

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Локтионова Оксана Геннадьевна

Должность: проректор по учебной работе

Дата подписания: 26.01.2021 18:25:30

Уникальный программный ключ:

0b817ca911e6668abb13a5d426d39e5f1c11eabb175e9745d14a4851fca56d089

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра «Вычислительная техника»



Информатика

Методические указания по выполнению лабораторных работ для студентов
направления подготовки 090301

Курск – 2017

УДК 621.(076.1)

Составитель: Д.Б. Борзов

Рецензент

Кандидат технических наук, доцент *Э.И. Ватутин*

Информатика: методические указания по выполнению лабораторных работ/ Юго-Зап. гос. ун-т; сост.: Д.Б. Борзов. - Курск, 2017. - 39 с.: ил. 11, табл. 2.

Содержатся методические указания к проведению лабораторных работ по дисциплине «Информатика», в которые входят правила и принципы сложения, вычитания и деления двоичных чисел с использованием кодов Д1, Д4 и двоично-десятичных чисел.

Методические рекомендации соответствуют требованиям программы, утвержденной учебно-методическим объединением по направлению подготовки «Информатика и вычислительная техника» (ИВТ) дневной и заочной форм обучения.

Текст печатается в авторской редакции

Подписано в печать . Формат 60x84 1/16.

Усл.печ. л. . Уч.-изд. л. . Тираж 100 экз. Заказ. Бесплатно.

Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

Лабораторная работа №1

Понятие информатики, информации. Характеристики информации

1. Цель лабораторной работы

Целью лабораторной работы является изучение основных понятий информатики, характеристик информации, получения информации об объеме полученного сообщения.

2. Теоретические сведения

Информатика — это основанная на использовании компьютерной техники дисциплина, изучающая структуру и общие свойства информации, а также закономерности и методы её создания, хранения, поиска, преобразования, передачи и применения в различных сферах человеческой деятельности.

Информация — сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые воспринимают информационные системы (живые организмы, управляющие машины и др.) в процессе жизнедеятельности и работы.

Одно и то же информационное сообщение (статья в газете, объявление, письмо, телеграмма, справка, рассказ, чертёж, радиопередача и т.п.) может содержать разное количество информации для разных людей — в зависимости от их предшествующих знаний, от уровня понимания этого сообщения и интереса к нему.

Например, сообщение, составленное на японском языке, не несёт никакой новой информации человеку, не знающему этого языка, но может быть высокоинформативным для человека, владеющего японским.

Информация есть характеристика не сообщения, а *соотношения между сообщением и его потребителем*. Без наличия потребителя говорить об информации бессмысленно.

Применительно к компьютерной обработке данных под информацией понимают некоторую последовательность символических обозначений (букв, цифр, закодированных графических образов и звуков и т.п.), несущую смысловую нагрузку и представленную в понятном компьютеру виде. Каждый новый символ в такой последовательности символов увеличивает информационный объём сообщения.

В каком виде существует информация?

Информация может существовать в самых разнообразных формах:

- в виде текстов, рисунков, чертежей, фотографий;
- в виде световых или звуковых сигналов;
- в виде радиоволн;
- в виде электрических и нервных импульсов;
- в виде магнитных записей;
- в виде жестов и мимики;
- в виде запахов и вкусовых ощущений;
- в виде хромосом, посредством которых передаются по наследству признаки и свойства организмов и т.д.

Как измеряется количество информации?

Американский инженер Р. Хартли процесс получения информации рассматривает как выбор одного сообщения из конечного наперёд заданного множества из N равнове-

роятных сообщений, а количество информации I , содержащееся в выбранном сообщении, определяет как двоичный логарифм N .

Формула Хартли: $I = \log_2 N$.

Допустим, нужно угадать одно число из набора чисел от единицы до ста. По формуле Хартли можно вычислить, какое количество информации для этого требуется: $I = \log_2 100 \approx 6,644$. То есть сообщение о верно угаданном числе содержит количество информации, приблизительно равное 6,644 единиц информации.

Приведем другие примеры равновероятных сообщений:

1. при бросании монеты: "выпала решка", "выпал орел";
2. на странице книги: "количество букв чётное", "количество букв нечётное".

Определим теперь, являются ли равновероятными сообщения "первой выйдет из дверей здания женщина" и "первым выйдет из дверей здания мужчина". Однозначно ответить на этот вопрос нельзя. Все зависит от того, о каком именно здании идет речь. Если это, например, станция метро, то вероятность выйти из дверей первым одинакова для мужчины и женщины, а если это военная казарма, то для мужчины эта вероятность значительно выше, чем для женщины.

Для задач такого рода американский учёный Клод Шеннон предложил в 1948 г. другую формулу определения количества информации, учитывающую возможную неодинаковую вероятность сообщений в наборе.

Формула Шеннона: $I = - (p_1 \log_2 p_1 + p_2 \log_2 p_2 + \dots + p_N \log_2 p_N)$,

где p_i — вероятность того, что именно i -е сообщение выделено в наборе из N сообщений.

Легко заметить, что если вероятности p_1, \dots, p_N равны, то каждая из них равна $1/N$, и формула Шеннона превращается в формулу Хартли.

3. Задания для выполнения

Вычислить объем полученной информации по формуле Шеннона и формуле Хартли:

Задача 1.

Сообщение, записанное буквами 64-символьного алфавита, содержит 20 символов. Какой объем информации оно несет?

Задача 2.

Жители планеты Принтер используют алфавит из 256 знаков, а жители планеты Плоттер — из 128 знаков. Для жителей какой планеты сообщение из 10 знаков несет больше информации и на сколько?

Задача 3.

Для кодирования нотной записи используется 7 значков-нот. Каждая нота кодируется одним и тем же минимально возможным количеством бит. Чему равен информационный объем сообщения, состоящего из 180 нот?

Задача 4.

Цветное растровое графическое изображение, палитра которого включает в себя 65 536 цветов, имеет размер 100X100 точек (пикселей). Какой объем видеопамати компьютера (в Кбайтах) занимает это изображение в формате BMP?

Задача 5.

В велокроссе участвуют 119 спортсменов. Специальное устройство регистрирует прохождение каждым из участников промежуточного финиша, записывая его номер с использованием минимально возможного количества бит, одинакового для каждого спортсмена. Каков информационный объем сообщения, записанного устройством, после того как промежуточный финиш прошли 70 велосипедистов?

Задача 6.

Словарный запас некоторого языка составляет 256 слов, каждое из которых состоит точно из 4 букв. Сколько букв в алфавите языка?

Задача 7.

Сколько вопросов надо задать, чтобы отгадать одну из 32 карт (колода без шестерок), если ответами могут быть лишь “да” или “нет”?

Задача 8.

Сообщение о том, что на светофоре красный сигнал, несет в себе информации больше, чем бит. Объясните почему.

Задача 9.

Пылкий влюбленный, находясь в разлуке с объектом своей любви, посылает телеграмму: “Любишь?”. В ответ приходит не менее лаконичная телеграмма: “Да!”. Сколько информации несет ответная телеграмма?

Задача 10.

Измерьте информационный объем сообщения “Ура! Закончились каникулы!!” (с точки зрения технического подхода, то есть не учитывая смысл сообщения). Выразите этот объем в битах, байтах, килобайтах.

Задача 11.

Сколько следует задать вопросов и как их следует формулировать, чтобы оценить сообщение о том, что вагон стоит на одном из 16 путей?

Задача 12.

Определите, сколько бит информации несет сообщение о том, что на светофоре горит зеленый свет.

Задача 13.

Вы бросаете два кубика с нанесенными на гранях цифрами от 1 до 6. Определите, сколько бит информации несет сообщение, что на одном кубике выпала тройка, а на другом - пятерка.

Задача 14.

Предположим, вероятность того, что вы получите за контрольную работу оценку “5”, равна 0,6; вероятность получения “4” равна 0,3; вероятность получения “3” - 0,1. Определите, сколько бит информации будет нести сообщение о результатах контрольной работы в каждом из возможных случаев.

4. Порядок выполнения работы

4.1. Изучить данные методические указания

4.2. Вычислить объем полученной информации согласно полученному варианту задания.

4.3. Подготовить отчет.

5. Содержание отчета

5.1. Тема лабораторной работы.

5.2. Цель работы.

5.3. Исходные данные для выполнения работы.

5.4. Решение задачи вычисления объема полученной информации.

5.5. Вывод

Библиографический список

1. Борзов, Д.Б. Информатика [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая; Учебное пособие.– Юго-Запад. Гос. ун-т. Курск, 2010. 303 с.
2. Борзов, Д.Б. Информатика [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая, Е.А. Титенко; Курск гос. тех. ун-т. Курск, 2007. 128 с.
3. Борзов, Д.Б. Интерфейсы периферийных устройств [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая; Курск гос. тех. ун-т. Курск, 2007. 175 с.
4. Борзов, Д.Б. Процедурно–ориентированное программирование на С++ [Текст]: Учеб. пособие / Д.Б. Борзов, И.В. Зотов, Э.И. Ватутин; Курск гос. тех. ун-т. Курск, 2007. 209 с.
5. Борзов, Д.Б. Интерфейсы периферийных устройств [Текст]: учеб. пособие для вузов / Д.Б. Борзов; Учебное пособие.– Юго-Запад. Гос. ун-т. Курск, 2015. 256 с.

Лабораторная работа №2

Равномерные и неравномерные коды

1. Целью лабораторной работы является изучение основных понятий о равномерных и неравномерных кодах, их характеристик, правилах и способах использования и применении.

