима M.
- 3.1.5 И в том и другом случае логическая 1 подаётся от 5V источника постоянного тока, логический 0 от заземления.
- 3.1.6 Шестой переключатель осуществляет роль входного переноса и подключается таким же образом к входу CN.
- 3.1.7 Значения четырёхразрядных операндов A и B задаются с помощью генератора слова и в шестнадцатеричном коде отображаются на алфавитно-цифровых индикаторах и подсоединяются к входам A0....A3 и B0.....B3.
- 3.1.8 На выходах F0....F3 формируется результат операции АЛУ. Для отображения результата к выходам F0....F3 присоединяется алфавитно-цифровой индикатор.
- 3.1.9 К выходу VCC присоединить источник 5V.
- 3.1.10 К входу GND подсоединить заземление.
- 3.1.11 При коде 1111 на выходах F и при равенстве операндов выход A=B переводится в единичное состояние. Поскольку этот выход представляет собой каскад с открытым коллектором, то на него подаётся питание +5 вольт через резистор 1 кОм. Выход A=B совместно с выходом переноса CN+4 и выходом P

подтверждения переноса используются для формирования признаков $A > B$ и $A < B$ с помощью дополнительных логических элементов ИЛИ-НЕ и НЕ.

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

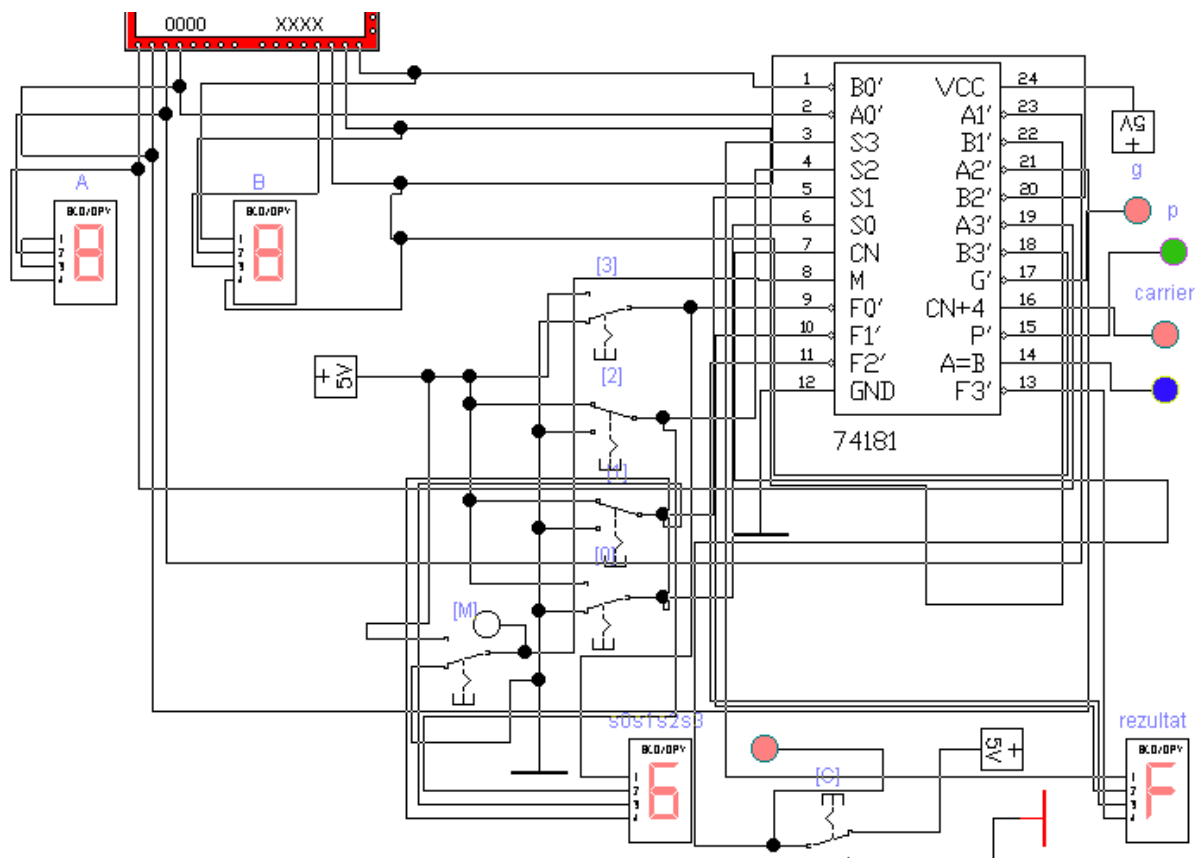


Рисунок 9 АЛУ

3.2 Проведите моделирование перечисленных в таблице 5 режимов.

4. Содержание отчёта:

4.1 Схема работы АЛУ

4.2 Моделирование работы логических и арифметических операций.

5. Контрольные вопросы:

5.1 Назначение входа переноса в АЛУ.

5.2 Чем отличаются логические операции от арифметических операций?

5.3 Как можно выполнить операцию инкремента?

6. Список литературы

Основная литература

1. Максимов Н. В. Архитектура ЭВМ и вычислительные системы: учебник / Н. В. Максимов, Т. Л. Партыка, И. И. Попов. - 4-е изд., перераб. и доп. - М.: ФОРУМ, 2012. - 512
2. Просветов Г.И. Дискретная математика: задачи и решения. – М.: Бином. Лаборатория знаний, 2010. – 222с.
3. Партыка Т. Л. Периферийные устройства вычислительной техники : учеб. пособие / Т. Л. Партыка, И. И. Попов. - 3-е изд., испр. и доп. - М. : ФОРУМ, 2012. - 432 с.
4. Юров В.И. Assembler. Учебник для вузов, 2-е издание. – СПб.: Питер, 2010. – 637с.

Дополнительная литература

1. Ижевский А.С., Калининко Б.Б. Методические указания для лабораторных работ «Моделирование и анализ электрических схем на основе логических элементов Electronic Workbench», 2009. - 38с.
2. Келим Ю.М. Вычислительная техника.- М.:Академия,2007. - 384с.
3. Вычислительная техника: метод. пособие /авт.-сост. Б.В. Цыбаков.- М., 2004. - 112с.

Тема 5.2 Архитектура МПС

Лабораторная работа №10. Архитектура ЭВМ и система команд. Вычисление значения выражения с применением различных способов адресации

Время, отводимое на выполнение работы - 2 часа.

Перечень необходимых технических средств обучения:

- ✓ персональные компьютеры Core i3 3.0, 4 Gb, 4 Гб ОЗУ, 250 Gb HDD;
- ✓ локальная сеть;
- ✓ коммутатор для подключения в сети Internet.

Перечень необходимых программных средств обучения:

- ✓ ОС Windows XP (7);
- ✓ Программа «Учебная ЭВМ»

1. **Цель занятия:** Познакомиться с интерфейсом учебной ЭВМ, узнать методы ввода и отладки программы, действия основных команд и способы адресации. Приобрести опыт программирования учебной ЭВМ в машинных кодах.

2. **Основные теоретические положения:**

2.1 Понятие программы

Программа на языке ЭВМ представляет собой последовательность команд. Код каждой команды определяет выполняемую операцию, тип адресации и адрес. Выполнение программы, записанной в памяти ЭВМ, осуществляется последовательно по командам в порядке возрастания адресов команд или в порядке, определяемом командами передачи управления.

Для того чтобы получить результат выполнения программы, пользователь должен:

- ✓ ввести программу в память ЭВМ;
- ✓ определить, если это необходимо, содержимое ячеек ОЗУ и РОН, содержащих исходные данные, а также регистров IR и BR;
- ✓ установить в РС стартовый адрес программы;
- ✓ перевести модель в режим **Работа**.

Ввод программы может осуществляться как в машинных кодах непосредственно в память модели, так и в мнемокодах в окне **программы** с последующим ассемблированием.

Команды в память учебной ЭВМ вводятся в виде шестизначных десятичных чисел, изменяющиеся в диапазоне "-99 999...+99 999", содержащие знак и 5 десятичных цифр. Старший разряд слова данных используется для кодирования знака: плюс (изображается как 0, минус (-) — как 1). Если результат арифметической операции выходит за пределы указанного диапазона, то говорят, что произойдет переполнение разрядной сетки. АЛУ этом случае вырабатывает сигнал переполнения $OV = 1$. Деление на ноль вызывает переполнение. Результатом операции деления является целая часть частного.

