

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Матвеев Александр Сергеевич
Должность: И.о. начальника учебно-методического управления
Дата подписания: 13.12.2023 15:58:15
Уникальный программный ключ:
49d49750726076f86fca1f5d4926362e70745ca

Приложение к ППССЗ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Российский государственный аграрный университет –
МСХА имени К.А. Тимирязева»
(ФГБОУ ВО РГАУ-МСХА имени К.А. Тимирязева)
ТЕХНОЛОГИЧЕСКИЙ КОЛЛЕДЖ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ

по дисциплине
ОП.07 «Основы вычислительной техники»

**специальность: 15.02.10 Мехатроника и
мобильная робототехника (по отраслям)**

форма обучения: очная

Пояснительная записка

Методические указания по выполнению практических работ подготовлены на основе рабочей программы учебной дисциплины «Основы вычислительной техники», разработанной на основе ФГОС СПО по специальности 15.02.10 «Мехатроника и мобильная робототехника (по отраслям)» и соответствующих профессиональных (ПК) компетенций:

ПК 1.2. Осуществлять настройку и конфигурирование программируемых логических контроллеров и микропроцессорных систем в соответствии с принципиальными схемами подключения.

ПК 1.3. Разрабатывать управляющие программы мехатронных систем в соответствии с техническим заданием.

ПК 3.1. Составлять схемы простых мехатронных систем в соответствии с техническим заданием.

ПК 3.2. Моделировать работу простых мехатронных систем.

ПК 4.1. Осуществлять настройку и конфигурирование управляющих контроллеров мобильных робототехнических комплексов в соответствии с принципиальными схемами подключения.

ПК 4.2. Разрабатывать управляющие программы мобильных робототехнических комплексов в соответствии с техническим заданием.

ПК 4.3. Осуществлять настройку датчиков и исполнительных устройств мобильных робототехнических комплексов в соответствии с управляющей программой и техническим заданием.

ПК 5.4. Диагностировать неисправности мобильных робототехнических комплексов с использованием алгоритмов поиска и устранения неисправностей.

Целью освоения учебной дисциплины «Основы вычислительной техники» является:

- теоретическая и практическая подготовка студентов в области информационных технологий в такой степени, чтобы они могли выбирать необходимые технические, алгоритмические, программные и технологические решения, уметь объяснить принципы их функционирования и правильно их использовать;
- формирование у студентов знаний по дисциплине, достаточных для самостоятельного освоения вычислительных систем с новыми архитектурами.

При выполнении практических работ студент должен **уметь**:

- настраивать и конфигурировать ПЛК в соответствии с принципиальными схемами подключения;
- программировать ПЛК с целью анализа и обработки цифровых и аналоговых сигналов и управления исполнительными механизмами мехатронных систем;

- применять специализированное программное обеспечение при разработке управляющих программ и визуализации процессов управления и работы мехатронных систем;
- проводить расчеты параметров типовых электрических, пневматических и гидравлических схем узлов и устройств, разрабатывать несложные мехатронные системы;
- составлять структурные, функциональные и принципиальные схемы мехатронных систем;
- применять специализированное программное обеспечение при моделировании мехатронных систем;
- использовать стандартные пакеты (библиотеки) языка для решения практических задач;
- решать исследовательские и проектные задачи с использованием компьютеров;
- решать конфигурационные задачи с использованием компьютеров при построении системы управления мобильным роботом;
- понимание систем программирования и управления мобильными роботами;
- понимание технологии построения беспроводной сети и взаимосвязи робота и компьютера, используя данную технологию;
- использование поставляемого производителем программного обеспечения для анализа передаваемых датчиками данных, и обеспечение диагностики роботом на основе данных, поступающих с датчиков.

При выполнении практических работ студент должен **знать:**

- принципы связи программного кода, управляющего работой ПЛК, с действиями исполнительных механизмов;
- методы непосредственного, последовательного и параллельного программирования;
- алгоритмы поиска ошибок управляющих программ ПЛК;
- промышленные протоколы для объединения ПЛК в сеть;
- языки программирования и интерфейсы ПЛК;
- технологии разработки алгоритмов управляющих программ ПЛК;
- типовые модели мехатронных систем;
- основные факты, базовые концепции и модели информатики; основы технологии работы на ПК в современных операционных средах;
- технологию работы на ПК в современных операционных средах, основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов, типовые алгоритмы обработки данных; основные принципы и методологию разработки прикладного программного обеспечения, включая типовые способы организации данных и построения алгоритмов обработки данных, синтаксис и семантику универсального алгоритмического языка программирования высокого уровня;

- современных основ информационно-коммуникационных технологий для решения некоторых типовых задач в проектировании мобильных роботов;
- методов построения современных мобильных роботов.

Содержание практических занятий определено рабочей программой и тематическим планированием, соответствует теоретическому материалу изучаемых разделов учебной дисциплины.

Объём практических занятий по дисциплине определяется учебным планом по данной специальности.

Продолжительность практического занятия - 2 академических часа. Перед проведением практического занятия преподавателем организуется инструктаж, а по ее окончании – обсуждение итогов.

Комплект методических указаний по выполнению практических работ дисциплины «Основы вычислительной техники» содержит 15 практических занятий.

**Перечень практических работ
по дисциплине «Основы вычислительной техники»**

ПРАКТИЧЕСКАЯ РАБОТА № 1.

Тема: «Виды информации и способы представления её в ЭВМ».

ПРАКТИЧЕСКАЯ РАБОТА № 2.

Тема: «Виды информации и способы представления её в ЭВМ».

ПРАКТИЧЕСКАЯ РАБОТА № 3.

Тема: «Виды информации и способы представления её в ЭВМ».

ПРАКТИЧЕСКАЯ РАБОТА № 4.

Тема: «Виды информации и способы представления её в ЭВМ».

ПРАКТИЧЕСКАЯ РАБОТА № 5.

Тема: «Логические элементы электронно-вычислительной техники (ЭВТ)».

ПРАКТИЧЕСКАЯ РАБОТА № 6.

Тема: «Последовательные цифровые устройства».

ПРАКТИЧЕСКАЯ РАБОТА № 7.

Тема: «Последовательные цифровые устройства».

ПРАКТИЧЕСКАЯ РАБОТА № 8.

Тема: «Последовательные цифровые устройства».

ПРАКТИЧЕСКАЯ РАБОТА № 9.

Тема: «Основные типы микропроцессоров, структуры команд, структура устройства управления».

ПРАКТИЧЕСКАЯ РАБОТА № 10.

Тема: «Основные типы микропроцессоров, структуры команд, структура устройства управления».

ПРАКТИЧЕСКАЯ РАБОТА № 11.

Тема: «Методы цифровой обработки сигналов».

ПРАКТИЧЕСКАЯ РАБОТА № 12.

Тема: «Методы цифровой обработки сигналов».

ПРАКТИЧЕСКАЯ РАБОТА № 13.

Тема: «Программное обеспечение в сфере профессиональной деятельности».

ПРАКТИЧЕСКАЯ РАБОТА № 14.

Тема: «Программное обеспечение в сфере профессиональной деятельности».

ПРАКТИЧЕСКАЯ РАБОТА № 15.

Тема: «Программное обеспечение в сфере профессиональной деятельности».

ИНСТРУКЦИИ ДЛЯ ОБУЧАЮЩИХСЯ ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ

Прежде чем приступить к выполнению заданий, внимательно прочитайте данные рекомендации. Практические работы включают в себя задания следующих видов:

Правила выполнения практических работ за компьютером.

1. Студент должен выполнить практическую работу самостоятельно (или в группе, если это предусмотрено заданием).
2. Если студент не выполнил практическую работу или часть работы за отведенное время, то он может выполнить работу или оставшуюся часть во внеурочное время, согласованное с преподавателем.
3. Каждый студент после окончания урока, должен представить преподавателю выполненную работу в электронном виде с анализом полученных результатов и выводом по работе.
4. Дифференцированную оценку по практической работе студент получает, с учетом срока выполнения работы, если:
 - работа выполнена правильно и в полном объеме;
 - сделан анализ проделанной работы и вывод по результатам работы;
 - студент может пояснить выполнение любого этапа работы.

Зачет по практическим работам студент получает при условии выполнения всех предусмотренной программой работ, после сдачи отчетов по работам при удовлетворительных оценках за опросы и контрольные вопросы во время практических занятий.

Инструкция по технике безопасности при выполнении практических работ студентами.

Запрещается:

1. Трогать разъёмы соединительных кабелей.
2. Прислоняться к экрану и тыльной стороне монитора.
3. Включать и выключать ЭВМ без разрешения преподавателя.
4. Прислоняться к проводам и устройствам заземления.
5. При обнаружении запаха гари немедленно остановить работу, выключить клавиатуру и сообщить преподавателю.

Перед началом работы:

1. Убедиться в отсутствии видимых повреждений рабочего места.
2. Запрещается работать во влажной одежде (и вообще в верхней одежде) и с влажными руками.
3. На рабочем месте размещать тетрадь и учебные пособия так, чтобы они не мешали работе.

Во время работы:

1. Работать 60-80 см на расстоянии от ЭВМ.
2. Строго выполняйте вышеуказанные правила.
3. Следить за исправностью аппаратуры.
4. Немедленно прекратить работу при появлении постороннего звука и сообщить преподавателю.
5. Работать на клавиатуре чистыми руками, правильно нажимать на клавиши.
6. Никогда не пытаться самим устранить неисправность при работе с аппаратурой.
7. Не вставать со своих мест, когда входит посетитель.

По окончании работы:

1. Отключить ЭВМ, навести порядок на рабочем месте.
2. Сдать рабочее место преподавателю.

Выполнение практических работ.

Студент должен:

- выполнять требования по охране труда;
- соблюдать инструкцию по правилам и мерам безопасности в учебном кабинете.
- строго выполнять весь объем работы, указанный в задании;
- соблюдать требования эксплуатации компьютерной техники (правила включения и выключения).
- предоставить отчет о проделанной работе по окончании выполненной работы, который должен содержать: название работы, цель работы, задание и его решение, вывод о проделанной работе.

Отчет о проделанной работе может быть выполнен на компьютере или в тетрадях для практических работ.

Требования к отчету по практическим работам, выполненным на компьютере.

Текст отчета по практической работе должен быть набран на компьютере шрифтом TimesNewRoman размером 14 пт. (при оформлении текста используется текстовый редактор MS Word). Шрифт, используемый в иллюстративном материале (таблицы и рисунки), рекомендуется уменьшить до 12 пт. Межстрочный интервал в основном тексте - полуторный. В иллюстративном материале межстрочный интервал рекомендуется сделать одинарным. Поля страницы должны быть: левое поле - 30 мм; правое поле – 1,5 мм; верхнее и нижнее поле - 20 мм.

Каждый абзац должен начинаться с красной строки. Отступ абзаца – 1,25 мм от левой границы текста.

Требования к отчету по практическим работам, выполненным в тетради.

1. В тетради для выполнения отчета по практическим занятиям пишется: «Практическое занятие №...»

2. Под надписью «Практическое занятие №...» укажите тему.
 3. Ниже напишите: «Цель занятия».
 4. Под надписью «Цель занятия» в центре укажите: «Вариант №...».
- Поставьте номер своего варианта.
5. Оформите порядок выполнения практической части занятия, опираясь задание.
 6. Напишите вывод по занятию.

Студент должен выполнить практическую работу самостоятельно (или в группе, если это предусмотрено заданием). Практическая работа выполняется согласно заданию и методическим рекомендациям. После выполнения практической работы обучающийся самостоятельно себя контролирует путем ответов на вопросы. Результат работы представляется преподавателю в виде файла (файлов) в личном каталоге, защищается обучающимися.

По ходу выполнения работы при возникновении вопросов обучающийся может получить консультацию у преподавателя или самостоятельно воспользоваться лекционным материалом, рекомендуемой литературой.

