МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Институт цифровых технологий, электроники и физики

Кафедра вычислительной техники и электроники (ВТиЭ)

Лабораторная работа № 003

Организация памяти

Выполнил студент 595 гр.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. В. Лаптев

Проверил: к.т.н,, доцент каф. ВТиЭ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ю.Г. Скурыдин

Лабораторная работа защищена

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Барнаул 2020

**ЗАДАНИЕ №1.**

1. **Формулировка задачи**

Создать программу, которая переводит введённое с клавиатуры семеричное число в двадцатеричное с использованием массивов.

1. **Постановка задачи**

На вход программы поступает семеричное трёхзначное число (при желании разрядность можно увеличить), использоваться могут только цифры от 0 до 6. Программа выполняет последовательный перевод числа сначала в десятичную систему, а затем в двадцатеричную систему счисления, в которой используются только цифры от 0 до 9 и цифры от A до J. На выход программы поступает трёхразрядное двадцатеричное число.

1. **Математическая модель**

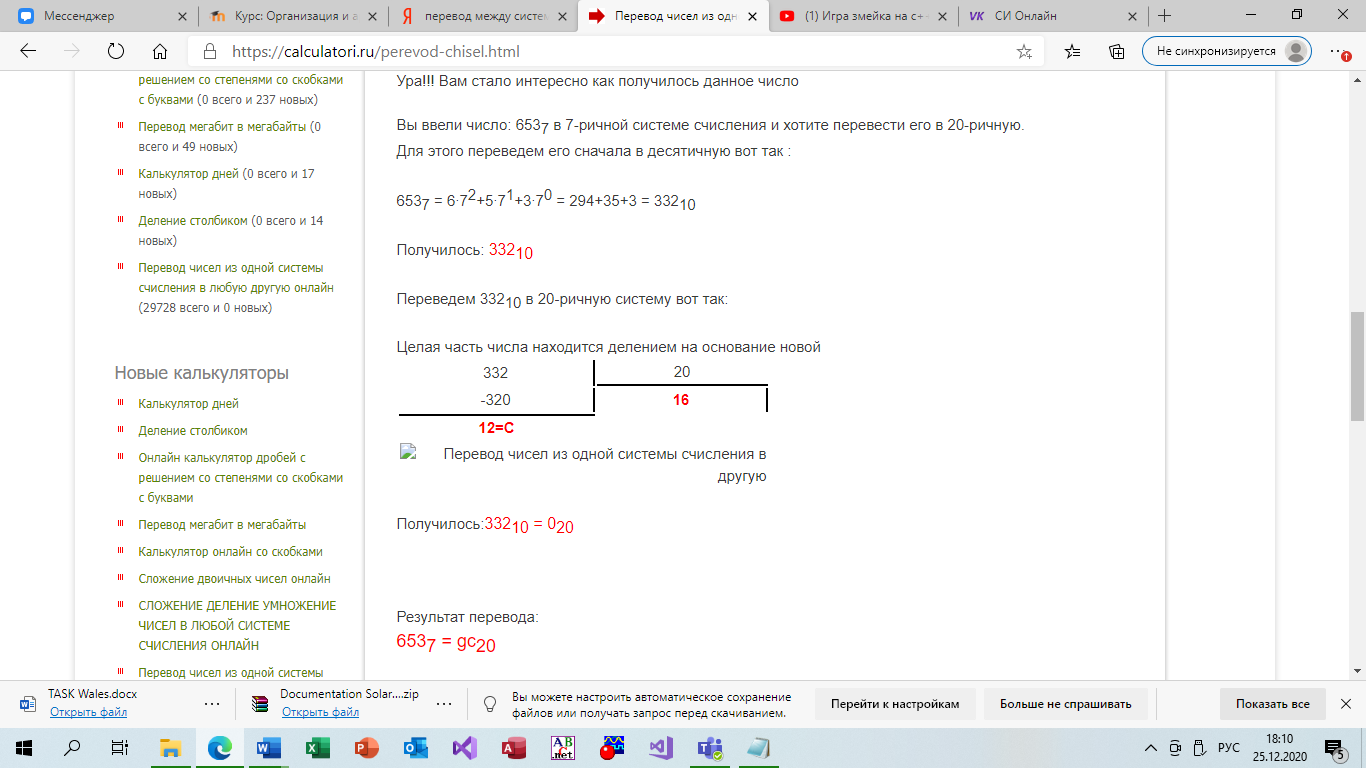
Математическую модель данной программы можно рассмотреть на примере:

На вход поступает число 6537.