2.2 Типы адресации

- ✓ **Прямая**, например, **add 23** – указание в команде непосредственно исполнительного адреса
- ✓ **Непосредственная**, например, **add #33** – которая заключается в указании в команде самого значения операнда, а не его адреса
- ✓ **Косвенная**, например, **add @33** – при которой в команде указывается адрес регистра или ячейки памяти, в которых хранится адрес операнда или его составляющие

2.3 Система команд

При рассмотрении системы команд ЭВМ обычно анализируют три аспекта: форматы, способы адресации и систему операций.

В форматах команд выделяется три поля: два старших разряда (0, 1) определяют код операции СОР, разряд 2 может определять тип адресации, разряды [3:5] могут определять прямой или косвенный адрес памяти, номер регистра (в команде MOV номера двух регистров), адрес перехода или короткий непосредственный операнд. В двухсловных командах непосредственный операнд занимает поле [6:11].

Таблица 25 Система команд учебной ЭВМ

КОП	Мнемокод	Название	КОП	Мнемокод	Название
00	NOP	Пустая операция	23	ADD	Сложение
01	IN	Ввод Асс←IR	24	SUB	Вычитание
02	OUT	Вывод OR← Асс	25	MUL	Умножение
03	IRET	Возврат из прерывания	17	JNRZ	Цикл
10	JMP	Безусловный переход	30	MOV	Пересылка
11	JZ	Переход, если 0 (Асс=0)	36	DIV	Деление
12	JNZ	Переход, если не 0 (Асс≠0)	19	CALL	Вызов подпрограммы
14	JNS	Переход, если положительно	09	HLT	Стоп
13	JS	Переход, если отрицательно	41	RDI	Чтение
15	JO	Переход, если переполнение (если Асс>99999)	43	ADI	Сложение
18	INT	Программное прерывание	44	SBI	Вычитание
21	RD	Чтение	45	MULI	Умножение
22	WR	Запись	46	DIVI	Деление

I — непосредственный операнд;

2.4 Программно-доступные регистры и флаги:

Асс — аккумулятор;

РС — счетчик адреса команды, содержащий адрес текущей команд;

SP — указатель стека, содержащий адрес верхушки стека;

RB — регистр базового адреса, содержащий базовый адрес;

RA — регистр адреса, содержащий исполнительный адрес при кос ной адресации;

IR — входной регистр;

OR — выходной регистр;

I — флаг разрешения прерываний.

2.5 Системные регистры и флаги:

DR — регистр данных АЛУ, содержащий второй операнд;

MDR — регистр данных ОЗУ;

MAR — регистр адреса ОЗУ;

RDR — регистр данных блока РОН;

RAR — регистр адреса блока РОН;

CR — регистр команд, содержащий поля:

✓ **СОР** — код операции;

✓ **ТА** — тип адресации;

✓ **ADR** — адрес или непосредственный операнд;
Z — флаг нулевого значения Асс;
S — флаг отрицательного значения Асс;
OV — флаг переполнения.

3. Задание к работе

3.1 Дана последовательность мнемочкодов, которую необходимо преобразовать в машинные коды, занести в ОЗУ ЭВМ, выполнить в режиме Шаг и зафиксировать изменение состояний программно-доступных объектов ЭВМ.

3.1.1 Запишите следующую программу в окно **Текст программы**

Таблица 26 Пример программы

Коды	Программа	Действие
21 1 020	rd #20	
22 0 030	wr 30	
23 0 005	add #5	
22 2 030	wr @30	
12 0 002	JNZ 002	

3.1.2 В меню **Вид** выберите **Программа**, нажмите кнопку **Компилировать**

3.1.3 Выполняя команду **Шаг** фиксируем изменения программно-доступных объектов (Асс, ячейки ОЗУ 020 и 030) в таблице 27 **Содержание регистров**:

Таблица 27 Содержание регистров для задачи 1

PC	Асс	М(30)	М(20)
000	000000	000000	000000
001	000020		
002		000020	
003	000025		
004			000025
002			
003	000030		
004			000030

3.1.4 В тетради заполните столбец **Действие** для каждой используемой команды программы.

3.1.5 Нарисуйте алгоритм работы программы.

3.2 Запишите в ОЗУ программу, состоящую из 5 команд (из своего варианта). Команды разместите в ячейках памяти.

3.2.1 При необходимости установить начальное значение в устройство ввода IR.

3.2.2 Определить те программно-доступные объекты ЭВМ, которые будут изменяться при выполнении этих команд.

3.2.3 Выполнить в режиме Шаг введенную последовательность команд, фиксируя изменения значений объектов, определенных в п. 3.1.3, в таблице 27. Содержание регистров, по примеру:

Таблица 28 Содержание регистров для задачи 2

PC	Acc		
000			
001			
002			
003			

3.2.4 Если в программе образуется цикл, необходимо просмотреть не более двух повторений каждой команды, входящей в тело цикла.

Таблица 29 Задание для выполнения

Вариант №	IR	Команда 1	Команда 2	Команда 3	Команда 4	Команда 5
1	0000007	IN	MUL #2	WR 10	WR@10	JNS 001
2	x	RD #17	SUB #9	WR 16	WR @16	JNS 001
3	100029	IN	ADD #16	WR 8	WR @8	JS 001
4	x	RD #2	MUL #6	WR 11	WR @11	JNZ 00
5	000016	IN	WR 8	DIV #4	WR @8	JMP 002
6	x	RD #4	WR 11	RD @11	ADD #330	JS 000
7	000000	IN	WR 9	RD @9	SUB #1	JS 001
8	x	RD 4	SUB #8	WR 8	WR @8	JNZ 001
9	100005	IN	ADD #12	WR 10	WR @10	JS 004
10	x	RD 4	ADD #15	WR 13	WR @13	JMP 001

3.2.5 Напишите машинные коды команд, соответствующие варианту задания.

а. Способ представления данных в модели.

4. Содержание отчёта:

- 6.1 Формулировка варианта задания.
- 6.2 Граф-схема алгоритма решения задачи.
- 6.3 Распределение памяти (размещение в ОЗУ переменных, программы и констант).
- 6.4 Программа с описанием действий.

5. Контрольные вопросы:

- 5.1 Что такое система команд ЭВМ?
- 5.2 Способы адресации. Рассмотреть на примере.
- 5.3 Какие классы команд были использованы в задании?

6. Список литературы

Основная литература

1. Максимов Н. В. Архитектура ЭВМ и вычислительные системы: учебник / Н. В. Максимов, Т. Л. Партыка, И. И. Попов. - 4-е изд., перераб. и доп. - М.: ФОРУМ, 2012. - 512
2. Просветов Г.И. Дискретная математика: задачи и решения. – М.: Бином. Лаборатория знаний, 2010. – 222с.
3. Партыка Т. Л. Периферийные устройства вычислительной техники : учеб. пособие / Т. Л. Партыка, И. И. Попов. - 3-е изд., испр. и доп. - М. : ФОРУМ, 2012. - 432 с.
4. Юров В.И. Assembler. Учебник для вузов, 2-е издание. – СПб.: Питер, 2010. – 637с.

Дополнительная литература

1. Горнец Н.Н. Организация ЭВМ и систем. - М.:Академия,2008. - 320с.
2. Жмакин А.П. Архитектура ЭВМ. – СПб: БХВ-Петербург, 2006. – 320с.
3. Келим Ю.М. Вычислительная техника.- М.:Академия,2007. - 384с.
4. Вычислительная техника: метод. пособие /авт.-сост. Б.В. Цыбаков.- М., 2004. - 112с.

Лабораторная работа №11. Программирование типовых управляющих структур на языке Ассемблер

Время, отводимое на выполнение работы - 2 часа.

Перечень необходимых технических средств обучения:

- ✓ персональные компьютеры Core i3 3.0, 4 Gb, 4 ГБ ОЗУ, 250 Gb HDD;
- ✓ локальная сеть;
- ✓ коммутатор для подключения в сети Internet.

Перечень необходимых программных средств обучения:

- ✓ ОС Windows XP (7);
- ✓ Программа «Учебная ЭВМ».