ПРАКТИЧЕСКАЯ РАБОТА № 1

Тема: «Виды информации и способы представления её в ЭВМ»

Цель: научиться выполнять перевод чисел в разные системы счисления.

Оборудование: справочные пособия.

Справочный материал

1. Основные понятия систем счисления

Система счисления — это система записи чисел с помощью определенного набора цифр. Количество цифр, необходимых для записи числа в системе, называют основанием системы счисления. Основание системы записывается в справа числа в нижнем индексе: $AF17816$.

Различают два типа систем счисления: 5_{10} ; 1110110_2 ;

– позиционные, когда значение каждой цифры числа определяется ее позицией в записи числа;

– непозиционные, когда значение цифры в числе не зависит от ее места в записи числа. Примером непозиционной системы счисления является римская: числа IX, IV, XV и т.д.

Примером позиционной системы счисления является десятичная система, используемая повседневно.

Десятичная система счисления – в настоящее время наиболее известная и используемая. Десятичная система использует десять цифр — 0, 1, 2, 3, 4, 5, 6,

7, 8 и 9, а также символы “+” и “-” для обозначения знака числа и запятую или точку для разделения целой и дробной частей числа.

В вычислительных машинах используется двоичная система счисления, её основание — число 2. Для записи чисел в этой системе используют только две цифры — 0 и 1.

Таблица 1. Соответствие чисел, записанных в различных системах счисления

Десятичная	Двоичная	Восьмеричная	Шестнадцатеричная
1	001	1	1
2	010	2	2
3	011	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

2. **Правила перевода чисел из одной системы счисления в другую**

Перевод чисел из одной системы счисления в другую составляет важную часть машинной арифметики. Рассмотрим основные правила перевода.

1. Для перевода двоичного числа в десятичное необходимо его записать в виде многочлена, состоящего из произведений цифр числа и соответствующей степени числа 2, и вычислить по правилам десятичной арифметики.

При переводе удобно пользоваться таблицей степеней двойки:

Таблица 2. Степени числа 2

n	0	1	2	3	4	5	6	7	8	9	10
2^n	1	2	4	8	16	32	64	128	256	512	1024

2. Для перевода восьмеричного числа в десятичное необходимо его записать в виде многочлена, состоящего из произведений цифр числа и соответствующей степени числа 8, и вычислить по правилам десятичной арифметики.

При переводе удобно пользоваться таблицей степеней восьмерки:

Таблица 3. Степени числа 8

n	0	1	2	3	4	5	6
-----	---	---	---	---	---	---	---

8^n	1	8	64	512	4096	32768	262144
-------	---	---	----	-----	------	-------	--------

Пример. Число 750138 перевести в десятичную систему счисления.

$$75013_8 = 7 \cdot 8^4 + 5 \cdot 8^3 + 0 \cdot 8^2 + 1 \cdot 8^1 + 3 \cdot 8^0 = 31243_{10}$$

3. Для перевода шестнадцатеричного числа в десятичное необходимо его записать в виде многочлена, состоящего из произведений цифр числа и соответствующей степени числа 16, и вычислить по правилам десятичной арифметики.

При переводе удобно пользоваться таблицей степеней числа 16:

Таблица 4. Степени числа 16

n	0	1	2	3	4	5	6
16^n	1	16	256	4096	65536	1048576	16777216

4. Для перевода десятичного числа в двоичную систему его необходимо последовательно делить на 2 до тех пор, пока не останется остаток, меньший или равный 1. Число в двоичной системе записывается как последовательность последнего результата деления и остатков от деления в обратном порядке.

Пример. Число 2210 перевести в двоичную систему счисления.

$$\begin{array}{r}
 22 \overline{) 22} \quad 2 \\
 \underline{22} \quad 11 \quad 2 \\
 \quad 10 \quad 5 \quad 2 \\
 \quad \quad 4 \quad 2 \quad 2 \\
 \quad \quad \quad 1 \quad 2 \quad 2 \\
 \quad \quad \quad \quad 0 \quad 1 \quad 1 \\
 \quad \quad \quad \quad \quad 0 \quad 1
 \end{array}$$

$$2210 = 101102$$

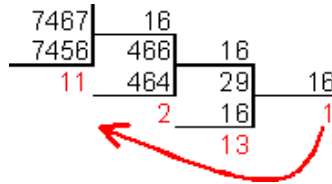
Для перевода десятичного числа в восьмеричную систему его необходимо последовательно делить на 8 до тех пор, пока не останется остаток, меньший или равный 7. Число в восьмеричной системе записывается как последовательность цифр последнего результата деления и остатков от деления в обратном порядке.

Пример. Число 57110 перевести в восьмеричную систему счисления.

$$\begin{array}{r}
 571 \overline{) 8} \\
 \underline{56} \quad 71 \quad 8 \\
 \quad 11 \quad 64 \quad 8 \\
 \quad \quad 8 \quad 7 \quad 8 \\
 \quad \quad \quad 3 \quad 7 \quad 0 \quad 1
 \end{array}$$

$$571_{10} = 1073_8$$

1. Для перевода десятичного числа в шестнадцатеричную систему его необходимо последовательно делить на 16 до тех пор, пока не останется остаток, меньший или равный 15. Число в шестнадцатеричной системе записывается как последовательность цифр последнего результата деления и остатков от деления в обратном порядке.



Пример. Число 74672 перевести в шестнадцатеричную систему счисления.

$$7467_{10} = 1D2B_{16}$$

2. Чтобы перевести число из двоичной системы в восьмеричную, его нужно разбить на триады (тройки цифр), начиная с младшего разряда, в случае необходимости дополнив старшую триаду нулями, и каждую триаду заменить соответствующей восьмеричной цифрой (табл. 3).

Пример. Число 10010112 перевести в восьмеричную систему счисления.

$$001\ 001\ 011_2 = 113_8$$

3. Чтобы перевести число из двоичной системы в шестнадцатеричную, его нужно разбить на тетрады (четверки цифр), начиная с младшего разряда, в случае необходимости дополнив старшую тетраду нулями, и каждую тетраду заменить соответствующей шестнадцатеричной цифрой (табл. 3).

Пример. Число 1011100012 перевести в шестнадцатеричную систему счисления.

$$0010\ 1110\ 0011_2 = 2E3_{16}$$

4. Для перевода восьмеричного числа в двоичное необходимо каждую цифру заменить эквивалентной ей двоичной триадой.

Пример. Число 5318 перевести в двоичную систему счисления.

$$531_8 = 101011001_2$$

5. Для перевода шестнадцатеричного числа в двоичное необходимо каждую цифру заменить эквивалентной ей двоичной тетрадой.

Пример. Число EE816 перевести в двоичную систему счисления.

$$EE8_{16} = 111011101000_2$$

6. При переходе из восьмеричной системы счисления в шестнадцатеричную и обратно, необходим промежуточный перевод чисел в двоичную систему.

Пример 1. Число FEA16 перевести в восьмеричную систему счисления.

$$FEA_{16} = 111111101010_2$$

$$111\ 111\ 101\ 010_2 = 7752_8$$

Пример 2. Число 66538 перевести в шестнадцатеричную систему счисления.

$$6653_8 = 110110101011_2$$

$$1101\ 1010\ 1011_2 = DAB_{16}$$

3. **Арифметические действия над целыми числами в 2-ой системе счисления:**

3 Операция сложения выполняется с использованием таблицы двоичного сложения в одном разряде:

Пример.

$$\begin{array}{r} \text{а) } \square 1001_2 \\ 1010_2 \\ 10011_2 \end{array} \quad \begin{array}{r} \text{б) } \square 1101_2 \text{ в) } \square 11111_2 \\ 1011_2 \\ 11000_2 \\ 1 \\ 100000_2 \end{array}$$

- 4 Операция вычитания выполняется с использованием таблицы вычитания, в которой 1 обозначается заем в старшем разряде.

Пример.

$$\begin{array}{r} \text{а) } -101110011_2 \\ 100011011_2 \\ 001011000_2 \end{array} \quad \begin{array}{r} \text{б) } 110101101_2 \\ 101011111_2 \\ 001001110_2 \end{array}$$

- 5 Операция умножения выполняется по обычной схеме, применяемой в десятичной с/с с последовательным умножением множимого на очередную цифру множителя.

Пример.

$$\begin{array}{r} \text{а) } \square 11001_2 \\ 1101_2 \\ 11001 \\ 11001 \\ 11001 \\ 101000101_2 \end{array} \quad \begin{array}{r} \text{б) } \square 101_2 \\ 11_2 \\ 101 \\ 101 \\ 1111_2 \end{array}$$

- 6 Операция деления выполняется по алгоритму, подобному алгоритму выполнения операции деления в 10-ой с/с.

Сложение и вычитание в восьмеричной системе счисления.

При выполнении сложения и вычитания в 8-ой с/с необходимо соблюдать следующие правила:

- 1) в записи результатов сложения и вычитания могут быть использованы только цифры восьмеричного алфавита;
- 2) десяток восьмеричной системы счисления равен 8, т.е. переполнение разряда наступает, когда результат сложения больше или равен 8. В этом случае для записи результата надо вычесть 8, записать остаток, а к старшему разряду прибавить единицу переполнения;
- 3) если при вычитании приходится занимать единицу в старшем разряде, эта единица переносится в младший разряд в виде восьми единиц.

Пример

$$\begin{array}{r} + 770_8 \\ 236_8 \\ 1226_8 \end{array} \quad \begin{array}{r} + 750_8 \\ 236_8 \\ 512_8 \end{array}$$

Сложение и вычитание в шестнадцатеричной системе счисления.

При выполнении этих действий в 16–ой с/с необходимо соблюдать следующие правила:

- 1) при записи результатов сложения и вычитания надо использовать цифры шестнадцатеричного алфавита: цифры, обозначающие числа от 10 до 15 записываются латинскими буквами, поэтому, если результат является числом из этого промежутка, его надо записывать соответствующей латинской буквой;
- 2) десяток шестнадцатеричной системы счисления равен 16, т.е. переполнение разряда поступает, если результат сложения больше или равен 16, и в этом случае для записи результата надо вычесть 16, записать остаток, а к старшему разряду прибавить единицу переполнения;
- 3) если приходится занимать единицу в старшем разряде, эта единица переносится в младший разряд в виде шестнадцати единиц.

Пример

$$\begin{array}{r} + \text{B09}_{16} \\ \text{TFA}_{16} \\ 1\text{A03}_{16} \end{array} \quad \begin{array}{r} + \text{B09}_{16} \\ \text{7FA}_{16} \\ 30\text{F}_{16} \end{array}$$

Содержание работы

Задание

1. Выполнить перевод чисел

- а) из 10–ой с/с в 2–ую систему счисления: 165; 541; 600; 720; 43,15; 234,99.
- б) из 2–ой в 10–ую систему счисления: 110101_2 ; 11011101_2 ; 110001011_2 ; $1001001,111_2$
- в) из 2–ой с/с в 8–ую, 16–ую с/с: 100101110_2 ; 100000111_2 ; 111001011_2 ; 1011001011_2 ; 110011001011_2 ; $10101,10101_2$; $111,011_2$
- г) из 10–ой с/с в 8–ую, 16–ую с/с: 69; 73; 113; 203; 351; 641; 478,99; 555,555
- д) из 8–ой с/с в 10–ую с/с: 35_8 ; 65_8 ; 215_8 ; 327_8 ; 532_8 ; 751_8 ; $45,454_8$
- е) из 16–ой с/с в 10–ую с/с: D8_{16} ; 1AE_{16} ; E57_{16} ; 8E5_{16} ; FAD_{16} ; $\text{AFF},6\text{A7}_{16}$

2. Выпишите целые десятичные числа, принадлежащие следующим числовым промежуткам:

$[10101_2; 110000_2]$; $[14_8; 20_8]$; $[18_{16}; 30_{16}]$

ПРАКТИЧЕСКАЯ РАБОТА № 2

Тема: «Виды информации и способы представления её в ЭВМ»

Цель: научиться выполнять перевод чисел в разные системы счисления.