2. Теоретические сведения

Объектом передачи и преобразования в ЭВМ является дискретная информация. Для представления её применяется так называемый алфавитный способ, основой которого является использование фиксированного конечного набора символов любой природы, называемого алфавитом. Символы из набора алфавита называются *буквами*, а любая конечная последовательность букв – *словом*.

В процессе ввода, хранения, вывода и обработки информации в ЭВМ осуществляется её преобразование из одной формы в другую. При этом с каждой из форм представления связаны различные алфавиты. Таким образом, процесс преобразования информации часто требует представлять буквы одного алфавита средствами (буквами, словами) другого алфавита. Такое представление называется *кодированием*.

Декодированием называется процесс обратного преобразования информации относительно ранее выполненного кодирования.

Для представления информации в ЭВМ преимущественное распространение получило *двоичное кодирование*, при котором символы вводимой в ЭВМ информации представляются средствами *двоичного алфавита*, состоящего из двух букв. В дальнейшем в качестве этих букв будут использоваться символы 0 и 1.

Двоичный алфавит по числу входящих в него символов является минимальным. Полный набор кодовых комбинаций, соответствующих двоичному представлению всех букв кодируемого алфавита, называется *кодом*.

Различают коды **равномерные** и **неравномерные**. Кодовые комбинации равномерных двоичных кодов содержат одинаковое число двоичных знаков, неравномерных – не одинаковое.

Примером неравномерного двоичного кода может служить азбука Морзе, в которой для каждой буквы и цифры определена двоичная последовательность коротких и длинных сигналов. В азбуке Морзе буква Е, например, соответствует один короткий сигнал (точка), а букве Ш – четыре длинных сигнала (четыре тире). Неравномерное кодирование позволяет повысить скорость передачи сообщений за счёт того, что наиболее часто встречающимся в передаваемых текстах символам (к ним относится и буква Е) назначается более короткая комбинация. В технике наибольшее применение нашли равномерные коды, как более удобные для реализации.

3. Задание.

Привести примеры равномерных и неравномерных кодов и закодировать задачи 1 – 14 из лабораторной работы №1

4. Порядок выполнения работы

4.1. Изучить данные методические указания

4.2. Согласно полученному варианту задания представить число в виде равномерного и неравномерного кода.

4.3. Подготовить отчет.

5. Содержание отчета

5.1. Тема лабораторной работы.

5.2. Цель работы.

5.3. Исходные данные для выполнения работы.

5.4. Решение задачи представления чисел в виде равномерного и неравномерного кода.

5.5. Вывод

Библиографический список

1. Борзов, Д.Б. Информатика [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая; Учебное пособие.– Юго-Запад. Гос. ун-т. Курск, 2010. 303 с.
2. Борзов, Д.Б. Информатика [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая, Е.А. Титенко; Курск гос. тех. ун-т. Курск, 2007. 128 с.
3. Борзов, Д.Б. Интерфейсы периферийных устройств [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая; Курск гос. тех. ун-т. Курск, 2007. 175 с.
4. Борзов, Д.Б. Процедурно–ориентированное программирование на С++ [Текст]: Учеб. пособие / Д.Б. Борзов, И.В. Зотов, Э.И. Ватутин; Курск гос. тех. ун-т. Курск, 2007. 209 с.
5. Борзов, Д.Б. Интерфейсы периферийных устройств [Текст]: учеб. пособие для вузов / Д.Б. Борзов; Учебное пособие.– Юго-Запад. Гос. ун-т. Курск, 2015. 256 с.

Лабораторная работа №3

Однородные и неоднородные системы счисления

1. Целью лабораторной работы является изучение основных понятий об однородных и неоднородных системах счисления, их характеристиках, правилах и способах использования и применении.

2. Теоретические сведения

Различают однородные и неоднородные системы счисления. В неоднородных системах счисления G_i не зависят друг от друга и могут принимать любые значения. Эти системы еще называют системами со смешанным основанием. В неоднородных системах счисления в каждом i -м разряде количество допустимых символов может быть различно, при этом $0 \leq a_i \leq p_{i-1}$, где p_i – основание системы в i -м разряде.

В общем виде число A может быть представлено следующим образом:

$$A = a_n p_{n-1} \dots p_1 + a_{n-1} p_{n-2} \dots p_1 + \dots + a_2 p_1 + a_1, \quad (1)$$

где a_i – цифра i -го разряда числа, причем $a_i = 0, p_{j-1}$ есть база системы счисления;

$$p_i = \prod_0^i p_j \quad \text{– вес } i\text{-го разряда числа}$$

Примером неоднородной системы счисления может служить система счисления времени, для которой $p_0 = 1c$;

$p_1 = 60c$; $p_2 = 60$ минут; $p_3 = 24$ часа; $p_4 = 365$ суток. Например, время в 2 года, 25 суток, 14 часов, 35 мин, 48 секунд, выраженное в единицах младшего разряда – секундах, определится по (1):

$$A = 2 \cdot 365 \cdot 24 \cdot 60 \cdot 60 \cdot 1 + 25 \cdot 24 \cdot 60 \cdot 60 \cdot 1 + 14 \cdot 60 \cdot 60 \cdot 1 + 35 \cdot 60 \cdot 1 + 48 \cdot 1.$$

Специально для применения в ЭВМ была создана неоднородная двоично-пятеричная система, в которой в нечетных разрядах основание $p_1 = 5$ ($a_i = 0 \dots 4$), а в четных разрядах основание $p_2 = 2$ ($a_i = 0,1$). Так как произведение весов двух соседних (четного и нечетного) разрядов равно десяти, то двумя двоично-десятичными разрядами можно кодировать одну десятичную цифру (таб.).

$a_{(10)}$	$a_{(2-5)}$
0	00
1	01
2	02
3	03
4	04
5	10
6	11
7	12
8	13
9	14

Пример 3.1. Записать число $748_{(10)}$ в двоично-пятеричной системе счисления.

Решение. $A_{(10)} = 12\ 04\ 13_{(2-5)}$.

Здесь $n = 6$, основания $p_1 = 5; p_2 = 2; p_3 = 5; p_4 = 2; p_5 = 5; p_6 = 2$, а цифры $a_1 = 3; a_2 = 1; a_3 = 4; a_4 = 0; a_5 = 2; a_6 = 1$. Для вычисления количественного эквивалента числа A подставим эти значения в (1).

$$A = 1 \cdot 5 \cdot 2 \cdot 5 \cdot 2 \cdot 5 + 2 \cdot 2 \cdot 5 \cdot 2 \cdot 5 + 0 \cdot 5 \cdot 2 \cdot 5 + 4 \cdot 2 \cdot 5 + 1 \cdot 5 + 3 = 500 + 200 + 0 + 40 + 5 + 3 = 748_{(10)}.$$

В настоящее время, в основном используются позиционные системы счисления, в которых $p_i = p_j$ при всех i и j . В них веса отдельных разрядов представляют собой степень основания, равную номеру позиции.

На практике используются сокращенную запись чисел:

$$A_{(a)} = a_{n-1} \dots a_1 a_0 a_{-1} \dots a_{-m} \quad (2)$$

Так, сокращенной записи десятичного числа 86,54 соответствует его значение, вычисляемое согласно равенству (2):

$$86,54_{(10)} = 8 \cdot 10^1 + 6 \cdot 10^0 + 5 \cdot 10^{-1} + 4 \cdot 10^{-2}.$$

3. Задание.

Выразить в единицах младшего разряда – секундах время:

3 года, 56 суток, 22 часа, 70 мин, 54 секунд;

5 лет, 43 суток, 23 часа, 33 мин, 45 секунд;

6 лет, 42 суток, 21 час, 12 мин, 56 секунд;

2 года, 25 суток, 51 час, 54 мин, 12 секунд;

5 лет, 43 суток, 23 часа, 33 мин, 45 секунд;

1 год, 16 суток, 18 часов, 18 мин, 19 секунд;

7 лет, 90 суток, 71 час, 25 мин, 16 секунд;

8 лет, 11 суток, 17 часов, 28 мин, 91 секунда.

Записать десятичные числа в двоично-пятеричной системе счисления:

345;

452;

178;

918;

726;

134;

871;

899;

178;

817.

3. Порядок выполнения работы

4.1. Изучить данные методические указания

4.2. Согласно полученному варианту задания представить исходное значение для неоднородной системы счисления.

4.3. Подготовить отчет.

5. Содержание отчета

- 5.1. Тема лабораторной работы.
- 5.2. Цель работы.
- 5.3. Исходные данные для выполнения работы.
- 5.4. Решение задачи представления числа для неоднородной системы счисления.
- 5.5. Вывод

Библиографический список

1. Борзов, Д.Б. Информатика [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая; Учебное пособие.– Юго-Запад. Гос. ун-т. Курск, 2010. 303 с.
2. Борзов, Д.Б. Информатика [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая, Е.А. Титенко; Курск гос. тех. ун-т. Курск, 2007. 128 с.
3. Борзов, Д.Б. Интерфейсы периферийных устройств [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая; Курск гос. тех. ун-т. Курск, 2007. 175 с.
4. Борзов, Д.Б. Процедурно–ориентированное программирование на С++ [Текст]: Учеб. пособие / Д.Б. Борзов, И.В. Зотов, Э.И. Ватулин; Курск гос. тех. ун-т. Курск, 2007. 209 с.
5. Борзов, Д.Б. Интерфейсы периферийных устройств [Текст]: учеб. пособие для вузов / Д.Б. Борзов; Учебное пособие.– Юго-Запад. Гос. ун-т. Курск, 2015. 256 с.