1. **Цель занятия:** Приобрести опыт программирования учебной ЭВМ в машинных кодах.

2. **Основные теоретические положения:**

2.1 Типы адресации

Прямая, например, **add 23** – указание в команде непосредственно исполнительного адреса

Непосредственная, например, **add #33** – которая заключается в указании в команде самого значения операнда, а не его адреса

Косвенная, например, **add @33** – при которой в команде указывается адрес регистра или ячейки памяти, в которых хранится адрес операнда или его составляющие

2.2 Оператор условного перехода

Оператор ветвления (условная инструкция, условный оператор) — оператор, конструкция языка программирования, обеспечивающая выполнение определённой команды (набора команд) только при условии истинности некоторого логического выражения, либо выполнение одной из нескольких команд (наборов команд) в зависимости от значения некоторого выражения.

Оператор ветвления применяется в случаях, когда выполнение или невыполнение некоторого набора команд должно зависеть от выполнения или невыполнения некоторого условия. Ветвление — одна из трёх (наряду с последовательным исполнением команд и циклом) базовых конструкций структурного программирования.

Условный оператор реализует выполнение определённых команд при условии, что некоторое логическое выражение (условие) принимает значение «истина» true.

Условный оператор *в стандартной форме* состоит из условия (логического выражения) и двух последовательностей операторов.

Алгоритмы, в которых в зависимости от выполнения некоторого логического условия происходит разветвление вычислений по одному из нескольких возможных направлений, называют **разветвляющимися**. Подобные алгоритмы предусматривают выбор одного из альтернативных путей продолжения вычислений. Каждое возможное направление вычислений называется *ветвью*. Логическое условие называют *простым*, если разветвляющийся процесс имеет две ветви, и *сложным*, - если процесс разветвляется на три и более ветви.

Любое сложное логическое условие может быть представлено в виде простых. Рассмотрим пример разветвляющегося алгоритма с простым логическим условием.

Пример. Даны два числа a и b .

$$\text{Найти } x = \begin{cases} a/b, & \text{если } a > b \\ b/a, & \text{если } a \leq b \end{cases}$$

Очевидно, что для определения ветви, по которому необходимо производить процесс вычисления значения x , достаточно проверить выполнение одного из условий, например $a > b$. Если условие $a > b$ не выполняется, то очевидно и без дополнительной проверки, что будет выполнено условие $a < b$. Следовательно, вариант схемы алгоритма будет выглядеть следующим образом:

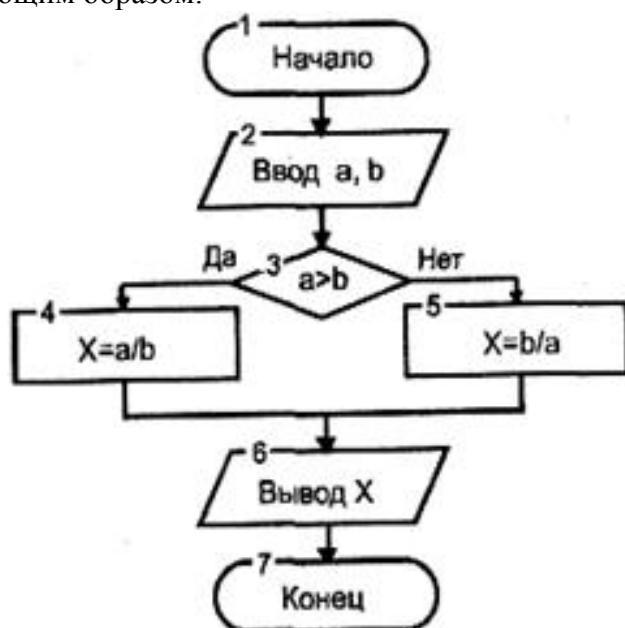


Рисунок 54 Схема простого разветвляющегося алгоритма

3. Задание к работе

3.1 Программирование разветвляющегося процесса. Разработать программу вычисления и вывода значения функции для вводимого из ИР значения аргумента x :

$$y = \begin{cases} f_i(x), & \text{при } x \geq a \\ f_j(x), & \text{при } x < a \end{cases}$$

Варианты заданий приведены ниже в таблице:

Таблица 60 Выбор значений параметров i, j, a

№	i	j	a	№	i	j	a	№	i	j	a	№	i	j	a	№	i	j	a
1	2	1	12	3	8	6	30	5	0	5	2	7	5	7	12	9	1	9	12
2	4	3	20	4	2	1	4	6	9	7	6	8	4	0	3	10	0	4	3

Выбор функции:

Таблица 31 Выбор функции

k	$f_k(x)$	k	$f_k(x)$	k	$f_k(x)$	k	$f_k(x)$	k	$f_k(x)$
0	$\frac{x+17}{10}$	2	$\frac{(x+3)^2}{10}$	4	$\frac{x^2+2x}{10}$	6	$(x+3)^3$	8	$\frac{x+3}{1-x}$
1	$\frac{(x+3)^2}{9}$	3	$\frac{(x+2)^2}{15}$	5	$\frac{81+x^2}{2}$	7	$\frac{(x+2)^2}{4}$	9	$\frac{(x+3)^2}{x}$

- Пример выполнения задания – программа для вычисления значения функции:

$$y = \begin{cases} (x - 11)^2 - 125, & \text{при } x \geq 16 \\ \frac{x^2 + 72x - 6400}{-168}, & \text{при } x < 16 \end{cases}$$
 . С устройства ввода IR вводится значение x, результат выводится на OR.

- Блок-схема выполнения задания на рисунке Рисунок 55. Блок-схема алгоритма задачи (см. ниже):

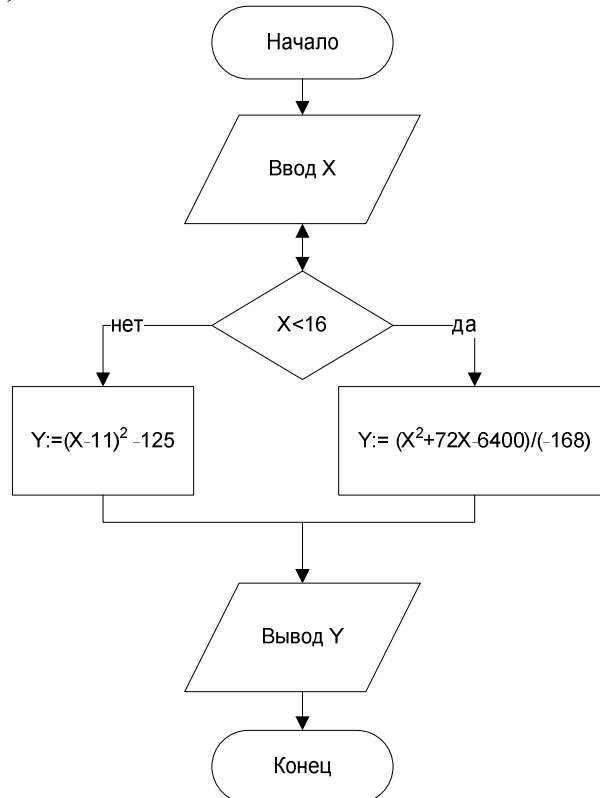


Рисунок 55. Блок-схема алгоритма задачи

- Реализация программы указана в таблице Таблица 317 Пример выполнения задания по блок-схеме

Таблица 32 Пример выполнения задания по блок-схеме

Мнемокод	Примечание
00) in	Ввод x
01) wr 30	Размещение x в ОЗУ
02) sub #16	Сравнение с границей – (x-16)
03) js m1	Переход по отрицательной разности
04) rd 30	Вычисления по первой формуле
05) sub #11	
06) wr 31	
07) mul 31	
08) sub #125	
09) jmp m1	Переход на вывод результата
010) m1: rd 30	Вычисления по второй формуле
011) mul 30	
012) wr 31	
013) rd 30	
014) mul #72	

015) add 31	
016) adi 106400	
017) divi 100168	
018) m2: out	Вывод результата
019) hlt	Стоп

4. Содержание отчёта:

4.1 Последовательность состояний регистров ЭВМ при выполнении программы в режиме Шаг для одного значения аргумента.