Оборудование: справочные пособия, микрокалькуляторы.

Содержание работы

Задание

1. Выполнить операции:

а) сложение в двоичной системе счисления

$$\begin{array}{r} + 1001001_2 \\ 1011011_2 \end{array} \quad \begin{array}{r} + 1011101_2 \\ 1110110_2 \end{array} \quad \begin{array}{r} + 1011001_2 \\ 1010101_2 \end{array} \quad \begin{array}{r} +10111001,1_2 \\ 10001101,1_2 \end{array}$$

вычитание в 2–ой системе счисления

$$\begin{array}{r} б) - 100001000_2 \\ 10110011_2 \end{array} \quad \begin{array}{r} - 110101110_2 \\ 10111111_2 \end{array} \quad \begin{array}{r} - 11101110_2 \\ 1011011_2 \end{array} \quad \begin{array}{r} -10111001,1_2 \\ 10001101,1_2 \end{array}$$

в) умножение в 2–ой системе счисления

$$\begin{array}{r} \square 100001_2 \\ 111111_2 \end{array} \quad \begin{array}{r} \square 100101_2 \\ 111011_2 \end{array} \quad \begin{array}{r} \square 111101_2 \\ 111101_2 \end{array} \quad \begin{array}{r} \square 11001,01_2 \\ 11,01_2 \end{array}$$

г) деление в 2–ой системе счисления

1) $111010001001_2 / 111101_2$

2) $100011011100_2 / 110110_2$

3) $10000001111_2 / 111111_2$

ПРАКТИЧЕСКАЯ РАБОТА № 3

Тема: «Виды информации и способы представления её в ЭВМ»

Цель: изучить выполнение арифметических операций $A+B$ в различных системах счисления, изучить принципы выполнения арифметических операций в различных системах счисления.

Оборудование: справочные пособия, микрокалькуляторы.

Справочный материал

Сложение двоичных чисел

Способ сложения столбиком в общем-то такой же как и для десятичного числа. То есть, сложение выполняется поразрядно, начиная с младшей цифры. Если при сложении двух цифр получается СУММА больше девяти, то записывается цифра=СУММА- 10, а ЦЕЛАЯ ЧАСТЬ (СУММА /10), добавляется в старшему разряду. (Сложите пару чисел столбиком вспомните как это делается.) Так и с двоичным числом. Складываем поразрядно, начиная с младшей цифры. Если

получается больше 1, то записывается 1 и 1 добавляется к старшему разряду (говорят "на ум пошло").

Выполним пример: $10011 + 10001$.

	1	0	0	1	1
	1	0	0	0	1
1	0	0	1	0	0

Первый разряд: $1+1 = 2$. Записываем 0 и 1 на ум пошло.

Второй разряд: $1+0+1$ (запомненная единица) $=2$. Записываем 0 и 1 на ум пошло.

Третий разряд: $0+0+1$ (запомненная единица) $= 1$. Записываем 1.

Четвертый разряд $0+0=0$. Записываем 0.

Пятый разряд $1+1=2$. Записываем 0 и добавляем к шестым разрядом 1.

Переведём все три числа в десятичную систему и проверим правильность сложения. $10011 = 1*2^4 + 0*2^3 + 0*2^2 + 1*2^1 + 1*2^0 = 16 + 2 + 1 = 19$

$10001 = 1*2^4 + 0*2^3 + 0*2^2 + 0*2^1 + 1*2^0 = 16 + 1 = 17$

$100100 = 1*2^5 + 0*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 0*2^0 = 32+4=36$

$17 + 19 = 36$ верное равенство

Вычитание двоичных чисел

Вычитать числа, будем также столбиком и общее правило тоже, что и для десятичных чисел, вычитание выполняется поразрядно и если в разряде не хватает единицы, то она занимается в старшем. Решим следующий пример:

	1	1	0	1
-		1	1	0
		1	1	1

Первый разряд. $1 - 0 = 1$. Записываем 1.

Второй разряд $0 - 1$. Не хватает единицы. Занимаем её в старшем разряде. Единица из старшего разряда переходит в младший, как две единицы (потому что старший разряд представляется двойкой большей степени) $2-1 = 1$. Записываем 1.

Третий разряд. Единицу этого разряда мы занимали, поэтому сейчас в разряде 0 и есть необходимость занять единицу старшего разряда. $2-1 = 1$. Записываем 1.

Проверим результат в десятичной системе: $1101 - 110 = 13 - 6 = 7$ (111) Верное равенство.

Умножение в двоичной системе счисления

Для начала рассмотрим следующий любопытный факт. Для того, чтобы умножить двоичное число на 2 (десятичная двойка это 10 в двоичной системе) достаточно к умножаемому числу слева приписать один ноль.

Пример. $10101 * 10 = 101010$

Проверка.

$$10101 = 1*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 1*2^0 = 16 + 4 + 1 = 21$$

$$101010 = 1*2^5 + 0*2^4 + 1*2^3 + 0*2^2 + 1*2^1 + 0*2^0 = 32 + 8 + 2 = 42$$

$$21 * 2 = 42$$

Если мы вспомним, что любое двоичное число разлагается по степеням двойки, то становится ясно, что умножение в двоичной системе счисления сводится к умножению на 10 (то есть на десятичную 2), а стало быть, умножение это ряд последовательных сдвигов. Общее правило таково: как и для десятичных чисел, умножение двоичных выполняется поразрядно. И для каждого разряда второго множителя к первому множителю добавляется один ноль справа. Пример (пока не столбиком):

$1011 * 101$ Это умножение можно свести к сумме трёх поразрядных умножений:

$1011 * 1 + 1011 * 0 + 1011 * 100 = 1011 + 101100 = 110111$ В столбик это же самое можно записать так:

		1	0	1	1
	*		1	0	1
		1	0	1	1
	0	0	0	0	
1	0	1	1		
1	1	0	1	1	1

Примечание: Кстати таблица умножения в двоичной системе состоит только из одного пункта $1 * 1 = 1$

Проверка:

$$101 = 5 \text{ (десятичное)}$$

$$1011 = 11 \text{ (десятичное)}$$

$$110111 = 55 \text{ (десятичное)}$$

$$5 * 11 = 55 \text{ верное равенство}$$

Деление в двоичной системе счисления

Мы уже рассмотрели три действия и думаю уже понятно, что в общем-то действия над двоичными числами мало отличаются от действий над десятичными числами. Разница появляется только в том, что цифр две а не десять, но это только упрощает арифметические операции. Так же обстоит дело и с делением, но для лучшего понимания алгоритм деления разберём более подробно. Пусть нам необходимо разделить два десятичных числа, например 234 разделить на 7. Как мы это делаем.

2	3	4	7	

Мы выделяем справа (от старшего разряда) такое количество цифр, чтобы получившееся число было как можно меньше и в то же время больше делителя. 2 - меньше делителя, следовательно, необходимое нам число 23. Затем делим полученное число на делитель с остатком. Получаем следующий результат:

	2	3	4	7	
-	2	1		3	
		2	4		

Описанную операцию повторяем до тех пор, пока полученный остаток не окажется меньше делителя. Когда это случится, число полученное под чертой, это частное, а последний остаток - это остаток операции. Так вот операция деления двоичного числа выполняется точно также. Попробуем

Пример: 10010111 / 101

1	0	0	1	0	1	1	1	1	0	1

Ищем число, от старшего разряда которое первое было бы больше чем делитель. Это четырехразрядное число 1001. Оно выделено жирным шрифтом. Теперь необходимо подобрать делитель выделенному числу. И здесь мы опять выигрываем в сравнении в десятичной системой. Дело в том, что подбираемый делитель это обязательно цифра, а цифры у нас только две. Так как 1001 явно больше 101, то с делителем всё понятно это 1.

	1	0	0	1	0	1	1	1	1	0	1
-		1	0	1					1		
		1	0	0							

Итак, остаток от выполненной операции 100. Это меньше чем 101, поэтому чтобы выполнить второй шаг деления, необходимо добавить к 100 следующую цифру, это цифра 0. Теперь имеем следующее число:

Полученное число 11 меньше 101, поэтому записываем в частное цифру 0 и опускаем вниз следующую цифру. Получается так:

	1	0	0	1	0	0	1	1		1	0	1		
-		1	0	1						1	1	1	0	
		1	0	0	0									
-		1	0	1										
				1	1	0								
			-	1	0	1								
							1	1	1					

Полученное число больше 101, поэтому в частное записываем цифру 1 и опять выполняем действия. Получается такая картина:

	1	0	0	1	0	0	1	1		1	0	1		
-		1	0	1						1	1	1	0	1
		1	0	0	0									
-		1	0	1										
				1	1	0								
			-	1	0	1								
							1	1	1					
					-	1	0	1						
								1	0					

Полученный остаток 10 меньше 101, но у нас закончились цифры в делимом, поэтому 10 это окончательный остаток, а 1110 это искомое частное.

Проверим в десятичных числах $10010011 = 147$

$101 = 5$

$10 = 2$ $11101 = 29$

	1	4	7	5	
-	1	0		2	9
		4	7		
	-	4	5		
			2		

Содержание работы

1. Выполнить сложение двоичных чисел:

а) $11001 + 101 =$ _____

б) $11001 + 11001 =$ _____

в) $1001 + 111 =$ _____

г) $10011 + 101 =$ _____

д) $11011 + 1111 =$ _____

е) $11111 + 10011 =$ _____

2. Выполнить вычитание двоичных чисел:

а) $11001 - 1001 =$ _____ б) $1011 - 110 =$ _____

в) $10001 - 101 =$ _____ г) $10101 - 11 =$ _____

д) $101001 - 1111 =$ _____ е) $111111 - 101010 =$ _____

3. Выполнить умножение двоичных чисел:

а) $1101 * 1110 =$ _____ б) $1010 * 110 =$ _____

в) $1011 * 11 =$ _____ г) $101011 * 1101 =$ _____

д) $10010 * 1001 =$ _____

ПРАКТИЧЕСКАЯ РАБОТА № 4

Тема: «Виды информации и способы представления её в ЭВМ»

Цель: научиться выполнять перевод чисел в разные системы счисления.

Оборудование: справочные пособия, микрокалькуляторы.

Содержание работы

Задание

1) сложение 8-ых чисел

$$\begin{array}{cccccc}
 + 715_8 & + 524_8 & + 712_8 & + 321_8 & + 5731_8 & + 6351_8 \\
 73_8 & 57_8 & 763_8 & 765_8 & 1376_8 & 737_8
 \end{array}$$

2) вычитание 8-ых чисел

$$\begin{array}{cccccc}
 - 137_8 & - 436_8 & - 705_8 & - 538_8 & - 7213_8 & \\
 72_8 & 137_8 & 76_8 & 57_8 & 537_8 &
 \end{array}$$

3) сложение 16-ых чисел

$$\begin{array}{cccccc}
 + A13_{16} & + F0B_{16} & + 2EA_{16} & + ABC_{16} & + A2B_{16} & \\
 16F_{16} & 1DA_{16} & FCE_{16} & C7C_{16} & 7F2_{16} &
 \end{array}$$

4) вычитание 16-ых чисел

$$\begin{array}{cccccc}
 - A17_{16} & - DFA_{16} & - FO5_{16} & - DE5_{16} & - D3C_{16} & \\
 1FC_{16} & 1AE_{16} & AD_{16} & AF_{16} & D1F_{16} &
 \end{array}$$

5) Вычислите выражения (ответ записать в восьмеричной C/C):

$$(1111101_2 + AF_{16}) / 36_8;$$

$$125_8 + 11101_2 \square A2_{16} / 1417_8$$

ПРАКТИЧЕСКАЯ РАБОТА № 5

Тема: «Логические элементы электронно-вычислительной техники (ЭВТ)»

Цель: научиться переводить основные логические элементы.