Лабораторная работа №4

Перевод чисел в системах счисления

1. Целью лабораторной работы является изучение правил и способов перевода десятичного целого и дробного числа в другие системы счисления.

2. Теоретические сведения

При подготовке вычислений на ЭВМ необходимы прямые и обратные преобразования между двоичными и десятичными числами. Эти преобразования различны для целых и дробных чисел.

Пусть целое десятичное число A разложено по другому основанию ССЧ P :

$$A = a_{m-1}P^{m-1} + a_{m-2}P^{m-2} + \dots + a_1P^1 + a_0P^0 \quad (1.3)$$

если A разделить на P , то получится число $A_1 = Ap^{-1}$ и в остатке a_0 , равно значению младшего разряда числа A и записанное в новой p -ричной ССЧ.

При этом

$$A_1 = a_{m-1}P^{m-2} + a_{m-2}P^{m-3} + \dots + a_1P^0 \quad (1.4)$$

Структура A_1 такая же как A в (1.3), поэтому путем повторных преобразований можно последовательно определить значения разрядов $a_1, a_2, a_3, \dots, a_{m-1}$.

Для перевода целых чисел и целых частей неправильных дробей используется метод, базирующийся на делении переводимого числа на основание новой системы счисления. В соответствии с (1.4) целое число $A_{(p_1)}$ в системе с основанием p_2 записывается в виде:

$$A_{(p_2)} = a_{n-1} * p_2^{n-1} + a_{n-2} * p_2^{n-2} + \dots + a_0 * p_2^0$$

При $p=2$ алгоритм преобразования десятичного целого в двоичное число сводится к последовательному делению до конца десятичного числа на два и к последовательному формированию двоичных разрядов начиная с младших по значению остатков от деления.

Возможен и другой более простой способ перевода $A_{10} \rightarrow A_2$, если воспользоваться таблицей значений степеней двойки

$$2^0 = 1 \quad 2^2 = 4 \quad 2^4 = 16 \quad 2^6 = 64 \quad 2^9 = 512 \quad 2^{10} = 1024$$

$$2^1 = 2 \quad 2^3 = 8 \quad 2^5 = 32 \quad 2^7 = 128 \quad 2^8 = 256 \quad \text{и т.д.}$$

для преобразования необходимо:

- определить ближайшую от A меньшую степень двойки 2^{m-1} и присвоить разряду с весом 2^{m-1} значение «1»;
- вычесть разность $A^1 = A - 2^{m-1}$
- повторить вычисления до конца, определяя разряды со значением «1»;
- присвоить остальным разрядам значения «0».

Пример: путем подбора целых степеней перевести $A_{10} \rightarrow A_2$. $A_{10} = 293$

$$A^1 = 293 - 2^8 = 37$$

$$A^2 = 37 - 2^5 = 5$$

$$A^3 = 5 - 2^2 = 1$$

$$A^4 = 1 - 2^0 = 0$$

$$293_{10} = [2^8 + 2^5 + 2^2 + 2^0 + 2^8] = 100100101_2$$

Для перевода правильных дробей из системы счисления с основанием q_1 в систему с основанием p_2 используемый метод, базирующийся на умножении переводимой правильной дроби на основание p_2 новой системы счисления. Правильная дробь $A_{(q_1)}$ в системе с основанием p_2 может быть записана в виде

$$A_{(q_2)} = a_{-1} * p_2^{-1} + a_{-2} * p_2^{-2} + \dots + a_{-m} * p_2^{-m}$$

Если правую часть выражения умножить на p_2 , то найдем неправильную дробь, в целой части которой будет число a_{-1} . Умножив затем оставшуюся дробную часть на величину основания p_2 , получим дробь, в целой части которой будет a_{-2} и т.д. Повторяя процесс умножения m раз, найдем все m цифр дробной части числа в новой системе счисления. При этом все действия должны выполняться по правилам арифметики и, следовательно, в целой части получающихся дробей будут появляться эквиваленты цифр новой системы счисления, записанные в исходной системе счисления.

При $p=2$ алгоритм преобразования десятичного дробного в двоичное число сводится к последовательному умножению дробных частей произведений на два и к последовательному формированию двоичных разрядов начиная со старших по значению частей произведений.

Пример. $A_{10}=0.752$ перевести с точностью до 6 двоичных разрядов.

$$0.752 \cdot 2 = 1.504 \quad A_{-1} = 1$$

$$0.504 \cdot 2 = 1.008 \quad A_{-2} = 1$$

$$0.008 \cdot 2 = 0.016 \quad A_{-3} = 0$$

$$0.016 \cdot 2 = 0.032 \quad A_{-4} = 0$$

$$0.032 \cdot 2 = 0.064 \quad A_{-5} = 0$$

$$0.064 \cdot 2 = 0.128 \quad A_{-6} = 0$$

$$0.752_{10} \approx 0.110000_2.$$

При переводе правильных дробей из одной системы счисления в другую можно получить дробь в виде бесконечного числа или расходящегося ряда. Процесс перевода можно закончить, если появиться дробная часть, имеющая во всех разрядах нули, или будет достигнута заданная точность перевода (получено требуемое число разрядов результата). Последнее означает, что при переводе дроби необходимо указать число разрядов в случае ее представления в новой системе счисления. Естественно, что при этом возникает погрешность перевода чисел. В ЭВМ точность перевода обычно ограничивается длиной разрядной сетки, отведенной для представления чисел.

3. Задание

2.1. Выполнить перевод десятичных чисел в двоичную, восьмеричную и шестнадцатеричную системы счисления любым способом.

343;

832;

761;

876;

657;

891;

887;
716;
891.

2.2. Выполнить перевод десятичных дробных чисел в двоичную, восьмеричную и шестнадцатеричную системы счисления с точностью до 6 разрядов после запятой.

0,343;
0,832;
0,761;
0,876;
0,657;
0,891;
0,887;
0,716;
0,891.

3. Порядок выполнения работы

4.1. Изучить данные методические указания

4.2. Согласно полученному варианту задания представить исходное значение для неоднородной системы счисления.

4.3. Подготовить отчет.

5. Содержание отчета

5.1. Тема лабораторной работы.

5.2. Цель работы.

5.3. Исходные данные для выполнения работы.

5.4. Решение задачи представления числа для неоднородной системы счисления.

5.5. Вывод

Библиографический список

1. Борзов, Д.Б. Информатика [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая; Учебное пособие.– Юго-Запад. Гос. ун-т. Курск, 2010. 303 с.

2. Борзов, Д.Б. Информатика [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая, Е.А. Титенко; Курск гос. тех. ун-т. Курск, 2007. 128 с.

3. Борзов, Д.Б. Интерфейсы периферийных устройств [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая; Курск гос. тех. ун-т. Курск, 2007. 175 с.

4. Борзов, Д.Б. Процедурно–ориентированное программирование на С++ [Текст]: Учеб. пособие / Д.Б. Борзов, И.В. Зотов, Э.И. Ватутин; Курск гос. тех. ун-т. Курск, 2007. 209 с.

5. Борзов, Д.Б. Интерфейсы периферийных устройств [Текст]: учеб. пособие для вузов / Д.Б. Борзов; Учебное пособие.– Юго-Запад. Гос. ун-т. Курск, 2015. 256 с.

Лабораторная работа №5 Тетрады, свойства кодов Д1, Д4

1. Целью лабораторной работы является изучение основных сведений, понятий и принципов представления чисел в виде двоично-десятичных чисел и в виде кодов Д1 и Д4.

2. Теоретические сведения

Непосредственное изображение десятичных чисел приводит к необходимости двоичного кодирования десятичных цифр. Устройствам, выполняющим арифметические преобразования с десятичными числами присваивается специальный термин «десятичная арифметика». Такие устройства должны иметь максимальное сходство с обычными двоичными устройствами.

Десятичная арифметика включается в состав аппаратурных средств высокопроизводительных систем с целью исключения преобразований исходных данных в двоичную форму и результатов в десятичную.

Четырехразрядным двоичным кодом $\alpha_3^i \alpha_2^i \alpha_1^i \alpha_0^i$ тетрадами можно закодировать 16 цифр, благодаря чему появляющаяся избыточность приводит к множеству кодирования десятичных цифр, из которых следует выбирать оптимальную. Оптимальность кодирования определяется шестью требованиями, которым должен удовлетворять десятичный код.

1. *Однозначность.* Каждой десятичной цифре должен соответствовать определенный, отличающийся от других двоичный код.

Невыполнение данного требования приводит к неоднозначности результатов.

2. *Упорядоченность.* Большим десятичным цифрам должны соответствовать большие тетрады десятичного кода и, наоборот, меньшим — меньшие тетрады.

Выполнение данного требования необходимо для организации количественного сравнения цифр в десятичных разрядах.

3. *Четность.* Четным цифрам должны соответствовать четные тетрады, нечетным цифрам — нечетные тетрады. Соответствие может быть отмечено любым способом.

Выполнение данного требования необходимо для округления.

4. *Дополнительность.* Если x_1 и x_2 — такие две цифры, для которых $x_1 + x_2 = 9$ и цифре x_1 сопоставляется тетрада $\alpha_3 \alpha_2 \alpha_1 \alpha_0$, то цифре x_2 , если удовлетворяется требование дополнительной,

должна сопоставляться тетрада $\overline{\alpha_3 \alpha_2 \alpha_1 \alpha_0}$, получаемая путем инверсии двоичных разрядов кода цифры x_1

Требование дополнительной необходимо для упрощения реализации дополнительных и обратных кодов десятичных чисел.