Сначала число переводится в десятичную систему счисления, путём умножения всех цифр числа на основание системы счисления в степени, соответствующей разряду числа, начиная с 0 и сложения полученных произведений между собой.

6537 = 6 \* 72 + 5 \* 71 + 3 \* 70 = 33210.

Потом десятичное число переводится в двадцатеричное путём деления этого числа на основание необходимой системы счисления (20) до тех пор, пока целая часть от деления не станет меньше 20. Новое число записывается в виде остатков от деления в обратном порядке.



**=G**

В результате получаем число 0GC20.

1. **Описание алгоритма**

Начало

1. Определяем модель вычисления
2. Определяем сегмент стека и определяем его размер
3. Определяем сегмент данных

Обнуление индексного регистра

1. В si заносим значение 0

Ввод числа с клавиатуры

1. В регистр ah заносим значение 01h
2. Вызываем 21 прерывание

Заносим введённое число в массив

1. Заносим в элемент массива array7, с индексом равным значению регистра si, значение al
2. Отнимаем от элемента массива array7, с индексом равным значению регистра si, значение 30h
3. Проверка условия, при котором каждый элемент массива не должен превышать число 6. Если условие не выполняется, то переход к пункту 49

Проверка заполнения всех элементов массива

1. Сравниваем значение регистра si со значением 2, если равно, то переход к пункту 13
2. Инкрементируем значение регистра si
3. Переходим к пункту 5

Обнуление индексного регистра

1. Заносим в регистр si значение 0

Перевод в десятеричную систему счисления

1. Заносим в регистр al значение 49
2. Заносим в регистр dx значение 0
3. Заносим в регистр bl значение регистра al
4. Умножаем значение регистра al на элемент массива array7 с индексом, равным значению регистра si, результат сохраняем в регистр ax
5. Прибавляем ко значению регистра dx значение регистра ax
6. Сравниваем значение регистра si со значением 2, если равно, то переход к пункту 26
7. Инкрементируем значение регистра si
8. Заносим в регистр al значение регистра bl
9. Заносим в регистр bl значение 7
10. Заносим в регистр ah значение 0
11. Делим значение регистра ax на значение регистра bl, целая часть от деления записывается в регистр al, остаток от деления записывается в регистр ah
12. Переход к пункту 16
13. Заносим в регистр si значение 0

Перевод в двадцатеричную систему счисления

1. Заносим в регистр ax значение регистра dx
2. Заносим в регистр dl значение 20
3. Делим значение регистра ax на значение регистра dl, целая часть от деления записывается в регистр al, остаток от деления записывается в регистр ah
4. Заносим в элемент массива array20, с индексом равным значению регистра si, значение регистра ah
5. Сравниваем значение регистра si со значением 2, если равно, то переход к пункту 33
6. Инкрементируем значение регистра si
7. Заносим в регистр ah значение 0
8. Переход к пункту 29

Перевод каретки на новую строку

1. Заносим в регистр bx значение 4
2. Заносим в регистр ah значение 02h
3. Заносим в регистр dl значение 0ah
4. Выполняем 02h функцию
5. Вызываем 21 прерывание

Вывод на экран двадцатеричного числа

1. Сравниваем значение элемента массива array20, с индексом равным значению регистра si, со значением 9, если больше, то переход к пункту 47
2. Прибавляем к элементу массива array20, с индексом равным значению регистра si, значение 30h
3. Заносим в регистр dl значение элемента массива array20, с индексом равным значению регистра si
4. Заносим в регистр ah значение 02h
5. Выполняем 02h функцию
6. Вызываем 21 прерывание
7. Сравниваем значение регистра si со значением 0, если равно, то переход к пункту 49
8. Декрементируем значение регистра si
9. Переход к пункту 39