5. Контрольные вопросы:

5.1 Какие действия выполняют команды передачи управления?

5.2 Как организовать безусловный переход в программе?

5.3 Как поведёт себя программа из примера к третьему заданию, если в ней будет отсутствовать команда wt 31 по адресу 014?

6. Список литературы

Основная литература

1. Максимов Н. В. Архитектура ЭВМ и вычислительные системы: учебник / Н. В. Максимов, Т. Л. Партыка, И. И. Попов. - 4-е изд., перераб. и доп. - М.: ФОРУМ, 2012. - 512
2. Просветов Г.И. Дискретная математика: задачи и решения. – М.: Бином. Лаборатория знаний, 2010. – 222с.
3. Партыка Т. Л. Периферийные устройства вычислительной техники : учеб. пособие / Т. Л. Партыка, И. И. Попов. - 3-е изд., испр. и доп. - М. : ФОРУМ, 2012. - 432 с.
4. Юров В.И. Assembler. Учебник для вузов, 2-е издание. – СПб.: Питер, 2010. – 637с.

Дополнительная литература

1. Горнец Н.Н. Организация ЭВМ и систем. - М.:Академия,2008. - 320с.
2. Жмакин А.П. Архитектура ЭВМ. – СПб: БХВ-Петербург, 2006. – 320с.
3. Келим Ю.М. Вычислительная техника.- М.:Академия,2007. - 384с.
4. Вычислительная техника: метод. пособие /авт.-сост. Б.В. Цыбаков.- М., 2004. - 112с.

Лабораторная работа №12. Программирование цикла с переадресацией

Время, отводимое на выполнение работы - 2 часа.

Перечень необходимых технических средств обучения:

- ✓ персональные компьютеры Core i3 3.0, 4 Gb, 4 ГБ ОЗУ, 250 Gb HDD;
- ✓ локальная сеть;
- ✓ коммутатор для подключения в сети Internet.

Перечень необходимых программных средств обучения:

- ✓ ОС Windows XP (7);
- ✓ Программа «Учебная ЭВМ».

1. **Цель занятия:** на примере модели учебной ЭВМ написать программы, реализующие типовые управляющие структуры для работы с циклом с переадресацией.

2. Основные теоретические положения:

Для решения задачи с помощью ЭВМ должна быть разработана программа. Программа представляет собой последовательность команд. Код команды определяет выполняемую операцию, тип адресации и адрес.

Для реализации программ с применением цикла требуется разместить исходные данные в ячейки оперативной памяти подряд, т.е. в ячейки памяти с последовательными адресами.

Цикл - это многократно повторяющиеся действия с разными значениями исходных данных.

Составляющими элементами любого цикла являются:

Подготовка цикла - ПЦ,

Тело цикла - ТЦ,

Подготовка данных - ПД,

Проверка условия - ПУ.

Выполнение любого цикла начинается с ПЦ. Последовательность выполнения ТЦ, ПД, ПУ зависит от структуры цикла.

В связи с этим различают следующие типовые структуры цикла:

1. Цикл с предусловием - ЦИКЛ ПОКА (WHILE ... WEND).
2. Цикл с постусловием - ЦИКЛ ДО (DO ... LOOP).
3. Цикл с параметром - ЦИКЛ со СЧЕТЧИКОМ (FOR ... NEXT).

2.1 Цикл с предусловием - ЦИКЛ ПОКА

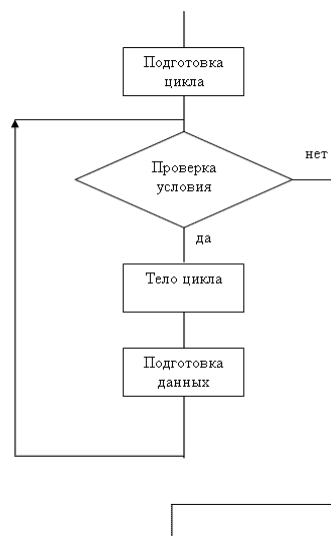


Рисунок 56 Цикл с предусловием / ПОКА

Перед выполнением операторов тела цикла осуществляется проверка условия на продолжение цикла. Если условие справедливо (ветвь «Да»), то цикл повторяется, иначе происходит выход из цикла.

Особенности данной структуры цикла:

- а) число повторений цикла заранее неизвестно;
- б) если при первой же проверке условия получается "Нет", то цикл не выполняется ни разу;
- в) возможен «бесконечный цикл», когда проверка условия не дает выхода на ветвь «Нет».

2.2 Цикл с постусловием - ДО

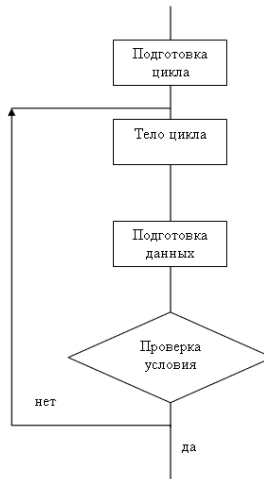


Рисунок 57 Цикл ДО

В блоке "Проверка условия" осуществляется проверка условия на прекращение цикла. Если условие справедливо (ветвь «Да»), то происходит выход из цикла, в противном случае цикл повторяется при новых значениях исходных данных. Особенности данной структуры цикла:

- а) число повторений цикла заранее неизвестно;
- б) так как условие проверяется в конце цикла, то тело цикла выполняется как минимум один раз;
- в) возможен «бесконечный цикл», когда проверка условия не дает выхода на ветвь «Да».

2.3 Цикл с параметром

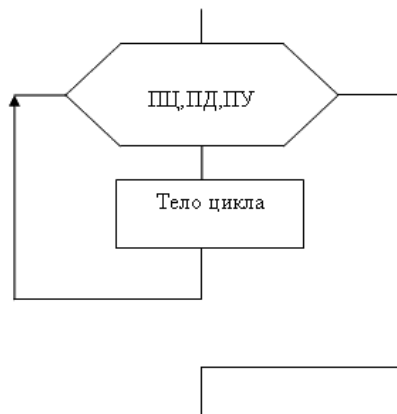


Рисунок 58 Цикл с параметром

Параметр цикла определяет число повторений цикла. Для параметра цикла указывается его начальное значение, конечное значение и шаг изменения. Тело цикла выполняется при каждом значении параметра цикла. Особенность данной структуры цикла

заключается в том, что уже перед началом выполнения цикла известно количество его повторений.

3. Задание к работе

Программирование цикла с переадресацией. Требуется разработать программу для определения заданной последовательности чисел $C_1, C_2, \dots, C_n$. Варианты заданий представлены ниже в таблице:

Таблица 33 Задание на тему Программирование цикла с переадресацией

Номер варианта	Характеристика последовательности чисел $C_1, C_2, \dots, C_n$
1	Количество чисел, имеющих чётный индекс
2	Номер минимального числа
3	Произведение всех чисел
4	Минимальное положительное число
5	Количество чисел, равных C_1
6	Количество отрицательных чисел
7	Максимальное положительное число
8	Номер максимального числа
9	Количество чисел, меньших C_1
10	Разность сумм элементов массива, имеющих чётные и нечётные индексы

Пример выполнения задания – программа вычисления суммы элементов массива чисел $C_1, C_2, \dots, C_n$. Исходными данными являются n – количество элементов массива, массив чисел $C_1, C_2, \dots, C_n$. Должно выполняться условие $n > 1$, так как алгоритм предусматривает хотя бы одно суммирование. Суммируемые числа записаны в ОЗУ подряд, т.е. в ячейки памяти с последовательными адресами. Результатом является сумма S .

Составим программу для вычисления суммы 10 чисел, элементы массива расположены в ячейках ОЗУ по адресам 040, 041, ..., 049. Программу распределим в памяти, начиная с адреса 000. Промежуточные переменные: A_i – в ячейке с адресом 030, k – по адресу 031, S – по адресу 032.