5. *Весомозначность.* Должны существовать четыре целых положительных числа p_3, p_2, p_1, p_0 , называемых весами, с помощью которых можно определить десятичную цифру x по значению двоичной тетрады $\alpha_3 \alpha_2 \alpha_1 \alpha_0$, сопоставленной x , по формуле

$$x = \alpha_3 p_3 + \alpha_2 p_2 + \alpha_1 p_1 + \alpha_0 p_0.$$

Выполнение данного требования способствует декодированию.

6. *Непрерывность.* Непрерывной последовательности изменений значения цифр должна соответствовать непрерывная последовательность изменений значения тетрад.

Ни один из десятичных кодов не удовлетворяет одновременно всем шести перечисленным требованиям.

Наибольшее распространение в ВТ нашел код прямого замещения с весом разрядов 8421. Этот код самый наглядный и удобный, т.к. в соответствии с названием кода десятичная цифра в нем соответствующим значением двоичного кода. Однако код 8421 не удовлетворяет требованию дополненности, поэтому действия в этом коде изменению знака десятичного числа связана с инверсией разрядов или взятия дополнения, требуют дополнительных коррекций и/или временных затрат.

Д1 (8421)		Д4 (8421+3) с избытком 3	
0	0000	0	0011
1	0001	1	0100
2	0010	2	0101
3	0011	3	0110
4	0100	4	0111
5	0101	5	1000
6	0110	6	1001
7	0111	7	1010
8	1000	8	1011
9	1001	9	1100

2. Задание.

Представить в виде кодов Д1 и Д4 следующие десятичные числа:

343;

832;

761;

876;

657;

891;

887;

716;

891.

3. Порядок выполнения работы

4.1. Изучить данные методические указания

4.2. Согласно полученному варианту задания представить исходное значение для неоднородной системы счисления.

4.3. Подготовить отчет.

5. Содержание отчета

5.1. Тема лабораторной работы.

5.2. Цель работы.

5.3. Исходные данные для выполнения работы.

5.4. Решение задачи представления числа в виде кодов Д1 и Д4.

5.5. Вывод

Библиографический список

6. Борзов, Д.Б. Информатика [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая; Учебное пособие.– Юго-Запад. Гос. ун-т. Курск, 2010. 303 с.

7. Борзов, Д.Б. Информатика [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая, Е.А. Титенко; Курск гос. тех. ун-т. Курск, 2007. 128 с.

8. Борзов, Д.Б. Интерфейсы периферийных устройств [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая; Курск гос. тех. ун-т. Курск, 2007. 175 с.

9. Борзов, Д.Б. Процедурно–ориентированное программирование на С++ [Текст]: Учеб. пособие / Д.Б. Борзов, И.В. Зотов, Э.И. Ватутин; Курск гос. тех. ун-т. Курск, 2007. 209 с.

Борзов, Д.Б. Интерфейсы периферийных устройств [Текст]: учеб. пособие для вузов / Д.Б. Борзов; Учебное пособие.– Юго-Запад. Гос. ун-т. Курск, 2015. 256 с.

Лабораторная работа №5

Изображение чисел с плавающей точкой

1. Целью лабораторной работы является изучение правил и способов перевода десятичного целого и дробного числа в другие системы счисления.

2. Теоретические сведения.

Естественная форма изображения числа приводит к некоторым неудобствам. При работе с ней постоянно приходится следить за правильным и эффективным использованием разрядной сетки, чтобы числа не выходили за старший разряд и в то же время, чтобы старшие разряды использовались как значение.

Для того, чтобы автоматизировать действия по эффективному использованию всей разрядной сетки, а также значительно увеличить диапазон изображения чисел, в ЭВМ введена еще одна форма представления чисел – с плавающей точкой, или полулогарифмический формат.

Это форма основана на том, что в изображении числа введен дополнительный коэффициент X^P , т.е.

$$A_{п.т.} = X^P M$$

где $A_{п.т.}$ – полулогарифмическая форма представления числа; P – степень основания порядка или просто порядок; X – основание порядка; M – мантисса, определяющая числовые значения.

Основание порядка X для каждой ЭВМ постоянно, назначается заранее и подразумевается, а само изображение с порядками формируется из двух частей: порядка P и мантиссы M . Порядок P содержит знаковый разряд, от значения которого коэффициент X^P может быть целым или дробным. Мантисса числа всегда считается дробной, а знак мантиссы определяет знак всего числа. Полулогарифмическая форма приводит к неоднозначности изображения, так одно и то же число м.б. изображено по-разному.

Пример: $X=2$ изобразить в форме с плавающей точкой число $13_{10}=1101_2$. возможны такие изображения:

P	0100	0101	0110
M	0.110100	0.011010	0.001101

и т.д.

Основание порядка X существенно влияет на алгоритм арифметических преобразований. Для упрощения алгоритма выбирается значение $X=2^q$, где $q=1,2,3$ или 4. при таком выборе основания X сдвиг мантиссы на q разрядов, т.е. умножение на X эквивалентно изменению значения порядка на одну единицу.

В памяти компьютера мантисса представляется как целое число, содержащее только значащие цифры (0 целых и запятая не храниться). Следовательно, внутреннее представление вещественного числа сводится к представлению пары целых чисел: мантиссы и порядка.

Предположим, существует 4-х байтовая ячейка памяти. В ячейке должна содержаться следующая информация о числе:

- знак числа;
- порядок;

- значащие цифры мантииссы.

±	МАНТИССА		
1-й байт	2-й байт	3-й байт	4-й байт

В старшем бите 1-го байта храниться знак числа: 0 обозначает плюс, 1 - минус. Оставшиеся 7 бит 1-го байта содержат машинный порядок. В следующих трех байтах охраняться значащие цифры мантииссы (24 разряда).

В семи двоичных разрядах помещаются двоичные числа в диапазоне от 0000000 до 1111111. Значит, машинный порядок изменяется в диапазоне от 0 до 127 (в десятичной системе счисления). Всего 128 значений. Порядок, может быть как положительным так и отрицательным. Разумно эти 128 значений разделить поровну между положительным и отрицательным значениями порядка: от -64 до 63.

Машинный порядок смещен относительно математического и имеет только положительные значения. Смещение выбирается так, чтобы минимальному математическому значению порядка соответствовал нуль.

Связь между машинным порядком (M_p) и математическим (p) в рассматриваемом случае выражается формулой:

$$M_p = p + 64$$

Полученная формула записана в десятичной системе. В двоичной системе формула имеет вид: $M_{p2} = p_2 + 1000000_2$ Для записи внутреннего представления вещественного числа в 4-х байтовой ячейке необходимо:

- 1) перевести модуль данного числа в двоичную систему счисления с 24 значащими цифрами;
- 2) нормализовать двоичное число;
- 3) найти машинный порядок в двоичной системе счисления;
- 4) учитывая знак числа, выписать его представление в 4-х байтовом машинном слове.

Например, необходимо записать внутреннее представление числа 250,1875 в форме с плавающей точкой.

Решение:

- 1) Приведем его в двоичную систему счисления с 24 значащими цифрами:

$$250,1875_{10} = 1111\ 1010,0011\ 0000\ 0000\ 0000_2.$$

- 2) Запишем в форме нормализованного двоичного числа с плавающей точкой:

$0,1111\ 1010\ 0011\ 0000\ 0000\ 0000 * 10_2^{1000}$. Здесь мантиисса, основание системы счисления ($2_{10} = 10_2$) и порядок ($8_{10} = 1000_2$) записаны в двоичной системе.

- 3) Вычислим машинный порядок в двоичной системе счисления: $M_{p2} = 1000 + 100\ 0000 = 100\ 1000$.

- 4) Запишем представление числа в 4-х байтовой ячейке памяти с учетом знака числа:

	1001000	11111010	00110000	00000000
--	---------	----------	----------	----------

1

24

23

0

Например, необходимо по шестнадцатеричной форме внутреннего представления числа в форме с плавающей точкой C9811000 восстановить само число.

Решение:

1) Перейдем к двоичному представлению числа в 4-х байтовой ячейке, заменив каждую шестнадцатеричную цифру 4-мя двоичными цифрами:

1100 1001 1000 0001 0001 0000 0000 0000

	1001001	10000001	00010000	00000000
--	---------	----------	----------	----------

1

23

0

2) Заметим, что получен код отрицательного числа, поскольку в старшем разряде с номером 31 записана 1. Получим порядок числа: $p=1001001_2 - 1000000_2 = 1001_2 = 9_{10}$.

3) Запишем в форме нормализованного двоичного числа с плавающей точкой с учетом знака числа:

$-0,1000\ 0001\ 0001\ 0000\ 0000\ 0000\ 0000 * 2^{1001}$

4) Число в двоичной системе счисления имеет вид: $-100000010,001_2$.

5) Переведем число в десятичную систему счисления:

$-100000010,001_2 = -(1*2^8 + 1*2^1 + 1*2^{-3}) = -258,125_{10}$

3. Задание.

3.1. Записать внутреннее представление десятичных чисел в форме с плавающей точкой.

234,132;

565,816

345,392;

698,989;

812,981;

982,817;

928,918;

234,561;

823,716;

812,817;

891,981.