Замена чисел от 10 до 20 на буквы A - J

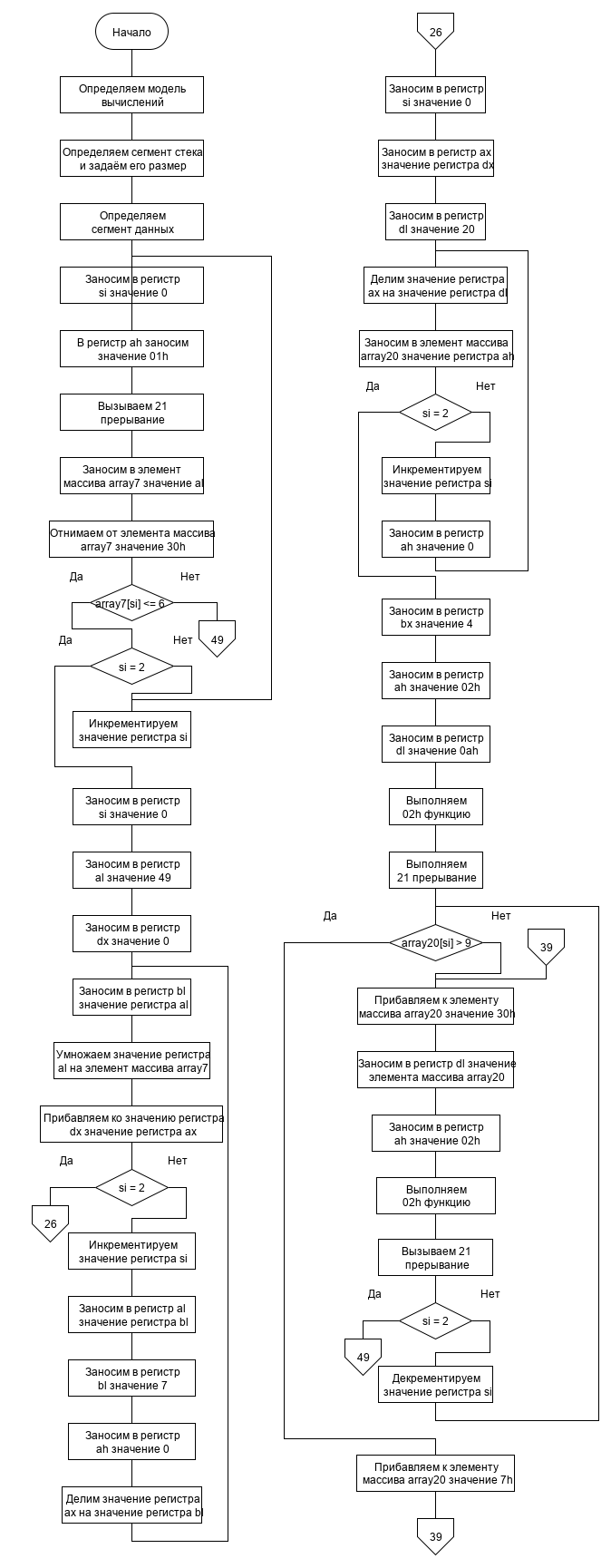
1. Прибавляем к элементу массива array20, с индексом равным значению регистра si, значение 7h
2. Переход к пункту 40

Выход из программы

1. Заносим в регистр ah значение 4ch
2. Вызываем 21h прерывание

Конец

**5. Опорный граф (блок-схема) алгоритма**

****

**6. Проект программы с определением замкнутых программных единиц и необходимых структур данных**

Для ввода с клавиатуры используем функцию 01h 21h прерывания. Для вывода на экран используем функцию 02h 21h прерывания. Для завершения работы программы используем функцию 4ch 21h прерывания. Программа разрабатывается в формате exe.

**7. Текст программы:**

**на языке Assembler:**

.386

.MODEL small

STACK 100h

data segment use16

array7 db 3 dup(0)

array20 db 3 dup(0)

data ends

text segment use16

assume CS:text, DS:data;10

begin:

mov ax, data

mov ds, ax

mov si, 0

input:

mov ah, 01h

int 21h

mov array7[si], al

sub array7[si], 30h

cmp array7[si], 6

jg exit

cmp si, 2

je ckl1

inc si

jmp input

ckl1:

mov si, 0

mov al, 49

mov dx, 0

des:

mov bl, al

mul array7[si]

add dx, ax

cmp si, 2

je ckl2

inc si

mov al, bl

mov bl, 7

mov ah, 0

div bl

jmp des

ckl2:

mov si, 0

mov ax, dx

mov dl, 20

dvad:

div dl

mov array20[si], ah

cmp si, 2

je stroka

inc si

mov ah, 0

jmp dvad

stroka:

mov bx, 4

mov ah, 02h

mov dl, 0ah

int 21h

view:

cmp array20[si], 9

ja AJ

ckl3:

add array20[si], 30h

mov dl, array20[si]

mov ah, 02h

int 21h

cmp si, 0

je exit

dec si

jmp view

AJ:

add array20[si], 7h

jmp ckl3

exit:

mov bx, 4

mov ah, 02h

mov dl, 0ah

int 21h

mov ah, 4ch

int 21h

text ends

end begin