Оборудование: справочный материал, персональный компьютер с выходом в Интернет.

Справочный материал

Логическое высказывание – это любое повествовательное предложение, в отношении которого можно однозначно сказать, истинно оно или ложно.

Математический аппарат алгебры логики очень удобен для описания того, как функционируют аппаратные средства компьютера, поскольку основной системой счисления в компьютере является двоичная, в которой используются цифры 1 и 0, а значений логических переменных тоже два: 1 и 0.

Логический элемент компьютера — это часть электронной логической схемы, которая реализует элементарную логическую функцию.

Логическими элементами компьютеров являются электронные схемы И, ИЛИ, НЕ, И-НЕ, ИЛИ-НЕ и др. (называемые также вентилями), а также триггер. С помощью этих схем можно реализовать любую логическую функцию, описывающую работу устройств компьютера. Работу логических элементов описывают с помощью таблиц истинности.

Схемы, реализующие логические функции, называются логическими элементами. Основные логические элементы имеют, как правило, один выход

(Y) и несколько входов, число которых равно числу аргументов ($X_1; X_2; X_3 \dots X_N$). На электрических схемах логические элементы обозначаются в виде прямоугольников с выводами для входных (слева) и выходных (справа) переменных. Внутри прямоугольника изображается символ, указывающий функциональное назначение элемента.

Схема И реализует конъюнкцию двух или более логических значений. Условное обозначение на структурных схемах схемы И с двумя входами представлено на рисунке 1.

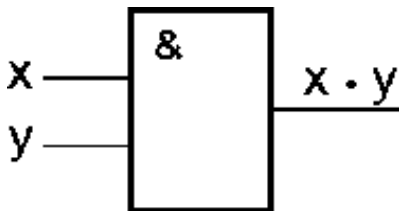


Рисунок 1 - Структурная схема схемы И

Таблица 1 - Таблица истинности схемы И

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

Единица на выходе схемы И будет тогда и только тогда, когда на всех входах будут единицы. Когда хотя бы на одном входе будет ноль, на выходе также будет ноль.

Связь между выходом z этой схемы и входами x и y описывается соотношением: $z = x \cdot y$ (читается как "x и y"). Операция конъюнкции на структурных схемах обозначается знаком "&" (читается как "амперсэнд"), являющимся сокращенной записью английского слова and.

Схема ИЛИ реализует дизъюнкцию двух или более логических значений. Когда хотя бы на одном входе схемы ИЛИ будет единица, на её выходе также будет единица.

Условное обозначение на структурных схемах схемы ИЛИ с двумя входами представлено на рисунке 2. Знак "1" на схеме — от устаревшего обозначения дизъюнкции как " ≥ 1 " (т.е. значение дизъюнкции равно единице, если сумма значений операндов больше или равна 1). Связь между выходом z этой схемы и входами x и y описывается соотношением: $z = x \vee y$ (читается как "x или y").

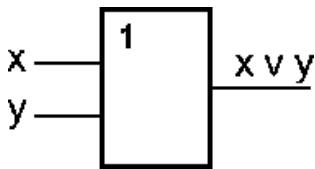


Рисунок 2 – Структурная схема схемы ИЛИ

Таблица 2 – Таблица истинности схемы ИЛИ

x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

Схема НЕ (инвертор) реализует операцию отрицания. Связь между входом x этой схемы и выходом z можно записать соотношением $z = \bar{x}$, где \bar{x} читается как "не x " или "инверсия x ".

Если на входе схемы 0, то на выходе 1. Когда на входе 1, на выходе 0. Условное обозначение на структурных схемах инвертора представлена на рисунке 3.

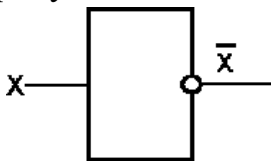


Рисунок 3 - Структурная схема инвертора

Таблица 3 - Таблица истинности схемы НЕ

x	\bar{x}
0	1
1	0

Схема И—НЕ состоит из элемента И и инвертора и осуществляет отрицание результата схемы И. Связь между выходом z и входами x и y схемы записывают следующим образом: $z = \overline{x \cdot y}$, читается как "инверсия x и y ". Условное обозначение на структурных схемах схемы И—НЕ с двумя входами представлено на рисунке 4.

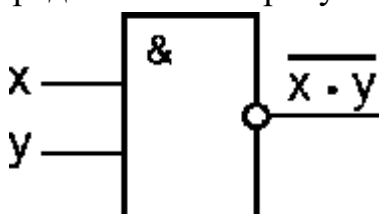


Рисунок 4 - Структурная схема схемы И—НЕ

Таблица 4 - Таблица истинности схемы И—НЕ

x	y	x * y
0	0	1
0	1	1
1	0	1
1	1	0

Схема ИЛИ—НЕ состоит из элемента ИЛИ и инвертора и осуществляет отрицание результата схемы ИЛИ. Связь между выходом z и входами x и y схемы записывают следующим образом: $x \vee y$, читается как "инверсия x или y". Условное обозначение на структурных схемах схемы ИЛИ—НЕ с двумя входами представлено на рисунок 5.

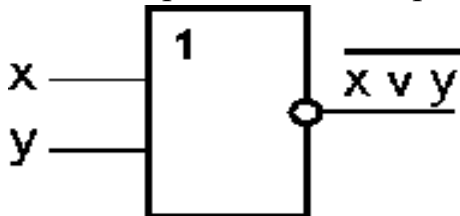


Рисунок 5 - Структурная схема схемы ИЛИ—НЕ

Таблица 5 - Таблица истинности схемы ИЛИ—НЕ

x	y	$x \vee y$
0	0	1
0	1	0
1	0	0
1	1	0

2. Пример выполнения заданий

Проверить тождественность логических функций X и Y

$$X = \neg(f_1 * f_2 + \neg f_1 * f_3) * (f_1 * \neg f_2 + f_2 * f_3) * (f_1 * f_3 + \neg f_2) \quad Y = (f_1 + f_2 * f_3) * (\neg f_1 + \neg f_2 * f_3) * (\neg f_1 * \neg f_2 + f_1 * f_3)$$

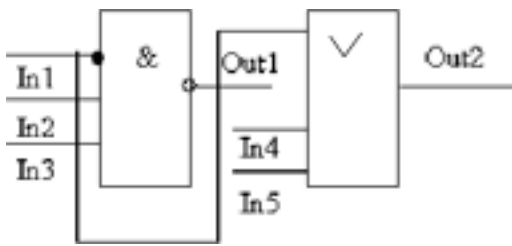
Таблица 6 – Значения функции x						
f1	f2	f3	$\wedge(f1*f2+\wedge f1*f3)$	$(f1*\wedge f2+f2*f3)$	$(f1*f3+\wedge f2)$	x
0	0	0	$(0*0+1*0) 1$	$(0*1+0*0) 0$	$(0*0+1) 1$	0
0	0	1	$(0*0+1*1) 0$	$(0*1+0*1) 0$	$(0*1+1) 1$	0
0	1	0	$(0*1+1*0) 1$	$(0*0+1*0) 0$	$(0*0+0) 0$	0
1	0	0	$(1*0+0*0) 1$	$(1*1+0*0) 1$	$(1*0+1) 1$	1
0	1	1	$(0*1+1*1) 0$	$(0*0+1*1) 1$	$(0*1+0) 1$	0
1	0	1	$(1*0+0*1) 1$	$(1*1+0*1) 1$	$(1*1+1) 1$	1
1	1	0	$(1*1+0*0) 0$	$(1*0+1*0) 0$	$(1*0+0) 0$	0
1	1	1	$(1*1+0*1) 0$	$(1*0+1*1) 1$	$(1*1+0) 1$	0

Таблица 7 – Значения функции y

f1	f2	f3	$(f1+f2*f3)$	$(\wedge f1+\wedge f2*f3)$	$(\wedge f1*\wedge f2+f1*f3)$	y
0	0	0	$(0+0*0) 0$	$(1+1*0) 1$	$(1*1+0*0) 1$	0
0	0	1	$(0+0*1) 0$	$(1+1*1) 1$	$(1*1+0*1) 1$	0
0	1	0	$(0+1*0) 0$	$(1+0*0) 1$	$(1*0+0*0) 0$	0
1	0	0	$(1+0*0) 1$	$(0+1*0) 0$	$(0*1+1*0) 0$	0
0	1	1	$(0+1*1) 1$	$(1+0*1) 1$	$(1*0+0*1) 0$	0
1	0	1	$(1+0*1) 1$	$(0+1*1) 1$	$(0*1+1*1) 1$	1
1	1	0	$(1+1*0) 1$	$(0+0*0) 0$	$(0*0+1*1) 0$	0
1	1	1	$(1+1*1) 1$	$(0+0*1) 0$	$(0*0+1*2) 0$	0

Вывод: при одинаковых значениях на входе функций f_1, f_2, f_3 , значения функций x и y различны, соответственно x и y не тождественны.

Составить логическое выражение по схеме и таблицы истинности для выходных функций.



$$\text{Out 1} = \text{In1} * \text{In2} * \text{In3}$$

Таблица 8 – Значения функции Out1

In1	In2	In3	$\overline{\text{In1}}$	$\overline{\text{In1}} * \text{In2} * \text{In3}$	$\overline{\overline{\text{In1}} * \text{In2} * \text{In3}}$
0	0	0	1	0	1
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	0	0	1
1	1	0	0	0	1
1	1	1	0	0	1

$$\text{Out2} = \text{In1} + \text{In4} + \text{In5}$$

Таблица 9 – Значения функции Out2

In1	In4	In5	$\overline{\text{In1}}$	$\overline{\text{In1}} + \text{In4} + \text{In5}$
0	0	0	1	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	0	0	0
1	0	1	0	0

1	1	0	0	0
1	1	1	0	1

Содержание работы

Задание 1. Составить логическое выражение по схеме и таблицы истинности для выходных функций.