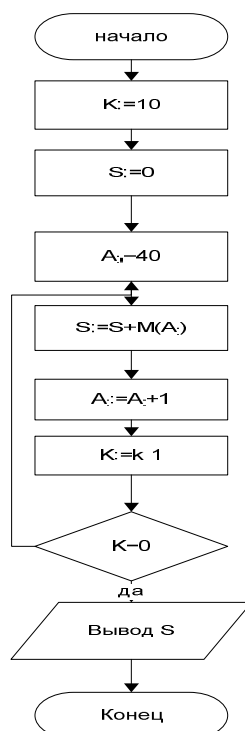


Рисунок 59 Алгоритм вычисления суммы элементов массива

Таблица 34 Реализация программы вычисления суммы элементов массива

Мнемокод	Примечание
00) rd #40	Загрузка начального адреса массива 040
01) wr 30	в ячейку 030
02) rd #10	Загрузка параметра цикла k=10 в ячейку 031
03) wr 31	
04) rd #0	Загрузка начального значения суммы S=0
05) wr 32	в ячейку 032
06) m1: rd 32	Добавление
07) add @30	к текущей суммы
08) wr 32	очередного элемента массива
09) rd 30	Модификация текущего
010) add #1	адреса массива
011) wr 30	(переход к следующему адресу)
012) rd 31	Уменьшение счётчика
013) sub #1	(параметра цикла)
014) wr 31	на 1
015) jnz m1	Проверка параметра цикла и переход при k≠0
016) rd 32	Вывод
017) out	результата
018) hlt	Стоп

4. Содержание отчёта:

- 4.1 Формулировка варианта задания
- 4.2 Граф-схема алгоритма решения задачи
- 4.3 Распределение памяти (размещение в ОЗУ переменных, программы и констант)
- 4.4 Программа с описанием действий
- 4.5 Последовательность состояний регистров ЭВМ при выполнении программы в режиме Шаг для одного значения аргумента (для задания 1).
- 4.6 Значения исходных данных и результата выполнения программы (для задания 1 – для нескольких значений аргументов, выбранных самостоятельно).

5. Контрольные вопросы:

- 5.1 Как организуется цикл?
- 5.2 Что такое параметр цикла?
- 5.3 Как поведёт себя программа из примера, если в ней метка m1 будет поставлена по адресу 005? 007?

6. Список литературы**Основная литература**

1. Юров В.И. Assembler. Учебник для вузов, 2-е издание. – СПб.: Питер, 2010. – 637с.
2. Горнец Н.Н. Организация ЭВМ и систем. - М.:Академия, 2008. - 320с.

Дополнительная литература

1. Жмакин А.П. Архитектура ЭВМ. – СПб: БХВ-Петербург, 2006. – 320с.
2. Келим Ю.М. Вычислительная техника. - М.:Академия, 2007. - 384с.
3. Вычислительная техника: метод. пособие /авт.-сост. Б.В. Цыбаков.- М., 2004. - 112с.

Лабораторная работа №13. Подпрограмма и стек

Время, отводимое на выполнение работы - 2 часа.

Перечень необходимых технических средств обучения:

- ✓ персональные компьютеры Core i3 3.0, 4 Gb, 4 ГБ ОЗУ, 250 Gb HDD;
- ✓ локальная сеть;
- ✓ коммутатор для подключения в сети Internet.

Перечень необходимых программных средств обучения:

- ✓ ОС Windows XP (7);
- ✓ Программа «Учебная ЭВМ».

1. **Цель занятия:** изучение организации программ с использованием подпрограмм.
2. **Основные теоретические положения:**

В программировании часто встречаются ситуации, когда одинаковые действия необходимо выполнять многократно в разных частях программы. В целях экономии памяти достаточно написать подпрограмму (процедуру) и обеспечить правильный вызов этой подпрограммы и возврат в точку вызова по завершению подпрограммы.

Для вызова подпрограммы необходимо указать её начальный адрес в памяти и передать процедуре, если это необходимо, те исходные данные, с которыми будут выполняться предусмотренные в подпрограмме действия. Адрес подпрограммы указывается в команде вызов CALL, а параметры могут передаваться через определённые ячейки памяти, регистры или стек.

Возврат в точку вызова обеспечивается сохранением адреса текущей команды (содержимого регистра PC) при вызове и использованием в конце подпрограммы команды возврата RET, которое возвращает сохранённое значение адреса возврата в PC.

Для реализации механизма вложенных подпрограмм (возможность вызова подпрограммы из другой подпрограммы) адреса возврата целесообразно сохранять в стеке. Стек («магазин») – особым образом организованная безадресная память, доступ к которой осуществляется через единственную ячейку, называемую верхушкой стека. При записи слово помещается в верхушку стека, предварительно все находящиеся в нем слова смещаются вниз на одну позицию, при чтении извлекаются содержимое верхушки стека (оно при этом из стека исчезает), а все оставшиеся слова перемещаются вверх на одну позицию. Такая дисциплина обслуживания называется LIFO (Last In First Out – последним пришёл – первым вышел), в отличие от дисциплины типа очередь FIFO (First In First Out – первым пришел – первым вышел).

В обычных ОЗУ нет возможности перемещать слова между ячейками, поэтому при организации стека перемещается не массив слов относительно неподвижной верхушки, а верхушка относительно неподвижного массива. Под стек отводится некоторая область ОЗУ, причём адрес верхушки хранится в специальном регистре процессора – указателе стека SP.

Таблица 35 Команды учебной ЭВМ

КОП	Мнемокод	Название
06	PUSH	Поместить содержимое регистра общего назначения в стек
07	POP	Извлечь в регистр общего назначения (R0...R9) из стека содержимое верхушки стека
19	CALL	Вызов подпрограммы, значение программного счетчика PC (адрес следующей команды) помещается в верхушку стека
08	RET	Возврат из подпрограммы в основную программу
17	JRNZ R,M	Цикл, содержимое указанного в команде регистра уменьшается на 1, и если в результате вычитания содержимое регистра равно 0, то управление передаётся на метку M. Эту команду следует ставить в конце цикла, а метку M – в первой команде цикла, а в регистр R помещать число повторений цикла.
30	MOV R3,R7	Пересылка данных с одного регистра в другой

В процессе организации циклов используются возможности систем команд ЭВМ, которые позволяют работать с новым классом памяти – сверхоперативной (регистры общего назначения РОН). В реальных ЭВМ доступ к РОН занимает значительно меньше времени, чем в ОЗУ. Команды обращения с регистрами короче команд обращения к памяти. Поэтому в РОН размещают часто используемые в программе данные, промежуточные результаты, счётчики циклов, косвенные адреса и т.п.

Кроме обычных способов адресации (прямой и косвенной) в регистровых командах используются два новых – постинкрементная (add @R3+) и прединкрементная (add -@R3).

3. Задание к работе

3.1 Даны три массива. Требуется вычислить их среднее арифметическое и их максимальных элементов. Каждый массив задаётся адресом первого элемента и их длиной.

3.1.1 Задайте первый массив в ячейках ОЗУ, начиная с адреса 085 и длиной 14 элементов, второй массив – 100 и 4, третий – 110 и 9.

3.1.2 Основная программа (самостоятельно заполните столбец действие)

Таблица 36. Основная подпрограмма

Программа	Действие
RD #85	
WR R1	
RD #14	
WR R2	
CALL M	
WR R6	
RD #100	
WR R1	
RD #4	
WR R2	
CALL M	
WR R7	
RD #110	
WR R1	
RD #9	
WR R2	
CALL M	
ADD R7	
ADD R6	
DIV #3	
OUT	

3.1.3 Подпрограмма нахождения максимального элемента массива (самостоятельно заполните столбец действие). Подпрограмма получает параметры через регистры: R1 – начальный адрес массива, R2 – длина массива. Регистры R6 и R7 используются для хранения максимальных элементов массива, R3 – для хранения текущего максимума, R4 – для временного хранения текущего элемента массива.

Таблица 37. 8 Реализация подпрограммы

Программа	Действие
HLT	
M: RD @R1	

Программа	Действие
WR R3 L2: RD @R1+ WR R4 SUB R3 JS L1 MOV R3,R4 L1: JRNZ R2,L2 RD R3 RET	

3.1.4 Нарисуйте алгоритм основной программы и подпрограммы

3.2 Составить и отладить программу учебной ЭВМ для решения следующей задачи:
Три массива заданы в память начальными адресами и длинами. Вычислить и вывести на устройство вывода среднее арифметическое параметров этих массивов:

- 3.2.1** Нарисуйте алгоритм основной программы
- 3.2.2** Нарисуйте алгоритм подпрограммы
- 3.2.3** Разместите в ОЗУ переменные, программы и константы
- 3.2.4** Напишите программу и подпрограмму для решения вашего варианта задания
- 3.2.5** Проверьте работу программы: вычислите результат работы для исходных данных и сравните с полученным вами результатом.