3.2. По шестнадцатеричной форме внутреннего представления числа в форме с плавающей точкой C9811000 восстановить само число.

F618A4;

A828B4;

C8537E;

7645C5;

7914BA;

8235Y7;

75783B;

981297;

F89273;

681ABC;

91827CE;

D8918E5;

4. Порядок выполнения работы

- 4.1. Изучить данные методические указания
- 4.2. Согласно полученному варианту задания представить исходные числа в виде чисел с плавающей точкой.
- 4.3. Подготовить отчет.

5. Содержание отчета

- 5.1. Тема лабораторной работы.
- 5.2. Цель работы.
- 5.3. Исходные данные для выполнения работы.
- 5.4. Решение задачи представления числа с плавающей точкой.
- 5.5. Вывод

Библиографический список

6. Борзов, Д.Б. Информатика [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая; Учебное пособие.– Юго-Запад. Гос. ун-т. Курск, 2010. 303 с.
7. Борзов, Д.Б. Информатика [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая, Е.А. Титенко; Курск гос. тех. ун-т. Курск, 2007. 128 с.
8. Борзов, Д.Б. Интерфейсы периферийных устройств [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая; Курск гос. тех. ун-т. Курск, 2007. 175 с.
9. Борзов, Д.Б. Процедурно–ориентированное программирование на С++ [Текст]: Учеб. пособие / Д.Б. Борзов, И.В. Зотов, Э.И. Ватутин; Курск гос. тех. ун-т. Курск, 2007. 209 с.
10. Борзов, Д.Б. Интерфейсы периферийных устройств [Текст]: учеб. пособие для вузов / Д.Б. Борзов; Учебное пособие.– Юго-Запад. Гос. ун-т. Курск, 2015. 256 с.

Лабораторная работа №7

Арифметические действия над десятичными и двоичными числами

1. Целью лабораторной работы является изучение основных сведений, понятий и принципов выполнения арифметических операций над десятичными и двоичными числами.

2. Теоретические сведения

Знаковый разряд двоичных чисел весом 2^m для целых и 2^0 для дробных чисел участвует совместно с числовыми разрядами в арифметических операциях. Знаковый разряд, также как цифровые разряды, принимает значение 1 (это знак «-») и 0 (это знак «+»).

Возможно несколько видов изображений чисел со знаком:

Изображение чисел в прямом коде (ПК)

$$\begin{array}{ll} A_{\text{ПК}}=0.0000011 & +3 \\ A_{\text{ПК}}=1.0000011 & -3 \end{array}$$

Изображение чисел обратным кодом (ОК)

Обратный код обладает определенными преимуществами. Он полностью симметричен, т.к. изображению максимального по абсолютному значению отрицательного числа соответствует $2^m + 2^0 / 2^0 + 2^{-m}$ сопоставляется изображение такого же максимального положительного числа $2^m - 2^0 / 2^0 - 2^{-m}$. Изображение положительных и отрицательных чисел взаимно дополняют друг друга, т.е. до последовательности единиц во всех двоичных разрядах, т.е. для получения ОК отрицательного числа необходимо взять инверсию всех двоичных разрядов.

В общем виде функцию изображения обратным кодом для целых чисел можно представить так:

$$A_{\text{обр}} = \begin{cases} A & \text{при } 0 \leq A \leq 2^m - 2^0, \\ 2^{m+1} - 2^0 - |A| & \text{при } -2^m + 2^0 \leq A \leq 0 \end{cases}$$

для дробных

$$A_{\text{обр}} = \begin{cases} A & \text{при } 0 \leq A \leq 2^0 - 2^{-m}, \\ 2^1 - 2^m - |A| & \text{при } -2^0 + 2^{-m} \leq A \leq 0. \end{cases}$$

пример: +37, -37

$$A_{\text{обр}}=0.0100101$$

$$A_{\text{обр}}=1.1011010$$

Изображение чисел в дополнительном коде (ДК)

Данное изображение наиболее распространенное, не требует каких-либо дополнительных аппаратных дополнений. Изображение положительных чисел равно значению самих чисел:

$$A_{\text{ПК}}=0.0000011 \quad +3$$

Изображение отрицательных чисел представляет собой дополнение до 2^{m-1} для целых чисел и до 2^1 для дробных чисел. Поэтому сумма изображений двух одинаковых

по абсолютному значению чисел разного знака равна $2^{m-1}/2^1$. В общем виде функцию изображения ДК для целых чисел можно представить следующим образом:

$$A_{ПК} = \begin{cases} A & \text{при } 0 \leq A < 2^m, \\ 2^{m+1} - |A| & \text{при } -2^m \leq A < 0 \end{cases}$$

для дробных

$$A_{ПК} = \begin{cases} A & \text{при } 0 \leq A < 2^0, \\ 2^1 - |A| & \text{при } -2^0 \leq A < 0 \end{cases}$$

Следует обратить внимание на несимметричность ДК в связи с тем, что изображению $A = -2^m / -2^0$ не сопоставляется изображение $A = +2^m / +2^0$. Необходимо помнить, что чем больше изображение отрицательного числа, тем меньше абсолютное значение самого изображаемого числа.

Существует два способа перевода чисел из прямого кода в дополнительный. Первый способ заключается в следующем:

1. Записать число в прямом коде.
2. Для ПК найти соответствующий обратный код.
3. К ОК прибавить к младшему разряду единицу.
4. Полученное число и будет дополнительным кодом заданного числа.

Второй способ перевода заключается в следующем:

1. Записать число в прямом коде.
2. Найти, просматривая с младших разрядов первую встретившуюся единицу и все разряды слева от нее перевести в обратный код.
3. Все разряды справа от найденной единицы, включая найденную оставить в прежнем виде.

Полученное число и будет дополнительным кодом заданного числа.

Сложение и вычитание двоичных чисел основаны на правилах этих действий в пределах одного разряда и правилах учета межразрядных переносов и займов.

сложение	Вычитание
0+0=0	0-0=0
0+1=1	заем → 0-1=1
1+0=1	1-0=1
1+1=1 → перенос	1-1=0

Перенос возникающий в i -ом разряде передается в следующий $(i+1)$ -разряд с увеличенным вдвое весом и уменьшенным вдвое значением.

Заем из $(i+1)$ -го разряда передается в i -й разряд с уменьшенным вдвое весом и увеличенным вдвое значением.

Например, рассмотрим основные операции (сложение и вычитание) с использованием ОК и ДК. Возьмем два числа 25, 31 и проведем с ними все возможные операции сложения и вычитания.

$$1) 31 + 25 = 56$$

Данный вариант самый простой и не требует никаких преобразований перед сложением, поэтому необходимо просто перевести числа в двоичную систему счисления и сложить с помощью обычной операции сложения. Получим:

$$\begin{array}{r}
 31 \rightarrow 0.0011111 \\
 25 \rightarrow 0.0011001 \\
 \hline
 0.0111000 \rightarrow 32+16+8=56
 \end{array}$$

$$2) 31+(-25)=-6$$

В данном случае перед сложением необходимо отрицательное число перевести предварительно в дополнительный код, после чего произвести обычную операцию сложения. Имеем:

$$\begin{array}{r}
 -25 \rightarrow -25_{\text{ДК}}=1.1100111 \\
 31 \rightarrow 0.0011111 \\
 -25_{\text{ДК}} \rightarrow 1.1100111 \\
 \hline
 0.0000110 \rightarrow +6, \text{ т.к. «+»}, \text{ то } C_{\text{доп}}=C_{\text{ПК}}
 \end{array}$$

В случае появления знака «-», результат представлен в ДК и необходим перевод в ПК для получения результата

$$3) -31+25=-6$$

В данном случае отрицательное число необходимо предварительно перевести в обратный код, после чего провести сложение. В случае получения отрицательного числа результат необходимо проинвертировать для получения ПК. $-31_{\text{ОК}}=1.1100000$ Имеем:

$$\begin{array}{r}
 -31 \rightarrow 1.1100000 \\
 25 \rightarrow 0.0011001 \\
 \hline
 1.1111001 \text{ После инверсии имеем ПК}=1.0000110
 \end{array}$$

Сумматор ОК отличается от других сумматоров тем, что в них имеет место цепь обратной связи: если есть перенос из знакового разряда, то к результату необходимо добавить «+1» в младший разряд

$$4) 31-25=+6$$

в данном случае отрицательное число необходимо перевести в обратный код. После перевода получим $-25_{\text{ОК}}=1.1100110$. Получаем:

$$\begin{array}{r}
 31 \rightarrow 0.0011111 \\
 -25_{\text{ОК}} \rightarrow 1.1100110 \\
 \hline
 \leftarrow 0.0000101 \\
 \quad \quad \quad +1 \\
 \hline
 0.0000110
 \end{array}$$

$$5) 31-(-25)=+56$$

В данном случае два отрицания превращаются в обычное сложение, и поэтому операция делать аналогично случаю 1.

$$\begin{array}{r}
 31 \rightarrow 0.0011111 \\
 25 \rightarrow 0.0011001 \\
 \hline
 0.0111000 \rightarrow 32+16+8=56
 \end{array}$$

$$6) -31-15=-56$$

В данном случае перед сложением необходимо оба числа сначала перевести в обратный код, после чего проводить операцию сложения.

$$\begin{array}{r}
 -31 \rightarrow \quad 1.1100000 \\
 -25 \rightarrow \quad 1.1100110 \\
 \hline
 \leftarrow 1.1000110 \\
 \quad \quad \quad +1 \\
 \hline
 1.0111000
 \end{array}$$

полученный результат представлен в обратном коде, поэтому для получения ПК его необходимо проинвертировать. Получим $1.0111000 \rightarrow -56$

3. Задание

Выполнить операции сложения и вычитания чисел с использованием обратного и дополнительного кода.