Вариант 1

✓

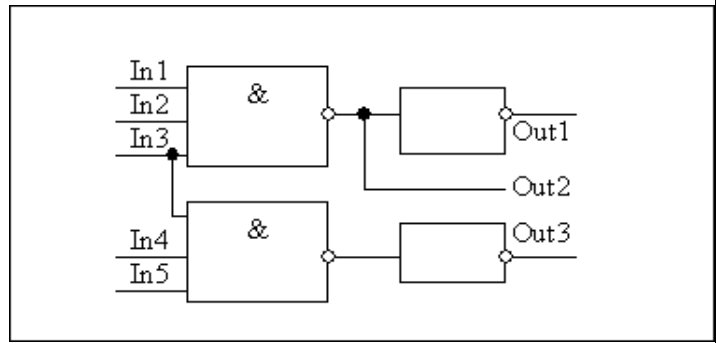
Вариант 2

✓

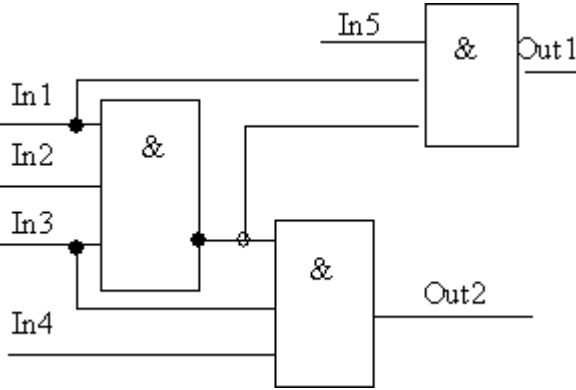
Вариант 3

Вариант 4

✓ ✓

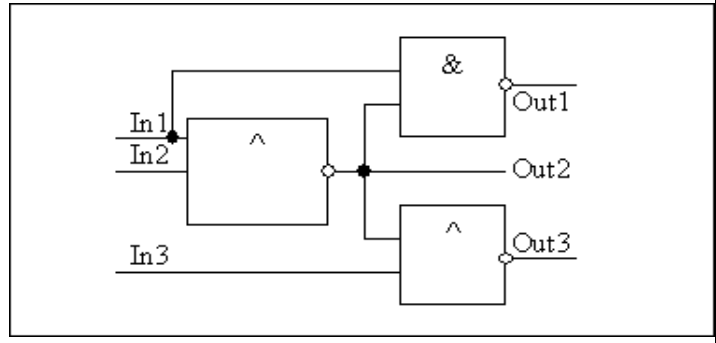


Вариант 5



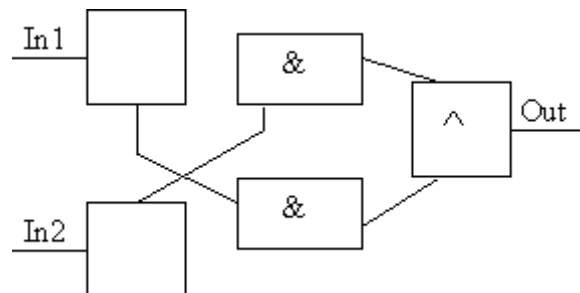
Вариант 6

✓ ✓

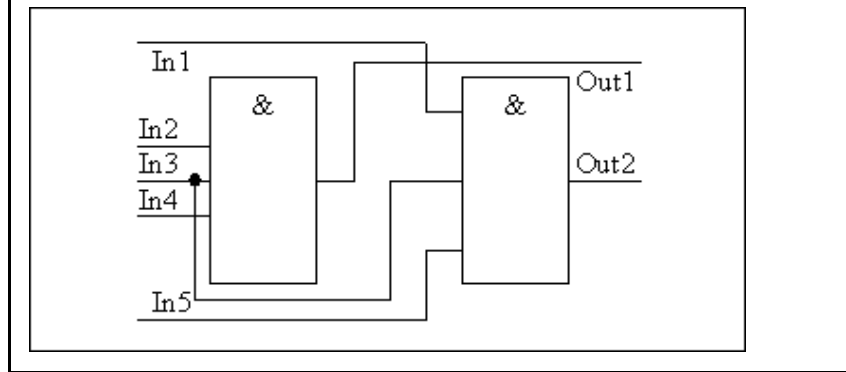


Вариант 7

✓ ✓



Вариант 8



Задание 2. Проверить тождественность логических функций X и Y.

1. $X=(f1+f2*f3)*(f1*f3+^f2)*(^f1*^f2+f1*f3)$

$Y=(^f1+^f2*f3)*(f1*f2+^f1*f3)*(f1*^f2+f2*f3)$

2. $X=(f1+f2*f3)*^(^f1*^f2+f*f3)*(^f1+^f2*f3)$

$Y=^(f1*f3+^f2)*(f1*^f2+f2*f3)*(f1*f2+^f1*f3)$

3. $X=(f1+f2*f3)*(^f1*^f2+f1*f3)*^(^f1+^f2*f3)$

$Y=(^f1*f2+^f2*f3)*^(f1*f3+f2)*^(f1*^f2+f2*f3)$

4. $X=(f1+f2*f3)*(^f1*^f2+f1*f3)*^(f1*^f2+f2*f3)$

$Y=(^f1+^f2*f3)*(f1*f3+^f2)*^(f1*f2+^f1*f3)$

5. $X=^(^f1+^f2*f3)*^(f1*^f2+f2*f3)*(^f1*^f2+f1*f3)$

$Y=^(f1*f3+f2)*^(f1+f2*f3)*(f1*f2+^f1*f3)$

6. $X=^(f1+f2*f3)*^(f1*f3+^f2)*(f1*^f2+f2*f3)$

$Y=(f1*f2+^f1*f3)*(^f1+^f2*f3)*(^f1*^f2+f1*f3)$

7. $X=(f1+f2*f3)*(f1*f2+^f1*f32)*(f1*^f2+f2*f3)$

$Y=^(^f1*f2+^f2*f3)*(^f1*^f2+f1*f3)*(f1*f3+^f2)$

8. $X=(^f1*f2+f2*f3)*^(f1+f2*f3)*(^f1+^f2*f3)$

$Y=^(f1*f3+^f2)*(f1*^f2+f2*f3)*^(^f1*^f2+f1*f3)$

9. $X=^(f1*f2+^f1*f3)*(f1*^f2+f2*f3)*(f1*f3+^f2)$

$Y=(f1+f2*f3)*(^f1+^f2*f3)*(^f1*^f2+f1*f3)$

ПРАКТИЧЕСКАЯ РАБОТА № 6

Тема: «Последовательные цифровые устройства»

Цель: изучить понятие триггер и способы его реализации.

Оборудование: справочный материал, персональный компьютер с выходом в Интернет.

Справочный материал

Входы триггера

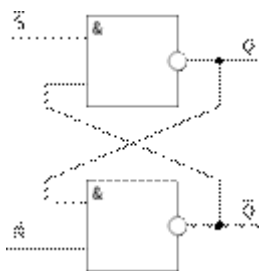
Входы триггеров обычно обозначают следующим образом:

· S (от англ. *Set*, установить) — вход в RS-триггере;

- R (от англ. *Reset*, сброс) — вход в RS-триггере;
- J (от англ. *Jump*^[4], прыжок) — вход в JK-триггере;
- K (от англ. *Kill*, убить) — вход в JK-триггере;
- T (от англ. *Toggles*, переключить) — счётный вход в T-триггере;
- C (от англ. *Clock*, время) вход синхронизирующего сигнала. При тактировании по фронту он часто обозначается стрелкой: стрелка внутрь — тактирование по переднему фронту, наружу — по заднему.
- D (от англ. *Delay*, задержка) — вход в D-триггере;
- E или EN (от англ. *Enable*, разрешить) — дополнительный асинхронный управляющий вход для разрешения приёма информации (иногда используют букву V).

Входы **J, K, T, D** всегда синхронные, т.е. тактируются по синхронизирующему сигналу на входе **C**. Разумеется, в каждом конкретном триггере имеются лишь некоторые из перечисленных входных линий. Входы **S** и **R** зачастую присутствуют не только в RS триггерах, но и в других типах триггеров, где предназначены, в основном, для асинхронного сброса устройства в 0 или установки в 1.

RS-триггер, или SR-триггер

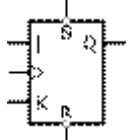


Одна из наглядных схем реализации асинхронного RS-триггера на базе двух элементов **2И-НЕ**(NAND2)

RS-триггер, или SR-триггер — триггер, который сохраняет своё предыдущее состояние при нулевых входах и меняет своё выходное состояние при подаче на один из его входов единицы. При подаче единицы на вход **S** (от англ. *Set* - установить) выходное состояние становится равным логической единице. А при подаче единицы на вход **R** (от англ. *Reset* - сбросить) выходное состояние становится равным логическому нулю. Если RS-триггер синхронный, то состояние его входов учитывается только в момент тактирования, например по переднему фронту импульса. Состояние, при котором на оба входа **R** и **S** одновременно поданы логические единицы, является запрещённым. Так, например, схема RS-триггера, изображённая на рисунке, при подаче на оба инверсных входа логического нуля перейдёт в состояние, когда на обоих выходах будут единицы, что не соответствует логике выхода триггера, поскольку инверсный выход \bar{Q} будет равен неинверсному Q , т.е.

RS-триггер используется для создания сигнала с положительным и отрицательным фронтами, отдельно управляемыми посредством стробов, разнесённых во времени.

JK-триггер

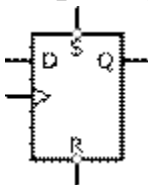


Символ JK-триггера с дополнительными асинхронными входами **S** и **R**, аналогично представлению в среде разработки

JK-триггер работает также как RS-триггер, с одним лишь исключением: при подаче логической единицы на оба входа **J** и **K** состояние выхода триггера изменяется на противоположное. Вход **J** (от англ. *Jump* - прыжок) аналогичен входу **S** у RS-триггера. Вход **K** (от англ. *Kill* - убить) аналогичен входу **R** у RS-триггера. При подаче единицы на вход **J** и нуля на вход **K** выходное состояние триггера становится равным логической единице. А при подаче единицы на вход **K** и нуля на вход **J** выходное состояние триггера становится равным логическому нулю. JK-триггер в отличие от RS-триггера не имеет запрещённых состояний на основных входах, однако это никак не помогает при нарушении правил разработки логических схем. На практике применяются только синхронные JK-триггеры, то есть состояния основных входов **J** и **K** учитываются только в момент тактирования, например по положительному фронту импульса на входе синхронизации.

На базе JK-триггера возможно построить D-триггер или T-триггер. Как можно видеть в таблице истинности JK-триггера, он переходит в инверсное состояние каждый раз при одновременной подаче на входы **J** и **K** логической 1. Это свойство позволяет создать на базе JK-триггера T-триггер, объединив входы **J** и **K**^[5].

D-триггер

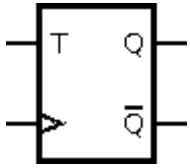


Символ D-триггера с дополнительными асинхронными входами **S** и **R**

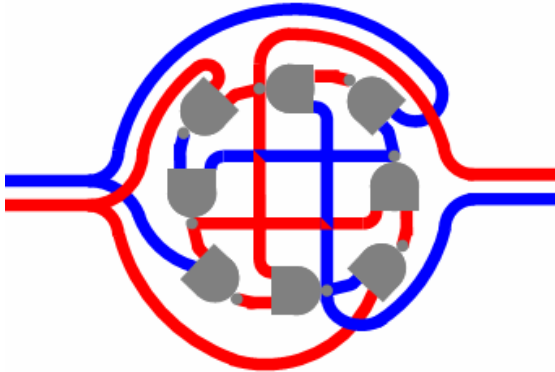
D-триггер (**D** от англ. *delay* - задержка) - запоминает состояние входа и выдаёт его на выход. D-триггеры имеют, как минимум, два входа: информационный **D** и синхронизации **S**. Сохранение информации в D-триггерах происходит в момент прихода активного фронта на вход **S**. Так как информация на выходе остаётся неизменной до прихода очередного импульса синхронизации, D-триггер называют также триггером с запоминанием информации или триггером-защёлкой. Рассуждая чисто теоретически, D-триггер можно образовать из любых RS- или JK-триггеров, если на их входы одновременно подавать взаимно инверсные сигналы.

D-триггер в основном используется для реализации защёлки. Так, например, для снятия 32 бит информации с параллельной шины, берут 32 D-триггера и объединяют их входы синхронизации для управления записью информации в защёлку, а 32 **D** входа подсоединяют к шине.

Т-триггер



Изображение Т-триггера на схемах.



Работа схемы Т-триггера (при $T=1$) на базе восьми **2И-НЕ** логических вентилях. Слева — входы, справа — выходы. Синий цвет соответствует 0, красный — 1. **Т-триггер** по каждому такту изменяет своё логическое состояние на противоположное при единице на входе T , и не изменяет выходное состояние при нуле на входе T . Т-триггер часто называют счётным триггером. Т-триггер может строиться как на JK, так и на D-триггерах. Как можно видеть в таблице истинности JK-триггера, он переходит в инверсное состояние каждый раз при одновременной подаче на входы J и K логической 1. Это свойство позволяет создать на базе JK-триггера Т-триггер, объединяя входы J и K . Наличие в D-триггере динамического С входа позволяет получить на его основе Т-триггер. При этом вход D соединяется с инверсным выходом, а на вход С подаются счётные импульсы. В результате триггер при каждом счётном импульсе запоминает значение \bar{Q} , то есть будет переключаться в противоположное состояние.