Таблица 38 Варианты задания по теме Подпрограммы

Вариант №	Параметры массива
1	Произведение всех элементов массива
2	Сумму элементов массива, равных С
3	Минимальное значение массива
4	Количество нечётных элементов массива
5	Количество чётных элементов массива
6	Номер максимального элемента массива
7	Номер минимального элемента массива
8	Количество элементов массива, не равных С
9	Сумма элементов массива
10	Количество элементов массива

4. Содержание отчёта:

- 4.1** Алгоритм основной программы и подпрограммы для реализации задания (таб.37).
- 4.2** Размещение исходных данных в ОЗУ.
- 4.3** Основная программа и подпрограмма для реализации задания (таб.37).

5. Контрольные вопросы:

- 5.1** Как работает команда MOV R3,R7?
- 5.2** Какие действия выполняет процессор при реализации команды CALL?
- 5.3** Дайте определение: что такое стек?
- 5.4** Расшифруйте LIFO (FIFO)?
- 5.5** Для каких целей используются регистры общего назначения?

6. Список литературы

Основная литература

1. Максимов Н. В. Архитектура ЭВМ и вычислительные системы: учебник / Н. В. Максимов, Т. Л. Партыка, И. И. Попов. - 4-е изд., перераб. и доп. - М.: ФОРУМ, 2012. - 512
2. Просветов Г.И. Дискретная математика: задачи и решения. – М.: Бином. Лаборатория знаний, 2010. – 222с.
3. Партыка Т. Л. Периферийные устройства вычислительной техники : учеб. пособие / Т. Л. Партыка, И. И. Попов. - 3-е изд., испр. и доп. - М. : ФОРУМ, 2012. - 432 с.
4. Юров В.И. Assembler. Учебник для вузов, 2-е издание. – СПб.: Питер, 2010. – 637с.

Дополнительная литература

1. Горнец Н.Н. Организация ЭВМ и систем. - М.: Академия, 2008. - 320с.
2. Жмакин А.П. Архитектура ЭВМ. – СПб: БХВ-Петербург, 2006. – 320с.
3. Келим Ю.М. Вычислительная техника.- М.: Академия, 2007. - 384с.
4. Вычислительная техника: метод. пособие /авт.-сост. Б.В. Цыбаков.- М., 2004. - 112с.

Лабораторная работа №14. Программирование внешних устройств.

Время, отводимое на выполнение работы - 2 часа.

Перечень необходимых технических средств обучения:

- ✓ персональные компьютеры Core i3 3.0, 4 Gb, 4 ГБ ОЗУ, 250 Gb HDD;
- ✓ локальная сеть;
- ✓ коммутатор для подключения в сети Internet.

Перечень необходимых программных средств обучения:

- ✓ ОС Windows XP (7);
- ✓ Программа «Учебная ЭВМ».

1. **Цель занятия:** изучение способов способами взаимодействия процессора и внешних устройств в составе ЭВМ, научиться подключать внешние устройства в составе учебной ЭВМ и составлять программы для вывода информации с клавиатуры на дисплей учебной ЭВМ

2. Основные теоретические положения:

2.1 Связь процессора и ВУ может осуществляться в синхронном и асинхронном режиме.

Синхронный режим используется для ВУ, всегда готовых к обмену. В нашей модели такими устройствами являются **дисплей** и **тоногенератор** – процессор может обращаться к этим ВУ, не анализируя их состояние (правда дисплей блокирует приём данных после ввода 128 символов, формируя флаг ошибки).

Асинхронный обмен предполагает анализ процессором состояния ВУ, которое определяет готовность ВУ выдать или принять данные или факт осуществления какого-нибудь события, контролируемого системой. К таким устройством в модели является **клавиатура** и **блок таймеров**.

2.2 Анализ состояния ВУ может осуществляться процессором двумя способами:

- в программно-управляемом режиме (предполагается обращение процессора к регистру состояния ВУ с последующим анализом значения соответствующего разряда слова состояния). Такое обращение следует предусмотреть в программе с некоторой периодичностью, независимо от фактического наступления контролируемого события (например, нажатия клавиши мыши)

- в режиме прерывания (при возникновении контролируемого события ВУ формирует процессору запрос на прерывание программы, по которому процессор и осуществляет связь в ВУ)

2.3 В состав контроллера клавиатуры входят три программно-доступных регистра:

- **DR** (адрес 0) — регистр данных;
- **CR** (адрес 1) — регистр управления, определяет режимы работы контроллера и содержит следующие флаги (устанавливаются и сбрасываются программно):
 - **E** — флаг разрешения приема кодов в буфер, при E = 0 контроллер игнорирует нажатие на клавиатуре, прием кодов в буфер не производится. На считывание кодов из буфера флаг E влияния не оказывает;
 - **I** — флаг разрешения прерывания, разрешает формирование запроса на прерывание от клавиатуры в момент установки флага готовности Rdy
 - **S** — флаг режима посимвольного ввода. При S = 0 флаг готовности Rdy формируется только после нажатия кнопки *Завершить ввод* в окне обозревателя клавиатуры, при S = 1 – после каждого нажатия клавиши;
- **SR** (адрес 2) — регистр состояния, содержит два флага, устанавливаемые контроллером «аппаратно»:
 - **Err** — флаг ошибки устанавливается при вводе в буфер 50-го символа, сбрасывается программно;
 - **Rdy** — флаг готовности, устанавливается в зависимости от значения флага S, сбрасывается «аппаратно» после выполнения команды ввода из регистра DR – считывание символа из буфера.

2.4 Дисплей представляет собой модель внешнего устройства, реализующую функции символьного дисплея. Дисплей может отображать символы, задаваемые ASCII-кодами, поступающими на его регистр данных.

Дисплей включает:

- видеопамять объемом 128 слов (ОЗУ дисплея);
- символьный экран размером 8 строк по 16 символов в строке;
- четыре программно-доступных регистра:
 - **DR** (адрес 0) — регистр данных;
 - **CR** (адрес 1) — регистр управления содержит следующие флаги (устанавливаются и сбрасываются программно):
 - **E** – флаг разрешения работы дисплея; при $E = 0$ запись в регистры AR и DR блокируется;
 - **A** – флаг автоинкремента адреса; при $A = 1$ содержимое AR автоматически увеличивается на 1 после любого обращения к регистру DR – по записи или чтению.
 - **SR** (адрес 2) – регистр состояния содержит единственный флаг *Err* – ошибки устанавливается аппаратно при попытке записать в регистр адреса число, большее 127, сбрасывается программно;
 - **AR** (адрес 3) – регистр адреса.

3. Задание к работе

3.1 Контроллер клавиатуры

3.1.1 Подключите внешнее устройство (меню Внешние устройства – Менеджер ВУ – Контроллер клавиатуры)

3.1.2 Для установки флага **E** в регистре **CR** требуется выполнить команды:

RD #10

OUT 1

3.1.3 Для установки **S** в «1» (режим посимвольного ввода) требуется выполнить следующие команды:

RD #103

OUT 1

3.1.4 Для заданного текста найдите ASCII-код каждого символа, согласно варианта задания из таблицы 13 Варианты текста:

Таблица 39 Варианты текста

№ варианта	
1	В лесу родилась ёлочка, в лесу она росла.
2	Зимой и летом стройная, зелёная была.
3	Метель ей пела песенку: «Спи, ёлочка, бай-бай!»
4	Мороз снежком укутывал: «Смотри, не замерзай»
5	Трусишка-зайка серенький, под ёлочкой трусил.
6	Порою волк, сердитый волк, рысцою пробегал.
7	Теперь она нарядная на праздник к нам пришла.
8	И много, много радости детишкам принесла.
9	Наша Таня громко плачет, уронила в речку мячик.
10	Тише, Танечка, не плач – не утонет в речке мяч.