- 1) A = 234 B = 132;
- 2) A = 565 B = 816;
- 3) A = 345 B = 392;
- 4) A = 698 B = 989;
- 5) A = 812 B = 981;
- 6) A = 982 B = 817;
- 7) A = 928 B = 918;
- 8) A = 234 B = 561;
- 9) A = 823 B = 716;
- 10) A = 812 B = 817;
- 11) A = 891 B = 981.

4. Порядок выполнения работы

- 4.1. Изучить данные методические указания
- 4.2. Согласно полученному варианту задания выполнить операции сложения над двоичными и десятичными числами..
- 4.3. Подготовить отчет.

5. Содержание отчета

- 5.1. Тема лабораторной работы.
- 5.2. Цель работы.
- 5.3. Исходные данные для выполнения работы.
- 5.4. Решение задачи выполнения операции сложения над двоичными и десятичными числами.
- 5.5. Вывод

Библиографический список

10. Борзов, Д.Б. Информатика [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая; Учебное пособие. – Юго-Запад. Гос. ун-т. Курск, 2010. 303 с.

11. Борзов, Д.Б. Информатика [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая, Е.А. Титенко; Курск гос. тех. ун-т. Курск, 2007. 128 с.

12. Борзов, Д.Б. Интерфейсы периферийных устройств [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая; Курск гос. тех. ун-т. Курск, 2007. 175 с.

13. Борзов, Д.Б. Процедурно–ориентированное программирование на С++ [Текст]: Учеб. пособие / Д.Б. Борзов, И.В. Зотов, Э.И. Ватулин; Курск гос. тех. ун-т. Курск, 2007. 209 с.

Борзов, Д.Б. Интерфейсы периферийных устройств [Текст]: учеб. пособие для вузов / Д.Б. Борзов; Учебное пособие.– Юго-Запад. Гос. ун-т. Курск, 2015. 256 с.

Лабораторная работа №8

Сложение чисел с помощью кодов Д1 и Д4

1. Целью лабораторной работы является изучение правил и способов выполнения операции сложения чисел с помощью кодов Д1 и Д4.

2. Теоретические сведения

Так как наибольшее десятичное одноразрядное число $=9$, то с учетом переноса в данный разряд, значение результата разрядного суммирования лежит в пределах от 0 до 19. при этом единица во втором разряде представляет собой десятичный перенос следующую тетраду, а сумма получается в двоичном коде, отличном от требуемого десятичного представления, т.е. требует коррекции. В связи с этим могут возникнуть следующие случаи:

1. Если $a_i + b_i + p_i < 10$, то при выполнении действий над разрядами тетрады по правилам двоичной арифметики сразу получается правильный результат;

2. Если $a_i + b_i + p_i \geq 10$, то возникает тетрадный перенос, который следует рассматривать как десятичный. Сумму следует скорректировать на +6 (+0110), т.к. перенос передан в старший разряд с весом 16 вместо 10; признаком необходимости коррекции является выходной тетрадный перенос;

3. Если $15 \geq a_i + b_i + p_i \geq 10$ необходима такая же коррекция из-за превышения допустимого значения суммы (т.е. появляется запрещенная комбинация).

Это следующие значения:

$$10 = 1010$$

$$11 = 1011$$

$$12 = 1100$$

$$13 = 1101$$

$$14 = 1101$$

$$15 = 1111$$

Т.е. признак необходимости коррекции является единица в старшем разряде и хотя бы одной единицы в 2-х соседних кодах первоначальной суммы.

Пример: $184 + 298 = 482$ (код 8421)

0001 1000 0100

0010 1001 1000

0100 ← 0001 1100

0000 0110 0110

0100 1000 0010

На первом шаге проводим основной цикл сложения, после этого производим коррекцию в тех тетрадах, которые требуют коррекции. В данном примере в младшей тетраде присутствует запрещенная комбинация, значит ее необходимо скорректировать на +6, а из второй тетрады в третью был межтетрадный перенос, поэтому тетраду из которой был перенос необходимо также скорректировать также на число +6.

Прямое вычитание десятичных разрядов всегда меньше 10, поэтому разность необходимо скорректировать на минус шесть только при возникновении шестнадцатерично-

го займа, так как десятичный разряд приобретает в данном случае лишних шесть единиц.

Пример: $615-396=219$

0110 0001 0101

0011 1001 0110

0010→0111→1111

0000 0110 0110

0010 0001 1001

В данном случае поправки делаются для тех тетрад, в которые был перенос.

Код Д₄

$$a'_i = a_i + 3 \quad b'_i = b_i + 3$$

1. Если $a_i + b_i + p_i \leq 15$, то результат необходимо скорректировать на величину -3 (-0011)

2. Если $a_i + b_i + p_i > 15$, то при переходе в старшую тетраду меняет свой вес с 16 на 10. Поэтому требуется коррекция на величину $+3$ (0011).

Пример: $184+298=482$ (код $8421+3$)

0100 1011 0111

0101 1100 1011

1010←1000←0010

-0011 +0011 +0011

0111 1011 0101

В данном случае первые две тетрады необходимо скорректировать на $+3$, т.к. в них появилось число <15 . В старшей тетраде появился перенос, поэтому эту тетраду необходимо скорректировать на величину -3 (0011).

Пример: $-298+127=-171$

Проведем вычисления в коде Д₁ с использованием обратного кода (ОК).

Существует два способа нахождения обратного кода. В первом случае для получения обратного кода числа записанного в виде тетрад необходимо для каждой тетрады найти ее дополнение до числа 9. после этого необходимо провести сложения по правилам сложения чисел в коде прямого замещения (в коде Д₁), после получения результата необходимо снова найти дополнение каждой тетрады до числа 9 для получения необходимого результата.

I.

-298 ПК: 1. 0010 1001 1000

ОК: 1. 0111 0000 0001

В данном случае для дополнения младшей тетрады до числа 9 необходимо число 8, для дополнения второй тетрады необходим ноль, а для дополнения старшей тетрады необходимо число 7. После перевода отрицательного числа в обратный код проводим основной цикл сложения

1. 0111 0000 0001

0. 0001 0010 0111

 1. 1000 0010 1000

Получили результат, записанный в обратном коде. Для получения конечного результата находим его дополнение до числа 9. Аналогично дополняя разряды до девяти получаем:

1. 0001 0111 0001 или -171.

При этом необходимо учитывать, что если существует перенос из знакового разряда, то эту единицу прибавляем к младшей тетраде.

II.

Во втором способе последовательность действий следующая. Записываем исходное число в виде тетрад в прямом коде. После этого прибавляем к каждой тетраде число +6 (0110), после чего результат инвертируем. Это и будет обратный код исходного числа. После этого проводим обычное сложение как и в способе I, после чего также находим дополнения до числа 9.

1. 0010 1001 1000
 + 0110 0110 0110

 1 1000 1111 1110

ОК 1 0111 0000 0001

0 0001 0010 0111

 ОК 1 1000 0010 1000

ПК 1 0001 0111 0001 (-171)

Тот же пример проведем с использованием кода D_4 (8421+3) с переводом отрицательного числа в дополнительный код. Для получения дополнительного кода, записанного в коде D_4 необходимо найти дополнение всех тетрад кроме младшей до числа 15, а младшую тетраду до числа 16. После основного цикла сложения необходимо произвести коррекции. В случае возникновения межтетрадных переносов, тетрады из которых был перенос необходимо скорректировать на число +3 (0011). Остальные тетрады корректируются на число +13 (1101) с блокировкой межтетрадных переносов. После коррекции, если число отрицательное, то для него также необходимо найти дополнения как и в начале вычислений.

$-298+127=-171$

ПК 1 0101 1100 1011

ДК 1 1010 0011 0101 $\rightarrow -298_{\text{ДК}}$

ПК 0 0100 0101 1010 $\rightarrow 127$

 1 1110 1000 1111

1101 1101 1101

 ДК 1 1011 0101 1100

ПК 1 0100 1010 0100 (-127)

3. Задание

Выполнить операции сложения и вычитания в кодах Д1 и Д4.

- 1) $A = 234$ $B = 132$;
- 2) $A = 565$ $B = 816$;
- 3) $A = 345$ $B = 392$;
- 4) $A = 698$ $B = 989$;
- 5) $A = 812$ $B = 981$;
- 6) $A = 982$ $B = 817$;
- 7) $A = 928$ $B = 918$;
- 8) $A = 234$ $B = 561$;
- 9) $A = 823$ $B = 716$;
- 10) $A = 812$ $B = 817$;
- 11) $A = 891$ $B = 981$.

4. Порядок выполнения работы

- 4.1. Изучить данные методические указания
- 4.2. Согласно полученному варианту задания выполнить операции сложения и вычитания в кодах Д1 и Д4.
- 4.3. Подготовить отчет.