Т-триггер — триггер, который меняет своё предыдущее состояние на противоположное под воздействием каждого входного сигнала равного логической «1».

Характеристическое уравнение, определяющее логику функционирования асинхронного Т-триггера:

$$Q^{n+1} = T^n \bar{Q}^n \vee \bar{T}^n Q^n$$

Экспериментальное исследование работы асинхронного Т-триггера.

Для исследования работы Т-триггера используется модуль «Триггеры» лабораторного стенда. Если текущее состояние триггера Q_n не совпадает с требуемым состоянием согласно таблице истинности, то для переключения триггера необходимо подать единичный импульс или сигнал логической «1» на вход триггера.

Перед началом работы с триггером его вход необходимо подключить с помощью соединительных проводников к одному из выходов логических сигналов модуля «Задание логических уровней и логические элементы». Исследование работы Т-триггера производится путем установки триггера в необходимое начальное состояние Q_n и подачи на его вход сигналов логического «0» или логической «1» в соответствии с таблицей истинности.

Таблица истинности асинхронного Т-триггера.

№ набора Т	Q_n	Q_{n+1}
0	0	0
1	0	1
2	1	0
3	1	1

Содержание работы

Задание. В рабочей тетради изобразить УГО исследуемого триггера и указать назначение входов/выходов. Полученные значения сигналов на выходах триггера занести в таблицу истинности в рабочей тетради.

ПРАКТИЧЕСКАЯ РАБОТА № 7

Тема: «Последовательные цифровые устройства»

Цель: Изучение назначения и функций регистра. Знакомство с принципом работы регистров.

Оборудование: справочный материал, персональный компьютер с выходом в Интернет.

Справочный материал

Регистры – устройства для временного хранения и преобразования информации в виде многоразрядных двоичных чисел. Регистры наряду со счетчиками и запоминающими устройствами являются наиболее распространенными устройствами цифровой техники. При сравнительной простоте регистры обладают большими функциональными возможностями.

Они используются в качестве управляющих и запоминающих устройств, генераторов и преобразователей кодов, счетчиков, делителей частоты, узлов временной задержки.

Элементами структуры регистров являются триггеры D- или JK- типа с динамическим или статическим управлением. Одиночный триггер может запоминать (регистрировать) один разряд (бит) двоичной информации. Такой триггер можно считать одноразрядным регистром. Занесение информации в

регистр называют операцией ввода или записи. Выдача информации к внешним устройствам характеризует операцию вывода или считывания.

Запись информации в регистр не требует его предварительного обнуления.

Все регистры в зависимости от функциональных свойств подразделяются на две

категории – накопительные (регистры памяти, хранения) и сдвигающие. В свою очередь, сдвигающие регистры делятся по способу ввода и вывода информации на параллельные и последовательно-параллельные, и комбинированные, по направлению передачи (сдвига) информации – на однонаправленные и реверсивные.

На рисунке 1 показана схема простейшего четырехразрядного регистра на D – триггерах, в котором информация заносится последовательно, начиная с младшего разряда.

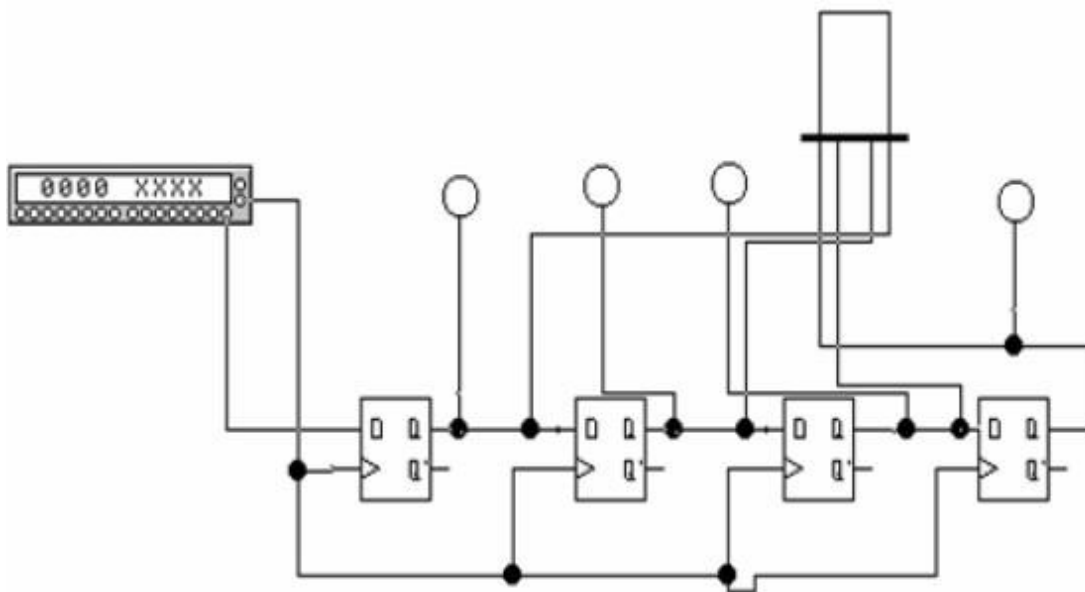


Рисунок 1

Содержание работы

Задание: Ответить на вопросы и выполнить задание

1. Что такое регистр, какие функции он может выполнять?
2. Назовите типы регистров и их возможные применения.
3. Смодулируйте приведенную выше схему и проанализируйте работу регистра.

ПРАКТИЧЕСКАЯ РАБОТА № 8

Тема: «Последовательные цифровые устройства»

Цель: Изучение назначения и функции устройства счетчик. Знакомство с принципом работы устройства счетчик.

Оборудование: справочный материал, персональный компьютер с выходом в Интернет.

Справочный материал

Счетчиком называют устройство, сигналы на выходе которого отображают число импульсов, поступивших на счетный выход. Триггер может служить примером простейшего счетчика. Такой счетчик считает до двух. Счетчик, образованный цепочкой из m триггеров, может подсчитать в двоичном коде 2^m импульсов. Каждый из триггеров такой цепочки называют разрядом счетчика. Число m определяет количество разрядов двоичного числа, которое может быть записано в счетчик. Число $K_{сч}=2^m$ называют коэффициентом (модулем) счета. Информация снимается с прямых и (или) инверсных выходов всех триггеров. В паузах между входными импульсами триггеры сохраняют свои состояния, т. е. счетчик запоминает число входных импульсов.

Нулевое состояние всех триггеров принимается за нулевое состояние счетчика в целом. Остальные состояния складываются по числу поступивших входных импульсов.

Когда число входных импульсов $N_{вх} > K_{сч}$ происходит переполнение, после чего счетчик возвращается в нулевое состояние и цикл повторяется. Коэффициент счета, таким образом, характеризует число входных импульсов, необходимое для одного цикла и возвращения в исходное состояние.

Счетчики различаются числом и типами триггеров, способами связей между ними, кодом, организацией счета и другими показателями. Цифровые счетчики классифицируются по следующим параметрам:

- Коэффициент счета – двоичные; двоично-десятичные или с другим основанием счета; с произвольным постоянным и переменным (программируемым) коэффициентом счета;
- Направление счета – суммирующие, вычитающие и реверсивные;
- Способ организации внутренних связей – с последовательным, параллельным или комбинированным переносом, кольцевые.

Классификационные признаки независимы и могут встречаться в различных сочетаниях: например, суммирующие счетчики бывают как с последовательным, так и с параллельным переносом, могут иметь двоичный, десятичный и иной коэффициент счета.

Схема четырехразрядного двоичного счетчика с последовательным переносом на D – триггерах приведена на рисунке 1.

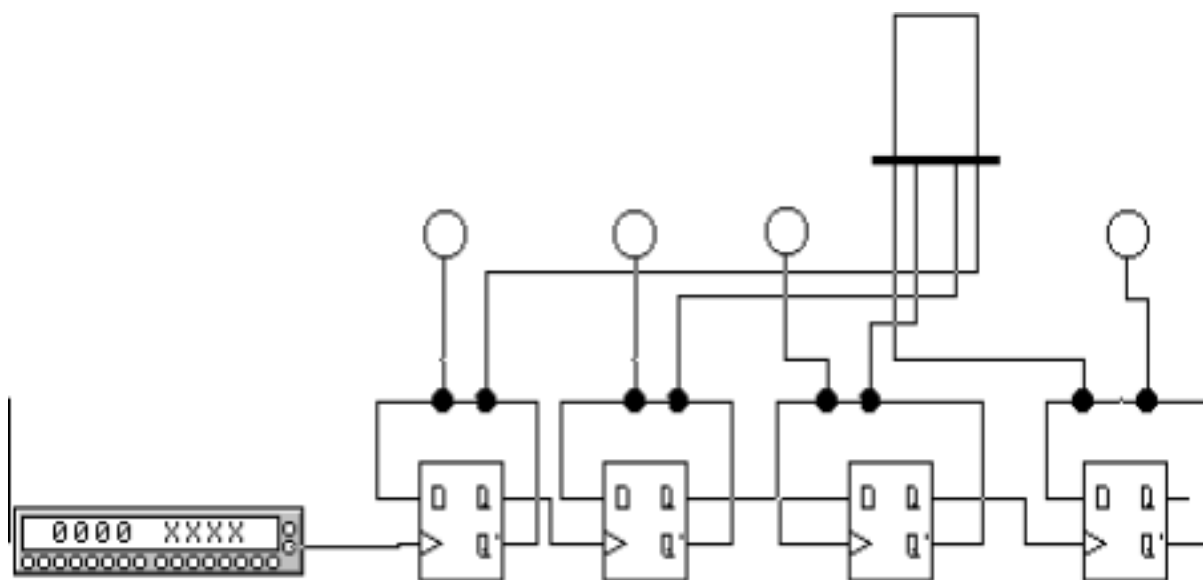


Рисунок 1

На вход счетчика подаются импульсы с выхода синхросигналов генератора слова, которые генерируются при каждом нажатии клавиши STEP. Каждый триггер счетчика осуществляет деление на 2, сигнал переноса передается последовательно от одного разряда к другому. Состояние разрядов счетчиков в двоичном коде индицируются логическим пробником (индикатором), а в десятичном – семисегментным индикатором.

Содержание работы

Задание: Ответить на вопросы и выполнить задание

1. Что такое счетчик, какие функции он может выполнять?
2. Назовите типы счетчиков и их возможные применения.
3. Смоделируйте приведенную выше схему и проанализируйте работу счетчика.

ПРАКТИЧЕСКАЯ РАБОТА № 9

Тема: «Основные типы микропроцессоров, структуры команд, структура устройства управления»

Цель: изучить принцип организации и функционирования современных микропроцессоров.

Оборудование: справочный материал, персональный компьютер с выходом в Интернет.

Справочный материал

1. Микропроцессор

Микропроцессор выполняют всю обработку информации в компьютере. Микропроцессор дешифрирует и выполняет команды программы, организует обращения к оперативной памяти, в нужных случаях инициирует работу

периферийных устройств, воспринимает и обрабатывает запросы, поступающие из устройств машины и из внешней среды (“запросы прерывания”).

Для определений временных соотношений между различными этапами операции используется понятие *машинного такта*. Машинный такт определяет интервал времени, в течение которого выполняется одна или одновременно несколько микроопераций. Границы тактов задаются синхросигналами, вырабатываемыми специальной схемой — генератором синхросигналов. Также данная характеристика микропроцессора носит название *тактовая частота*, которая определяет, сколько микроопераций процессор выполнит за одну секунду.