3.2 Контроллер дисплея

3.2.1 Подключите внешнее устройство (меню Внешние устройства – Менеджер ВУ – Контролер дисплея)

3.2.2 Пример программы, выводящей слово «Май» приведён в таблице 40 Пример программы, выводящей слово «Май»

Таблица 40 Пример программы, выводящей слово «Май»

Метка	Команда	Примечание
	RD #11	; включаем дисплей и устанавливаем
	OUT 11	; флаг автоинкремента;
	RD #0	; задаём начальный адрес
	OUT 13	; выводимого слова;
	RD #204	; ввод кода буквы «М»
	OUT 10	; вывод на дисплей
	RD #224	; ввод кода буквы «а»
	OUT 10	; вывод на дисплей
	RD #233	; ввод кода буквы «й»
	OUT 10	; вывод на дисплей
	HLT	

3.2.3 Выведите на экран один текстовый фрагмент вашего варианта.

3.2.4 Выведите данный текст – каждое слово на новой строке, строка должна начинаться со своего номера.

3.3 Пример программы, выставляющей в начало каждой строки её номер приведён в таблице 41:

Таблица 41 Пример программы, выставляющей в начало каждой строки её номер

Метка	Команда	Примечание
	RD #10	; включаем
	OUT 11	; дисплей;
	RD #0	; задаём начальный
	WR R1	; адрес вывода;
	RD #49	; вводим код
	WR R2	; цифры «1»;
	RD #8	; вводим число
	WR R3	; повторений цикла;
M1:	RD R1	; читаем текущий адрес
	OUT 13	; и передаём в регистр адреса дисплея;
	ADD #16	; увеличиваем адрес на 16 – на начало следующей строки;
	WR R1	; сохраняем изменённый адрес;
	RD R2	; читаем код цифры – номер строки;
	OUT 10	; передаём код цифры на дисплей;
	RD @R2+	; увеличиваем содержимое R2 (код цифры) на единицу;
	JRNZ R3,M1	; декремент R3 и переход на начало цикла, если R3 ≠ 0;
	HLT	

3.4 Напишите программу для вывода на экран всех символов русского алфавита.

3.5 Выведите на экран один из трех текстовых сообщений, в зависимости от нажатой клавиши (1 – один текст, 2 – другой текст, 3 – третий текст). Текст выбрать по своему усмотрению.

3.6 Проанализируйте работу программы, дайте описание каждой команде в программе, приведённой в таблице 17:

Таблица 42 Примеры программ для работы с дисплеем

Вариант №	
1, 3, 5, 7, 9	RD #10 OUT 1 OUT 11 RD #103 OUT 1 M1:IN 2 JZ M1 IN 0 OUT 10 JMP M1
2, 4, 6, 8, 10	RD #10 OUT 11 RD #0 WR R1 RD #4 WR R2 M1:RD R1 OUT 13 ADD #16 WR R1 RD #42 OUT 10 JRNZ R2,M1 HLT

4. Содержание отчёта:

- 4.1 Формулировка варианта задания
- 4.2 Граф-схема алгоритма решения задачи
- 4.3 Распределение памяти (размещение в ОЗУ переменных, программы и констант)
- 4.4 Программа с описанием действий

5. Контрольные вопросы:

- 5.1 При каких условиях устанавливается и сбрасывается флаг готовности клавиатуры RD?
- 5.2 По какой команде происходит вывод на дисплей символа?

6. Список литературы

Основная литература

- 1. Максимов Н. В. Архитектура ЭВМ и вычислительные системы: учебник / Н. В. Максимов, Т. Л. Партыка, И. И. Попов. - 4-е изд., перераб. и доп. - М.: ФОРУМ, 2012. - 512
- 2. Просветов Г.И. Дискретная математика: задачи и решения. – М.: Бином. Лаборатория знаний, 2010. – 222с.
- 3. Партыка Т. Л. Периферийные устройства вычислительной техники : учеб. пособие / Т. Л. Партыка, И. И. Попов. - 3-е изд., испр. и доп. - М. : ФОРУМ, 2012. - 432 с.

4. Юров В.И. Assembler. Учебник для вузов, 2-е издание. – СПб.: Питер, 2010. – 637с.

Дополнительная литература

1. Горнец Н.Н. Организация ЭВМ и систем. - М.: Академия, 2008. - 320с.
2. Жмакин А.П. Архитектура ЭВМ. – СПб: БХВ-Петербург, 2006. – 320с.
3. Келим Ю.М. Вычислительная техника.- М.: Академия, 2007. - 384с.
4. Вычислительная техника: метод. пособие /авт.-сост. Б.В. Цыбаков.- М., 2004. - 112с.

Лабораторная работа №15. Подсистема прерывания

Время, отводимое на выполнение работы - 2 часа.

Перечень необходимых технических средств обучения:

- ✓ персональные компьютеры Core i3 3.0, 4 Gb, 4 ГБ ОЗУ, 250 Gb HDD;
- ✓ локальная сеть;
- ✓ коммутатор для подключения в сети Internet.

Перечень необходимых программных средств обучения:

- ✓ ОС Windows XP (7);
- ✓ Программа «Учебная ЭВМ».

1. Цель занятия: изучение механизма векторных прерываний.

2. Основные теоретические положения:

В модели предусмотрен **механизм векторных прерываний**. Источниками прерываний являются модели внешних устройств, в которых могут возникать события, требующие реакции программы.

Среди подключаемых ВУ формировать запросы на прерывания могут контроллер клавиатуры и блок таймеров, которым по умолчанию присваиваются вектора прерываний 0 и 2 соответственно. При этом вектор блока таймеров является общим для всех трёх таймеров, при необходимости обработчик прерывания может определить, какой из них вызвал прерывание по значению разрядов регистра SR состояния блока таймеров.

Таблица векторов прерываний располагается в ячейках ОЗУ 100 – 109, таким образом возможно обслуживать до 10 различных векторных источников прерываний. В ячейках таблицы должны располагаться адреса соответствующих программ – обработчиков прерываний. Эти адреса становятся известными после компиляции и заносятся в ячейки «вручную».

При вызове обработчика в стеке сохраняется адрес возврата (текущее значение PC) и значение регистр флагов IF, OV, S, Z. Возврат из обработчика прерывания осуществляется командой IRET, которая восстанавливает значения PC и регистра флагов.

Дисциплина обслуживания прерываний поддерживается с помощью программно-доступных флагов. Флаг IF разрешает/запрещает все прерывания, он устанавливается в «1» командой EI (Разрешить прерывания) и сбрасывается в «0» командой DI (Запретить прерывания). После RESET процессора IF = 0, поэтому, если предполагается использовать прерывания, следует выполнить команду EI. При вызове любого обработчика IF аппаратно сбрасывается, поэтому если требуется реагировать на другие запросы внутри обработчика прерывания, в его начале следует выполнить команду EI. Команда IRET восстанавливает все флаги, в том числе IF = 1.

В системе команд модели имеются команды INT0 – INT9, которые позволяют программно вызывать обработчики прерываний независимо от значения IF.

На ВУ предусмотрены т.н. «локальные маски», позволяющие запретить/разрешить прерывание от любого источника независимо от остальных.

3. Задание к работе

3.1 Программа для обработки прерывания от клавиатуры.

3.1.1 Подключите внешние устройства: клавиатуру и дисплей.

3.1.2 Адрес обработчика прерывания от клавиатуры (значение метки ОбрПр) необходимо записать в ячейку 100, так как вектор клавиатуры по умолчанию равен 0. Если компилировать приведённую программу с нулевого адреса подряд, то адрес команды IN 0 (начало обработчика) равен 8. Поэтому число 8 следует перед запуском программы записать в ячейку 100.

3.1.3 Напишите программу для обработки прерывания пустого цикла при вводе символа с клавиатуры и передаче этого символа на дисплей.

Таблица 43. Программа прерывания пустого цикла при вводе символа с клавиатуры и передаче этого символа на дисплей.

Метка	Команда	Примечание
	RD #11	; число 11 – в аккумулятор;
	OUT 1	; включить клавиатуру и разрешить прерывание от неё;
	OUT 11	; включить дисплей в режиме автоинкремента адреса;
	RD #103	; передаём в контроллер код команды
	OUT 1	; Установить S в «1» (режим посимвольного ввода);
	EI	; установить FI в «1» (разрешить прерывания);
M1:	NOP	; пустой
	JMP M1	; цикл;
ОбрПр:	IN 0	; считывание введённого символа из буфера в аккумулятор;
	OUT 10	; передача ASCII-кода на символьный дисплей
	IRET	; возврат из прерывания.