5. Содержание отчета

- 5.1. Тема лабораторной работы.
- 5.2. Цель работы.
- 5.3. Исходные данные для выполнения работы.
- 5.4. Решение задачи выполнения операции сложения и вычитания в кодах Д1 и Д4.
- 5.5. Вывод

Библиографический список

11. Борзов, Д.Б. Информатика [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая; Учебное пособие.– Юго-Запад. Гос. ун-т. Курск, 2010. 303 с.
12. Борзов, Д.Б. Информатика [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая, Е.А. Титенко; Курск гос. тех. ун-т. Курск, 2007. 128 с.
13. Борзов, Д.Б. Интерфейсы периферийных устройств [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая; Курск гос. тех. ун-т. Курск, 2007. 175 с.
14. Борзов, Д.Б. Процедурно–ориентированное программирование на C++ [Текст]: Учеб. пособие / Д.Б. Борзов, И.В. Зотов, Э.И. Ватутин; Курск гос. тех. ун-т. Курск, 2007. 209 с.
15. Борзов, Д.Б. Интерфейсы периферийных устройств [Текст]: учеб. пособие для вузов / Д.Б. Борзов; Учебное пособие.– Юго-Запад. Гос. ун-т. Курск, 2015. 256 с.

Лабораторная работа №9

Выполнение операции умножения в двоичной системе счисления

1. Целью лабораторной работы является изучение основных сведений, понятий и принципов выполнения арифметической операции умножения в двоичной системе счисления.

2. Теоретические сведения

При точном умножении двух чисел количество значащих цифр произведения может в пределе достичь двойного количества значащих цифр сомножителей. Правила приближенных вычислений рекомендуют оставлять в произведении столько же значащих цифр, сколько их содержится в наименее точном из сомножителей.

Наиболее просто операция умножения выполняется в ПК. При этом на первом этапе определяется знак произведения путем сложения знаковых цифр сомножителей по модулю два.

Затем производится перемножение модулей сомножителей по правилам арифметики согласно двоичной таблице умножения. Результату присваивается полученный ранее знак. Т.к. умножение производится в двоичной системе счисления частные произведения либо равны 0, либо самому сомножителю, сдвинутому на соответствующее количество разрядов.

Управления процессом умножения может начинаться как с младших разрядов множителя, так и со старших. При этом полную сумму можно получить двумя путями:

1. Сдвигом множителя на требуемое количество разрядов и добавлением полученной очередной суммы частичных произведений (СЧП) к ранее накопленной сумме.
2. Сдвигом сумм ранее полученных СЧП на каждом шаге на один разряд и последующим добавлением к сдвинутой сумме неподвижного множителя, либо 0.

Таким образом, существует четыре варианта схем машинного умножения.

- 1) Умножение младшими разрядами множителя со сдвигом СЧП вправо.

Процесс умножения сводится к n-кратному выполнению цикла:

$$C_{i+1} = (C_i + Ab_{n-i})2^{-1} \text{ при начальных условиях } i=0; C_0=0$$

В каждом цикле множимое либо добавляется к СЧП (если $b_i=1$), либо нет (если $b_i=0$), после этого сумма умножается на 2^{-1} , т.е. сдвигается на один разряд вправо. После окончания n-го цикла образуется искомое произведение, то есть $C_n=C=AB$.

Очередную цифру множителя, управляющую суммированием частичных произведений, удобнее всего снимать с младшего разряда регистра множителя, в котором в каждом цикле производится сдвиг содержимого на один разряд вправо.

Например, выполнить умножение 12 на 13.

0.0001100 множимое

0.0001101 множитель

0.0000000 СЧП

0|0001100

0000000

0001100 1 шаг

00001100 сдвиг

000001100 2 шаг + 0 → сдвиг
 0001100
 000111100 3 шаг + A
 0000111100 сдвиг
 0001100
 0010011100 4 шаг + A
 00010011100 сдвиг
 00001001100 + 0 → сдвиг
 000001001100 + 0 → сдвиг
 0000001001100 + 0 → сдвиг
 128+16+8+4 = 156

2) Умножение младшими разрядами множителя со сдвигом множимого влево при неподвижном СЧП

Вычисление этого выражения сводится к n -кратному выполнению цикла $C_{i+1} = C_i + A_i b_{n-i}$, где $A_i = 2A_{i-1}$ при начальных значениях $i=0$, $C_0=0$, $A_0=A$.

В каждом цикле умножения множимое сдвигается на один разряд влево и либо прибавляется к СЧП (при $b_i=1$), либо нет (при $b_i=0$).

Например, выполним вычисление $12 \times 13 = 156$:

0.0001100 множимое
 0.0001101 множитель
 0.0000000 СЧП

 0001100
 0000000 1 шаг СЧП + A
 0001100 СЧП
 00011000 сдвиг A на один разряд
 000110000 сдвиг A на один разряд ($b_i=0$)
 0001100 СЧП
 000111100 СЧП
 0001100000 сдвиг A на один разряд
 0010011100 СЧП+A ($b_i=1$)
 0001100000 сдвиг A на один разряд
 00011000000 сдвиг A на один разряд ($b_i=0$)
 000110000000 сдвиг A на один разряд ($b_i=0$)
 0001100000000 сдвиг A на один разряд ($b_i=0$)
 В результате имеем $128+16+8+4 = 156$.

3) Умножение старшими разрядами множителя со сдвигом СЧП влево при неподвижном множимом

умножение сводится к n -кратному повторению цикла $C_{i+1} = (C_i + Ab_{i+1})2, i = 0, C_0 = 0$.

Тогда управление умножением будет производиться цифрами множителя, начиная со старших разрядов. СЧП в каждом цикле будет сдвигаться на один разряд влево.

Например, выполним вычисления $12 \times 13 = 156$:


```

0.0001100 множимое
0.0001101 множитель
0000000 СЧП
00000000 сдвиг СЧП влево (bi=0)
000000000 сдвиг СЧП влево (bi=0)
0000000000 сдвиг СЧП влево (bi=0)
0001100 + A, т.к. (bi=1)
0000001100 СЧП
00000011000 сдвиг СЧП влево
0001100 + A, т.к. (bi=1)
00000100100
000001001000 сдвиг СЧП влево
0000010010000 сдвиг СЧП влево (bi=0)
0001100 + A, т.к. (bi=1)
0000010011100 СЧП
Получаем результат 128+16+8+4 = 156.

```

4) Умножение старшими разрядами множителя со сдвигом множимого вправо при неподвижном СЧП

Вычисление произведения может быть сведено к n -кратному выполнению цикла:

$$A_{i+1} = A_i 2^{-1};$$

$$C_{i+1} = C_i + A_{i+1} b_{i+1}, \quad i = 0, A_0 = A, C_0 = 0,$$

то есть в каждом цикле множимое сдвигается на один разряд вправо и в зависимости от значения управляющего разряда множителя либо прибавляется к СЧП, либо нет.

Как и в предыдущих случаях, выполним вычисление $12 \times 13 = 156$. После выполнения преобразований в двоичную систему счисления получаем:

```

0001100 множимое
0001101 множитель
0000000 СЧП
00001100 сдвиг A на разряд вправо, т.к. разряд bi=0
000001100 сдвиг A на разряд вправо, т.к. разряд bi=0
0000001100 сдвиг A на разряд вправо, т.к. разряд bi=0
0000000 + СЧП
0000001100 СЧП
00000001100 сдвиг A на разряд вправо
0000001100 СЧП
00000100100 СЧП
000000001100 сдвиг A на разряд вправо
0000000001100 сдвиг A на вправо, т.к. разряд bi=0
00000100100
0000010011100 СЧП → 156.

```

3. Задание

Выполнить операцию умножения двух четырьмя способами

- 1) $A = 34$ $B = 32$;
- 2) $A = 65$ $B = 16$;
- 3) $A = 45$ $B = 92$;
- 4) $A = 98$ $B = 89$;
- 5) $A = 12$ $B = 81$;
- 6) $A = 82$ $B = 17$;
- 7) $A = 28$ $B = 18$;
- 8) $A = 34$ $B = 61$;
- 9) $A = 23$ $B = 16$;
- 10) $A = 12$ $B = 17$;
- 11) $A = 91$ $B = 81$.

4. Порядок выполнения работы

- 4.1. Изучить данные методические указания
- 4.2. Согласно полученному варианту задания выполнить операцию умножения над десятичными числами в двоичной системе счисления.
- 4.3. Подготовить отчет.

5. Содержание отчета

- 5.1. Тема лабораторной работы.
- 5.2. Цель работы.
- 5.3. Исходные данные для выполнения работы.
- 5.4. Решение задачи выполнения операции умножения над десятичными числами в двоичной системе счисления.
- 5.5. Вывод

Библиографический список

14. Борзов, Д.Б. Информатика [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая; Учебное пособие.– Юго-Запад. Гос. ун-т. Курск, 2010. 303 с.
15. Борзов, Д.Б. Информатика [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая, Е.А. Титенко; Курск гос. тех. ун-т. Курск, 2007. 128 с.
16. Борзов, Д.Б. Интерфейсы периферийных устройств [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая; Курск гос. тех. ун-т. Курск, 2007. 175 с.
17. Борзов, Д.Б. Процедурно–ориентированное программирование на C++ [Текст]: Учеб. пособие / Д.Б. Борзов, И.В. Зотов, Э.И. Ватутин; Курск гос. тех. ун-т. Курск, 2007. 209 с.
- Борзов, Д.Б. Интерфейсы периферийных устройств [Текст]: учеб. пособие для вузов / Д.Б. Борзов; Учебное пособие.– Юго-Запад. Гос. ун-т. Курск, 2015. 256 с.

Лабораторная работа №10

Методы деления с восстановлением и без восстановления остатков

1. Целью лабораторной работы является изучение правил и способов выполнения операции деления двоичных чисел по правилам с восстановлением и без восстановления остатков.