2. Общая классификация микропроцессоров

В настоящее время насчитывается большое количество разнообразных процессоров. Приведем их общую классификацию.

По числу больших интегральных схем (БИС) в составе микропроцессора различают:

- однокристалльные микропроцессоры;
- многокристалльные микропроцессоры;
- многокристалльные секционные микропроцессоры.

Однокристалльные микропроцессоры получают при реализации всех аппаратных средств процессора в виде одной БИС или СБИС (сверхбольшой интегральной схемы). При усложнении микропроцессора приходится разбивать его на отдельные блоки. В этом случае каждый блок реализуется на отдельном кристалле, в результате чего процессор становится многокристалльным.

Многокристалльные секционные микропроцессоры получают в том случае, когда отдельные блоки процессора приходится логически разбивать дополнительно на секции. Секционность микропроцессоров дает возможность наращивать разрядность обрабатываемых данных или усложнять устройство управления микропроцессора.

По назначению различают:

- универсальные микропроцессоры;
- специализированные микропроцессоры.

Универсальные микропроцессоры могут быть применены для решения широкого круга разнообразных задач. Специализация МП, т.е. его проблемная ориентация на ускоренное выполнение определенных функций позволяет резко увеличить эффективную производительность при решении только определенных задач.

По виду обрабатываемых входных сигналов различают:

- цифровые микропроцессоры;
- аналоговые микропроцессоры.

Цифровые микропроцессоры работают с информацией представленной в виде числовых значений (дискретная форма). Аналоговые микропроцессоры работают с информацией, которая представлена в аналоговой форме, т.е. в виде непрерывного ряда значений.

По характеру временной организации работы микропроцессоры делятся на:

- синхронные микропроцессоры;
- асинхронные микропроцессоры.

Синхронные микропроцессоры – это микропроцессоры, в которых начало и конец выполнения операций задаются специальным устройством управления. Т.е. если в микропроцессоре присутствует устройство управления, то он относится к синхронным.

Асинхронные микропроцессоры позволяют начало выполнения каждой следующей операции определить по сигналу фактического окончания выполнения предыдущей операции.

По количеству выполняемых программ различают:

- однопрограммные микропроцессоры;
- многопрограммные микропроцессоры.

В однопрограммных микропроцессорах выполняется только одна программа. Переход к выполнению другой программы происходит после завершения текущей программы. В много- или мультипрограммных микропроцессорах одновременно выполняется несколько программ.

Содержание работы

Практическое задание

1. Используя свой персональный компьютер (или его макет) определите модель используемого микропроцессора в вашем персональном компьютере. Запишите **ответ в отчет**.
2. Определите фирму-производителя микропроцессора. Запишите **ответ в отчет**.
3. Определите тактовую частоту микропроцессора. Запишите ответ в отчет.
4. Определите установочный разъем микропроцессора (можете использовать Интернет для поиска информации). Запишите **ответ в отчет**.
5. Самостоятельно **проведите классификацию** имеющегося микропроцессора.
Запишите ответ в отчет.

ПРАКТИЧЕСКАЯ РАБОТА № 10

Тема: «Основные типы микропроцессоров, структуры команд, структура устройства управления»

Цель: изучить структуру современных микропроцессоров.

Оборудование: справочный материал, персональный компьютер с выходом в Интернет.

Справочный материал

1. Архитектуры микропроцессоров

Существует две основные архитектуры современных процессоров – это архитектуры CISC и RISC. **CISC** (CISC - Complete Instruction Set Computer) – это процессоры с полным набором команд, **RISC** (RISC - Reduced Instruction Set Computer) – это процессоры с сокращенным набором команд. Разберемся, чем одна архитектура отличается от другой.

Набор команд CISC был разработан для удобства программистов, которые вынуждены были писать программы для компьютеров на языке Ассемблер. Для ускорения процесса разработки программ в систему команд CISC были введены удобные команды, которые как бы представляли собой подпрограммы. В итоге, команды CISC-процессора имеют разную длину и время выполнения. К тому же CISC-процессор отличается невысокой производительностью, т.к. для выполнения некоторых команд требуется несколько машинных тактов.

В общем случае для **CISC-процессоров** характерно следующее:

- небольшое число регистров общего назначения;
- большое количество машинных команд, которые выполняются за много тактов;
- большое количество методов адресации;
- большое количество форматов команд различной разрядности;
- наличие команд обработки типа регистр-память.

К процессорам класса CISC относятся широко распространенные в персональных компьютерах процессоры фирм Intel, AMD, Cyrix.

В процессорах с набором команд **RISC** все команды имеют одинаковую длину и формат, а также простую адресацию памяти. Каждая команда выполняет только простые действия за один такт.

В общем случае для для **RISC-процессоров** характерно следующее:

- отделение команд обработки данных от команд работы с памятью;
- выполнение любой команды занимает небольшое количество машинных тактов (предпочтительно один машинный такт);
- логика выполнения команд с целью повышения производительности ориентируется на аппаратную, а не на микропрограммную реализацию;
- используются команды фиксированной длины и фиксированного формата;
- наличие большого числа регистров, что позволяет большему объему данных храниться в регистрах на процессорном кристалле большее время. Это значительно увеличивает быстродействие процессора.

Проще говоря, сущность архитектуры RISC состоит в том, что в процессоре выполняются простые команды за один такт. При этом любую сложную команду можно разбить на несколько простых. Выполнение простых команд происходит быстрее, чем сложных, причем выполнение простых команд может происходить параллельно. Поэтому быстродействие RISC процессоров в общем случае выше, чем у CISC.

Считается, что в будущем процессоры с архитектурой RISC заменят менее перспективные процессоры с архитектурой CISC.

Существует еще одно понятие архитектуры процессоров, которые мы также рассмотрим. Наверняка вы часто встречались с термином «x86» (мы его несколько раз упомянули выше), или «Intel-совместимый процессор». Что за этим скрывается на самом деле? Современный x86-процессор – это процессор, способный исполнять машинный код архитектуры IA32 (архитектура 32-битных процессоров Intel). Этот код исполнял процессор Intel 80386 (известный как «386-й»). В настоящее время всё программное обеспечение для ПК разрабатывается именно для x86-процессоров. Оно выполняется на любом x86-процессоре, независимо от того, кто его произвел.

Кроме того, у архитектуры IA32 существуют дополнительные наборы команд от разработчика, компании Intel: MMX, SSE, SSE2 и SSE3. Также существуют неофициальные расширенные наборы команд: EMMX, 3DNow! и Extended 3DNow! – их разработала компания AMD.

Все перечисленные дополнительные наборы команд служат для увеличения быстродействия при выполнении некоторых операций. Одна команда из дополнительного набора, как правило, выполняет действие, для которого понадобилась бы небольшая программа, состоящая из команд основного набора.

2. Понятия irq и dma

Прерывание IRQ (Interrupt ReQuest - запрос прерывания) – это сигнал, по которому процессор узнает о совершении некоторого события, на которое необходимо “обратить” внимание. Пусть, к примеру, микропроцессор выполняет некоторую программу, и пусть в это время в каком-то внешнем устройстве произошло событие, на которое нужно обратить внимание, (например, на клавиатуре нажата клавиша). Естественно, ждать пока закончится выполнение текущей программы нельзя, она может работать еще долго и за это время может быть нажато много других клавиш, так что информация о первой из нажатых клавиш будет потеряна. Надо сразу, оперативно прореагировать на это событие.

Получив сигнал прерывания, микропроцессор прерывает выполнение текущей последовательности команд, а вместо нее начинает выполнять другую последовательность, соответствующую данному прерыванию.

Все прерывания делятся на три группы:

- аппаратные прерывания;
- логические прерывания;
- программные прерывания.

Аппаратные прерывания связаны с запросами от внутренних или периферийных устройств. Логические возникают при работе самого микропроцессора. Программные инициируются выполняемой программой.

Для IBM PC AT на базе процессоров Pentium предусмотрено было **16 линий IRQ**, часть которых заняты внутренними устройствами, а остальные

используется внешними или не используются. В настоящее время число прерываний составляет несколько десятков.

Таким образом, число периферийных устройств, подключаемых к персональному компьютеру с использованием прерываний IRQ, не может превышать пяти.

DMA (Direct Memory Access) – это режим прямого доступа к памяти, когда периферийное устройство связано с оперативной памятью компьютера непосредственно, минуя микропроцессор. Этот режим наиболее эффективен, когда требуется высокая скорость обмена при передаче большого количества информации.

На IBM PC AT есть **8** независимых каналов DMA. Каналы DMA распределены следующим образом:

- 0 - микропроцессор;
- 1 - не используется;
- 2 - контроллер флоппи-диска;
- 3 - не используется;
- 4 - не используется;
- 5 - не используется;
- 6 - не используется;
- 7 - не используется.

Таким образом, к ПК можно подключить 5 различных устройств, которые используют режим DMA. При этом следует помнить, что не все устройства, требующие применения прерываний IRQ, используют DMA.

Содержание работы

Практическое задание

1. Загрузите ПК. Вызовите программу **Сведения о системе** (**Пуск – Программы – Стандартные – Служебные** или файл **MSINFO32.EXE**).
2. Используя программу **Сведения о системе**, выпишите в отчет общее число прерываний IRQ вашего компьютера.
3. **Выпишите в отчет** основные устройства, которые используют прерывания IRQ вашего компьютера.
4. **Укажите в отчете**, сколько свободных прерываний есть в вашем компьютере.
5. Выпишите в отчет все занятые каналы DMA вашего компьютера.
6. Выпишите в отчет все свободные каналы DMA вашего компьютера.
7. Определите, какой процессор в настоящее время является наиболее оптимальным при выборе компьютера для дома, для выполнения графических работ, для офисной работы. Обоснуйте и докажите свой ответ.

ПРАКТИЧЕСКАЯ РАБОТА № 11

Тема: «Методы цифровой обработки сигналов»

Цель: изучить структуру и группы команд, выполняемые процессором.

Оборудование: справочный материал, персональный компьютер с выходом в Интернет.

Справочный материал

Программа помещается в оперативную память вместе с данными. Каждая команда хранится в ячейке или группе ячеек и имеет свой *адрес*.

Команда – это элементарная инструкция машине, выполняемая ею автоматически без каких либо дополнительных указаний и пояснений.

Все команды имеют одинаковую *структуру*. Машинная команда состоит из двух частей: операционной и адресной.

Код Операции (КОП)	Адреса
--------------------	--------

1. Код операции (операционная часть) определяет, какую команду нужно выполнить

2. Операнд (адресная часть) – над чем выполняется операция

Операционная часть команды (КОП) – это группа разрядов в команде, предназначенная для представления кода операции машине.

Адресная часть – это группа разрядов в команде, в которых записывают коды адресов операндов.

Операнд может быть простым и составным. Он может содержать в себе адресную часть, которая определяет, где хранятся данные и куда поместить результат операции.

В зависимости от количества используемых адресов, различают одноадресные, двух-, трех-, четырехадресные и безадресные команды.

- *Одноадресные команды.* Здесь указывается, где находится одно из чисел, второе должно быть помещено в АЛУ. Для этого существуют специальные команды пересылки данных.
- *Двухадресные команды.* Оба операнда в памяти, их адреса указаны. Результат заносится по одному из адресов.
- *Трехадресные команды.* Два адреса являются операндами, а третий служит для помещения туда результата.
- *Четырехадресные команды.* Два адреса – операнды, третий – результат, четвертый – адрес следующей команды.
- *Безадресные команды.* используются для выполнения служебных операций (очистить экран, заблокировать клавиатуру и так далее).

Команды выполняются друг за другом, а также могут быть выполнены ветвления и циклы.

Группы команд микропроцессора

Семейство микропроцессоров фирмы Intel от 8086 до Pentium имеет *базовую систему команд*, в состав которых входят:

Команды пересылки данных:

внутри процессора (Mov, Push, Pop и др.)