3.2 Программа для обработки прерывания от таймера:

3.2.1 Подключите внешнее устройство: таймер.

3.2.2 Таблица векторов прерываний занимает адреса 100-109. Соответственно, при поступлении запроса на прерывание N, от ВУ или программно, процессор берет байт из ячейки 100+N и делает переход, считая выбранный байт адресом обработчика нужного прерывания. Требуется создать в программе обработчик и занести его адрес в таблицу прерываний (т.е. по адресу N+100). Для обработчиков клавиатуры и таймера N будет равна 0 и 2, т.е. по адресу 100 прописываем адрес обработчика клавиатуры, а по адресу 102 прописываем адрес обработчика таймера. Не забудьте разрешить прерывания после записи в таблицу прерываний, командой EI.

3.2.3 Дайте описание каждой строчке программы. Что будет выведено на дисплей и с каким промежутком времени в результате выполнения программы:

RD #11	OUT 22
OUT 11	M2: IN 20
RD #50	JZ M2
WR R1	RDI 1000
RD #5	OUT 22
WR R2	RD #101
M1: RD @R1+	OUT 20
WR R3	JRNZ R2,M1
JNS L1	HLT
RD #45	
OUT 10	
RD R3	
MULI #100001	
ADD #48	
OUT 10	
JMP L2	
L1: RD R3	
ADD #48	
OUT 10	
L2: RD #625	
OUT 21	
RDI 1101	

3.2.4 Самостоятельно, напишите программу для реализации алгоритма, с применением прерывания:

Таблица 44. Задание на тему Подсистема прерываний

Вариант №	
1	Напишите программу для ввода двух слов из памяти с выводом на дисплей, предусмотрите задержку вывода 10 секунд
2	Напишите программу для вывода трёх слов, хранящихся в памяти, с задержкой вывода второго слова, равной 10 секунд, третьего слова – 20 секунд
3	Вывод содержимого участка памяти на экран посимвольно с заданным промежутком времени между символами
4	Напишите программу для ввода символов с клавиатуры с выводом на дисплей, предусмотрите очистку дисплея каждые 10 секунд
5	Напишите программу для ввода слов, вводимых с клавиатуры в память, начиная с ячейки памяти с номером 30
6	Напишите программу для вывода на экран цифр от 1 до 10, причём вывод каждой цифры сопровождается звуковым сигналом
7	Напишите программу для вывода на экран вводимых символов, причём вывод каждого символа сопровождается троекратным звуковым сигналом
8	Напишите программу для вывода на экран вводимых символов, причём вывод каждого символа сопровождается звуковым сигналом
9	Напишите программу для ввода символов с клавиатуры с выводом на дисплей, предусмотрите очистку дисплея через каждые 35 символов
10	Вывод на дисплей содержимого группы ячеек памяти в числовой форме, при переходе к выводу содержимого очередной ячейки формируется задержка – 10 секунд

4. Содержание отчёта:

- 4.1 Выполненные программы (пп.3.1, 3.2.3)
- 4.2 Формулировка варианта задания (п.3.2.4)
- 4.3 Программа с описанием действий (п.3.2.4)

5. Контрольные вопросы:

- 5.1 При каких условиях устанавливается и сбрасывается флаг готовности клавиатуры Rd?
- 5.2 В какой области памяти модели ЭВМ могут располагаться программы – обработчик прерываний?

6. Список литературы

Основная литература

- 1. Максимов Н. В. Архитектура ЭВМ и вычислительные системы: учебник / Н. В. Максимов, Т. Л. Партыка, И. И. Попов. - 4-е изд., перераб. и доп. - М.: ФОРУМ, 2012. - 512
- 2. Просветов Г.И. Дискретная математика: задачи и решения. – М.: Бином. Лаборатория знаний, 2010. – 222с.
- 3. Партыка Т. Л. Периферийные устройства вычислительной техники : учеб. пособие / Т. Л. Партыка, И. И. Попов. - 3-е изд., испр. и доп. - М. : ФОРУМ, 2012. - 432 с.
- 4. Юров В.И. Assembler. Учебник для вузов, 2-е издание. – СПб.: Питер, 2010. – 637с.

Дополнительная литература

- 1. Горнец Н.Н. Организация ЭВМ и систем. - М.:Академия,2008. - 320с.
- 2. Жмакин А.П. Архитектура ЭВМ. – СПб: БХВ-Петербург, 2006. – 320с.
- 3. Келим Ю.М. Вычислительная техника.- М.:Академия,2007. - 384с.
- 4. Вычислительная техника: метод. пособие /авт.-сост. Б.В. Цыбаков.- М., 2004. - 112с.

Информационное обеспечение обучения

Основные источники:

5. Даниловских М. Г. Периферия персональных компьютеров : учеб. пособие : в 2 ч. Ч. 1 / М. Г. Даниловских, Л. И. Винник, Н. В. Курмышев ; Новгород. гос. ун-т им. Ярослава Мудрого. - Великий Новгород, 2015. - 302, [1] с.
6. Даниловских М. Г. Периферия персональных компьютеров: учеб. пособие : в 2 ч. Ч. 2 / М. Г. Даниловских, Л. И. Винник, Н. В. Курмышев ; Новгород. гос. ун-т им. Ярослава Мудрого. - Великий Новгород, 2016.
7. Гвоздева, В. А. Информатика, автоматизированные информационные технологии и системы: учебник / В. А. Гвоздева. - М. : Форум ; Инфра-М, 2014. - 544 с.

Дополнительные источники:

1. Максимов Н. В. Архитектура ЭВМ и вычислительные системы: учебник / Н. В. Максимов, Т. Л. Партыка, И. И. Попов. - 4-е изд., перераб. и доп. - М.: ФОРУМ, 2012. - 512
2. Просветов Г.И. Дискретная математика: задачи и решения. – М.: Бином. Лаборатория знаний, 2010. – 222с.
3. Партыка Т. Л. Периферийные устройства вычислительной техники : учеб. пособие / Т. Л. Партыка, И. И. Попов. - 3-е изд., испр. и доп. - М. : ФОРУМ, 2012. - 432 с.
4. Юров В.И. Assembler. Учебник для вузов, 2-е издание. – СПб.: Питер, 2010. – 637с.
5. Браммер Ю. А. Импульсная техника.- М.:ФОРУМ:ИНФРА-М,2005.- 207с.
6. Бойт К. Цифровая электроника. Москва: Техносфера, 2007. - 422с.
7. Вычислительная техника:метод.пособие /авт.-сост.Б.В.Цыбаков.- М.,2004.- 112с
8. Горнец Н.Н.Организация ЭВМ и систем. - М.: Академия, 2008. - 320с.
9. Жмакин А.П. Архитектура ЭВМ. – СПб: БХВ-Петербург, 2006. – 320с.
10. Калиш Г.Г. Основы вычислительной техники.- М.:Выш.шк.,2000. - 257с.
11. Келим Ю.М. Вычислительная техника. - М.: Академия, 2007. - 384с.
12. Нефёдова Н.В. Карманный справочник по электронике и электротехнике. – Изд. 2-е. – Ростов на Дону: Феникс, 2007. – 283с.
13. Мышляева И.М.Цифровая схемотехника.- М.: Академия, 2005. - 400с.
14. Преснухин Л. Н. Расчет элементов цифровых устройств.- М.:Выш.шк.,1991.- 526с.
15. Управляющие вычислительные комплексы /ред.Н.Л.Прохоров.- М.: Финансы и статистика, 2003. - 352с.

Сайты дистанционного обучения:

1. <http://dvo.sut.ru/libr>
2. <http://sdo.uspi.ru/mathem&inform/>
3. <http://it.fitib.altstu.ru/neud/shemotechnika>
4. <http://sasoft.qrz.ru/indexr.htm>
5. <http://www.intuit.ru>
6. <http://www.automationlab.ru/>
7. <http://www.siblec.ru/>

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Номер изме- нения	Номер листа				Всего листов в документе	ФИО и подпись ответственного за внесение изменения	Дата внесения изменения	Дата введения изменения
	измененного	замененного	нового	изъятого				