2. Теоретические сведения

Наиболее очевидный алгоритм носит название алгоритма *деления с неподвижным делителем и восстановлением остатка*. Он рассматривается в силу того, что он очень похож на общепринятый способ деления столбиком. Данный алгоритм может быть описан следующим образом:

1) исходное значение частичного остатка полагается равным старшим разрядам делимого;

2) частичный остаток удваивается путем сдвига на один разряд влево. При этом освобождающийся при сдвиге младший разряд ЧО заносится очередная цифра частного.

3) из сдвинутого ЧО вычитается делитель и анализируется знак результата вычитания;

4) очередная цифра модуля частного равна единице, когда результат вычитания положителен, и нулю, если отрицателен. В последнем случае значение остатка восстанавливается до того значения, которое было до вычитания;

5) пункты 2-4 последовательно выполняются для получения всех цифр модуля частного.

Например, рассмотрим процесс деления с восстановлением остатка числа 41 на 7.

Делимое	0 0 1 0 1 0 0 1	0 1 1 1	Делитель 7_{10}
Исходное значение ЧО	0 0 1 0 1 0 0 1	0 1 0 1	Частное 5_{10}
Сдвиг ЧО влево	0 0 1 0 1 0 0 1 0		
Вычитание делителя	- 0 1 1 1		
Результат < 0	1 1 1 1 0 0 0 1 0		
Восстановление ЧО	+ 0 1 1 1		
Восстановленный ЧО	0 0 1 0 1 0 0 1 0		
Сдвиг ЧО влево	0 0 1 0 1 0 0 1 0 0		
Вычитание делителя	- 0 1 1 1		
Результат > 0	0 0 0 0 1 1 0 1 0 0		
Сдвиг ЧО влево	0 0 0 0 1 1 0 1 0 0 0		
Вычитание делителя	- 0 1 1 1		
Результат < 0	1 1 1 1 1 1 1 1 0 0 0		
Восстановленный ЧО	+ 0 1 1 1		
Восстановленный ЧО	0 0 0 0 1 1 0 1 0 0 0		
Сдвиг ЧО влево	0 0 0 0 1 1 0 1 0 0 0 0		
Вычитание делителя	- 0 1 1 1		
Результат > 0	0 0 0 0 0 1 1 0 0 1 0 0		
	0 1 1 0		Остаток 6_{10}

В данном случае результат деления будет $41:7 = 5\frac{6}{7}$

Недостаток алгоритма деления с восстановлением остатков заключается в необходимости выполнения на отдельных шагах дополнительных операций сложения для восстановления частичного остатка. Это увеличивает время выполнения деления, которое в этом случае может меняться в зависимости от конкретного сочетания кодов операндов. В силу указанных причин реальные делители строятся на основе алгоритма деления с неподвижным делителем без восстановления остатка. Приведем описание этого алгоритма:

1) исходное значение частичного остатка полагается равным старшим разрядам делимого;

2) частичный остаток удваивается путем сдвига на один разряд влево, при этом в освобождающийся при сдвиге младший разряд ЧО заносится очередная цифра частного;

3) из сдвинутого частичного остатка вычитается делитель, если остаток положителен, и к сдвинутому частичному остатку прибавляется делитель, если остаток отрицательный;

4) очередная цифра модуля частного равна единице, когда результат вычитания положителен, и нулю, если он отрицателен;

5) пп. 2-4 последовательно выполняются для получения всех цифр модуля частного.

Рассмотрим деление без восстановления остатка числа 41 на 7:

	Делимое 41_{10}	0 0 1 0 1 0 0 1	0 1 1 1	Делитель 7_{10}
Исходное значение ЧО		0 0 1 0 1 0 0 1	0 1 0 1	Частное 5_{10}
Сдвиг ЧО влево		0 0 1 0 1 0 0 1 0	↑	
Вычитание делителя		- 0 1 1 1	↑	
Результат < 0		(1) 1 1 1 0 0 0 1 0	↑	
Сдвиг ЧО влево		1 1 1 1 0 0 0 1 0	↑	
Прибавление делителя		+ 0 1 1 1	↑	
Результат > 0		(0) 0 0 0 1 1 0 1 0 0	↑	
Сдвиг ЧО влево		0 0 0 0 1 1 0 1 0 0 0	↑	
Вычитание делителя		- 0 1 1 1	↑	
Результат < 0		(1) 1 1 1 1 1 1 1 0 0 0	↑	
Сдвиг ЧО влево		1 1 1 1 1 1 1 1 0 0 0 0	↑	
Прибавление делителя		+ 0 1 1 1	↑	
		(0) 0 0 0 0 1 1 0 0 1 0 0	↑	
		0 1 1 0		Остаток 6_{10}

В результате получаем $41:7 = 5\frac{6}{7}$

2. Задание

Выполнить операцию деления чисел с восстановлением и без восстановления остатков.

1) $A = 34$ $B = 32$;

2) $A = 65$ $B = 16$;

- 3) A=45 B=92;
- 4) A=98 B=89;
- 5) A=12 B=81;
- 6) A=82 B=17;
- 7) A=28 B=18;
- 8) A= 34 B=61;
- 9) A=23 B=16;
- 10) A=12 B=17;
- 11) A=91 B=81.

4. Порядок выполнения работы

- 4.1. Изучить данные методические указания
- 4.2. Согласно полученному варианту задания выполнить операцию умножения над десятичными числами в двоичной систем счисления.
- 4.3. Подготовить отчет.

5. Содержание отчета

- 5.1. Тема лабораторной работы.
- 5.2. Цель работы.
- 5.3. Исходные данные для выполнения работы.
- 5.4. Решение задачи выполнения операции умножения над десятичными числами в двоичной систем счисления.
- 5.5. Вывод

Библиографический список

- 18. Борзов, Д.Б. Информатика [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая; Учебное пособие.– Юго-Запад. Гос. ун-т. Курск, 2010. 303 с.
- 19. Борзов, Д.Б. Информатика [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая, Е.А. Титенко; Курск гос. тех. ун-т. Курск, 2007. 128 с.
- 20. Борзов, Д.Б. Интерфейсы периферийных устройств [Текст]: учеб. пособие для вузов / Д.Б. Борзов, И.Е. Чернецкая; Курск гос. тех. ун-т. Курск, 2007. 175 с.
- 21. Борзов, Д.Б. Процедурно–ориентированное программирование на C++ [Текст]: Учеб. пособие / Д.Б. Борзов, И.В. Зотов, Э.И. Ватутин; Курск гос. тех. ун-т. Курск, 2007. 209 с.
- Борзов, Д.Б. Интерфейсы периферийных устройств [Текст]: учеб. пособие для вузов / Д.Б. Борзов; Учебное пособие.– Юго-Запад. Гос. ун-т. Курск, 2015. 256 с.

Лабораторная работа №11

Основы сетей, организации ЭВМ, булевой алгебры, способы хранения информации в ЭВМ, смежные вопросы

1. Теоретические сведения

Карта Карно может быть составлена для любого кол-ва переменных, однако удобно работать при кол-ве переменных не более пяти. По сути Карта Карно — это таблица истинности составленная в 2-х мерном виде. В ней верхняя строка является соседней с нижней, а правый столбец соседний с левым, т.о. вся Карта Карно сворачивается в фигуру тор (бублик). На пересечении строки и столбца проставляется соответствующее значение из таблицы истинности. После того как Карта заполнена можно приступать к минимизации.

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 находится 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 объединятся 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 тоже самое что и для первой, и т. д. для всех областей. Конъюнкции областей объединяем дизъюнкцией.

Для 3-х переменных сама карта выглядит следующим образом

bc	00	01	11	10
a				
0	0	1	3	2
1	4	5	7	6

В карте Карно $y(a, b, c) = \langle 0, 3, 2, 5, 7 \rangle$ – константа 1

1 фаза – заполнение клеток

2 фаза – поиск минимального числа покрытий максимального ранга, покрывающих все 1 заданной функции. Соседними могут быть 2 единицы, 4 единицы, 8 единиц, но клетки должны быть соседними и число соседей должно быть кратно степени двойки.

bc	00	01	11	10
a				
0	1	0	1	1
1	0	1	1	0

$$y_{\min} = ac \vee \overline{ab} \vee \overline{ac}$$

для 4-х переменных $\langle a, b, c, d \rangle = y(a, b, c, d) = \langle 2, 3, 4, 5, 7, 8, 9, 10, 11, 14 \rangle$

2. Задание

Выполнить минимизацию функции от четырех переменных с использованием карты Карно.

$$y(a, b, c, d) = \langle 1, 3, 4, 6, 8, 9, 12, 15 \rangle$$

$$y(a, b, c, d) = \langle 1, 4, 5, 7, 10, 14, 15 \rangle$$

$$y(a, b, c, d) = \langle 1, 2, 3, 4, 5, 10, 14 \rangle$$

$$y(a, b, c, d) = \langle 3, 4, 6, 7, 10, 12, 15 \rangle$$

$$y(a, b, c, d) = \langle 1, 5, 6, 7, 11, 12, 14 \rangle$$

$$y(a, b, c, d) = \langle 3, 5, 6, 8, 9, 14, 15 \rangle$$

$$y(a, b, c, d) = \langle 2, 5, 6, 8, 10, 13, 14 \rangle$$

$$y(a, b, c, d) = \langle 4, 5, 6, 7, 8, 11, 14 \rangle$$

$$y(a, b, c, d) = \langle 2, 3, 6, 7, 8, 11, 15 \rangle$$