ввода-вывода (In, Out)

Арифметические команды:

основные (сложение, вычитание, умножение, деление)

дополнительные (INC, DEC)

Логические команды:

сдвиг, дизъюнкция, конъюнкция, и/или и др.

Обработка строковых данных:

пересылка, сканирование, сравнение, слияние.

Передачи управления:

безусловный переход, условный переход, прерывания, переход с возвратом.

Управления:

нет операции, внешняя синхронизация и так далее.

Каждая команда имеет много модификаций, чаще всего определяемых режимом адресации операндов.

Типы операндов команд

- Регистровые операнды указываются именами используемых регистров процессора.
- Непосредственные являются всегда числовыми величинами и могут быть в различных системах счисления. Числа различаются по последней букве, сопровождающей число:

b – двоичное число;

q – восьмеричное число;

d – десятичное число;

h – шестнадцатеричное число.

- Операнды в памяти могут указываться с помощью адресов ячеек, символическими именами, константами.

Различные комбинации этих элементов в команде называются *способами адресации*. Например, команда mov (переслать числа) может иметь следующие способы адресации:

mov	r, r
mov	r, m
mov	m, r
mov	imed, r
mov	imed, m
mov	Sr, m
mov	Sr, r
mov	m, Sr

mov	r, Sr
-----	-------

Где r – РОН (регистр общего назначения); m – адрес в памяти;
imed – число; Sr – c

Содержание работы

Задание:

1. Изучить материал теоретической части;
2. Записать микропрограмму суммирования двух чисел;
3. Сделать вывод о проделанной работе.

ПРАКТИЧЕСКАЯ РАБОТА № 12

Тема: «Методы цифровой обработки сигналов»

Цель: изучить структуру и группы команд, выполняемые процессором.

Оборудование: справочный материал, персональный компьютер с выходом в Интернет.

Содержание работы

Задание:

Составить программу для условия: переписать 10000 байтов начиная с адреса А в другое место памяти начиная с адреса В. Оба эти имени относятся к сегменту данных, на начало которого указывает регистр A_5 .

ПРАКТИЧЕСКАЯ РАБОТА № 13

Тема: «Программное обеспечение в сфере профессиональной деятельности»

Цель: познакомиться с основными понятиями языка Турбо Паскаль, правилами записи арифметических выражений. формировать навыки представления арифметических выражений на Паскале.

Оборудование: справочный материал, персональный компьютер с выходом в Интернет.

Справочный материал

Ознакомиться с теоретическим материалом «Основные сведения о языке Турбо Паскаль»:

Для того чтобы правильно записывать арифметические выражения, нужно соблюдать следующие правила:

1. Все символы пишутся в строчку на одном уровне. Проставляются все знаки операций (нельзя пропускать знак умножения).
2. Не допускаются два следующих подряд знака операций (нельзя $A+-B$; можно $A+(-B)$).
3. Операции с более высоким приоритетом выполняются раньше операций с меньшим приоритетом. Порядок убывания приоритетов:
 - вычисление функции;
 - унарная операция смены знака (-);
 - *, /, div, mod;
 - +, - .

4. Несколько записанных подряд операций одинакового приоритета выполняются последовательно слева направо.

5. Часть выражения, заключенная в скобки, вычисляется в первую очередь.

(Например, $(A+B) * (C-D)$ — умножение производится после сложения и вычитания.)

Не следует записывать выражений, не имеющих математического смысла. Например, деление на нуль, логарифм отрицательного числа и т.п.

В Паскале нет операции или стандартной функции возведения числа в произвольную степень. Для вычисления x^y рекомендуется поступать следующим образом:

- если y — целое значение, то степень вычисляется через умножение; например, x^3 — $x \cdot x \cdot x$.
- если y — вещественное значение, то используется следующая математическая формула:
$$x^y = e^{y \ln(x)}$$

На Паскале это будет выглядеть так: $\text{Exp}(y * \text{Ln}(x))$.

Содержание работы

Используя правила записи арифметических выражений на Паскале, выполнить следующие задания:

1) Для следующих формул записать соответствующие арифметические выражения на Паскале:

в) $\frac{a+b}{c} + \frac{c}{ab}$;

г) $\frac{x+y}{a_1} \cdot \frac{a_2}{x-y}$;

д) $10^4 \alpha - 3 \frac{1}{5} \beta$;

е) $\left(1 + \frac{x}{2!} + \frac{y}{3!}\right) / \left(1 + \frac{2}{3+xy}\right)$.

2) Записать математические формулы, соответствующие следующим выражениям на Паскале:

а) $(p+q)/(r+s) - p * q / (r * s)$;

б) $1E3 + beta / (x - gamma * delta)$;

в) $a/b * (c+d) - (a-b)/b/c + 1E-8$.

3) Для следующих формул записать соответствующие арифметические выражения на Паскале:

а) $(1+x)^2$; б) $\sqrt{1+x^2}$; в) $\cos^2 x^2$; г) $\log_2 \frac{x}{5}$;

д) $\arcsin x$; е) $\frac{e^x + e^{-x}}{2}$; ж) $x^{\sqrt{2}}$; з) $\sqrt[3]{1+x}$;

и) $\sqrt{x^8 + 8^x}$; к) $\frac{xyz - 3,3|x + \sqrt[4]{y}|}{10^7 + \ln 4!}$; л) $\frac{\beta + \sin^2 \pi^4}{\cos 2 + |\operatorname{ctg} \gamma|}$.

4) Вычислить значения выражений:

а) $\operatorname{round}(6.9)$; в) $20 \operatorname{div} 6$; д) $20 \operatorname{mod} 6$;

б) $\operatorname{round}(6.)$; г) $2 \operatorname{div} 5$; ж) $2 \operatorname{mod} 5$;

и) $3 * 7 \operatorname{div} 2 \operatorname{mod} 7 / 3 - \operatorname{trunc}(\sin(1))$.

ПРАКТИЧЕСКАЯ РАБОТА № 14

Тема: «Программное обеспечение в сфере профессиональной деятельности»

Цель: разобрать структуру программ в общем виде; сформировать навыки объявления переменных, использование инструкции присваивания, ввода, вывода.

Оборудование: справочный материал, персональный компьютер с выходом в Интернет.

Содержание работы

Задание. Инструкция присваивания

Приступая к решению задач этого раздела, следует вспомнить, что:

- инструкция присваивания используется для изменения значений переменных, в том числе и для вычислений по формулам;
- тип выражения, находящегося в правой части инструкции присваивания, должен соответствовать типу переменной, имя которой стоит слева от символа инструкции присваивания (при нарушении соответствия типа переменной и выражения компилятор выводит сообщение об ошибке **Type miss match** — несоответствие типов).

Пример. Запишите инструкцию, которая увеличивает на единицу значение переменной n .

Решение: $n := n + 1$;

2.1. Запишите инструкцию, которая присваивает переменной summa нулевое значение.

2.2. Запишите инструкцию вычисления среднего арифметического переменных x_1 и x_2 .

2.3. Запишите в виде инструкции присваивания формулу вычисления значения функции

$$y = -2,7x^3 + 0,23x^2 - 1,4.$$

2.4. Запишите в виде инструкции присваивания формулу пересчета веса из фунтов в килограммы (один фунт равен 409,5 г).

2.5. Запишите в виде инструкции присваивания формулу пересчета расстояния из километров в версты (одна верста равна 1066,8 м).

2.6. Запишите в виде инструкции присваивания формулу вычисления площади треугольника:

$s = 1/2 ah$, где a — длина основания треугольника, h — его высота.

ПРАКТИЧЕСКАЯ РАБОТА № 15

Тема: «Программное обеспечение в сфере профессиональной деятельности»

Цель: учиться составлять простейшие программы.

Оборудование: справочный материал, персональный компьютер с выходом в Интернет.

Содержание работы

Задание. Вывод

Приступая к решению задач этого раздела, следует вспомнить, что:

- инструкции `write` и `writeln` предназначены для вывода на экран монитора сообщений и значений переменных;
- одна инструкция `write` (`writeln`) может вывести на экран значения нескольких переменных и (или) несколько сообщений;
- инструкция `writeln` без параметров переводит курсор в начало следующей строки экрана.

Пример. Написать инструкции вывода значений переменных a , b и c (тип вещественный, 2 знака после запятой). Значение каждой переменной должно быть выведено на отдельной строке.

Решение: `writeln (a:6:2);`
`writeln (b:6:2);`
`writeln (c:6:2);`

Написать программу, которая выводит на экран имя и фамилию.

Написать программу, которая выводит на экран четверостишие:

Унылая пора! Очей очарованье!
Приятна мне твоя прощальная краса —
Люблю я пышное природы увяданье,
В багрец и золото одетые леса.

А. С. Пушкин

3.3. Написать инструкцию вывода значения переменной a (тип *real*) с тремя цифрами в дробной части.

Задание. Ввод

Приступая к решению задач этого раздела, следует вспомнить, что:

- для ввода с клавиатуры во время работы программы исходных данных (значений переменных) предназначена инструкция `readln`;
- используя одну инструкцию `readln`, можно ввести значения нескольких переменных;
- тип данных, вводимых во время работы программы, должен соответствовать типу переменной, указанной в инструкции `readln`;
- в случае несоответствия типа введенных данных типу переменной, значение которой вводится с клавиатуры, программа завершает работу и на экран выводится сообщение `Error 106: Invalid numeric format` (если программа запущена из среды разработки, т. е. из Turbo Pascal) или `Run time error 106` (если программа запущена из операционной системы).

Пример. Написать инструкцию, которая обеспечивает ввод значений переменных `u` и `r`. Предполагается, что во время работы программы пользователь будет набирать числа в одной строке.

Решение: `readln (u,r);`

- 4.1. Написать инструкции, которые обеспечивают ввод значений переменных `u` и `r`. Предполагается, что во время работы программы пользователь будет после набора каждого числа нажимать клавишу `<Enter>`.
- 4.2. Объявите необходимые переменные и напишите фрагмент программы вычисления объема цилиндра, обеспечивающий ввод исходных данных.
- 4.3. Объявите необходимые переменные и напишите инструкции ввода исходных данных для программы вычисления стоимости покупки нескольких тетрадей и карандашей. Предполагается, что во время работы программы пользователь будет вводить данные о каждой составляющей покупки в отдельной строке: сначала цену, затем количество.

Информационное обеспечение обучения

Основные литература:

1. Замятина, О. М. Инфокоммуникационные системы и сети. Основы моделирования: учебное пособие для среднего профессионального образования / О. М. Замятина. — Москва: Издательство Юрайт, 2023. — 167 с. — (Профессиональное образование). — ISBN 978-5-534-17558-5. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru>
2. Гаврилов, М. В. Информатика и информационные технологии : учебник для среднего профессионального образования / М. В. Гаврилов, В. А. Климов. — 5-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 355 с. — (Профессиональное образование). — ISBN 978-5-534-15930-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/>

Дополнительные источники:

Казанский, А. А. Программирование на Visual C# : учебное пособие для среднего профессионального образования / А. А. Казанский. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 192 с. — (Профессиональное образование). — ISBN 978-5-534-14130-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/>

Учебно-методические материалы:

1. Методические указания к практическим/лабораторным работам (Электронный ресурс)/ Коровин Ю.И., Горохов Д.В., – Москва: РГАУ-МСХА, 2021 – ЭБС –«РГАУ-МСХА»

Интернет – ресурсы

1. Электронно-библиотечная система РГАУ-МСХА им. К.А. Тимирязева (далее ЭБС) сайт www.library.timacad.ru
2. Научная электронная библиотека «КиберЛенинка» - <https://cyberleninka.ru/>
3. Сетевая электронная библиотека аграрных вузов - <https://e.lanbook.com/books>