**Лабораторная работа № 5**

***Представление чисел и определения типа оборудования***

***Цель работы.*** Идентификация оборудования и программного окружения ЭВМ, изучение представления вещественных чисел в ЭВМ.

***Методические указания.***

**1. Представление беззнаковых целых чисел**

Для представления беззнаковых целых чисел необходимо перевести из десятичной системы исчисления в двоичную. Например, число  можно представить в виде суперпозиции по степеням двойки:

,

где единицы в двоичном представлении числа стоят на позиции соответствующих степеней двоек. Для простоты рассмотрим беззнаковый однобайтовый тип данных (тип *unsigned char* в языках С/С++). Для записи числа  в такой тип данных необходимо дополнить двоичное представление до 8 знаков  и записать полученные значения в соответствующие биты:

unsigned

Рисунок 1.

**2. Представление знаковых целых чисел**

В случае знаковых типов данных старший бит отвечает за знак числа (1 – отрицательное число, 0 – положительное число). Основной проблемой является представление отрицательных чисел. Для такого представления существует следующий алгоритм:

1. нахождение двоичного представления модуля числа,
2. нахождение двоичного дополнения числа,
3. прибавление единицы.

Рассмотрим алгоритм на примере. Представим число  в знаковом однобайтном типе данных (тип *char* в языках С/С++):

1. Двоичное представление ,
2. Для нахождения двоичного представления инвертируем все биты числа ,
3. Прибавляем единицу .

После этого записываем полученные значения в соответствующие биты:

sign1

Рисунок 2.

**3. Представление вещественных чисел**

Основной интерес в вычислениях представляют вещественные типы данных и погрешности округления, связанные с ними. По стандарту IEEE 754 вещественное число  представляется в виде:

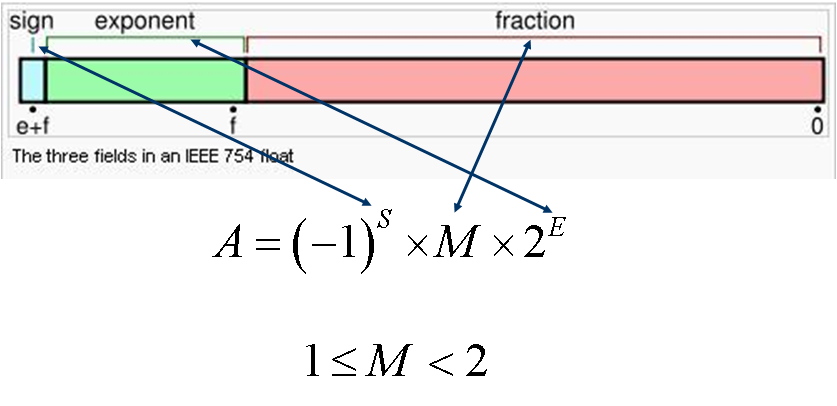


Рисунок 3.

Где  – однобитовый знак числа,  – нормализованная мантисса,  – показатель степени двойки. В случае типа *float* под мантиссу выделяется 23 бита, экспоненту 8 бит, в случае типа *double* 52 бита, экспоненту 11 бит.

Приведём пример представления вещественного числа  в типе *float*. Основной задачей является запись числа в виде . Число можно записать в виде , в данном случае мантисса имеет вид  нормализация мантиссы позволяет отбросить единицу и записывать только дробную часть. Таким образом . Далее записываем показатель степени двойки. При этом нужно учитывать, что эта степень может быть как отрицательной, так и положительной. Для этого показатель степени имеет вид:

,

где  – количество бит на показатель степени двойки. В результате число  представимо в виде:

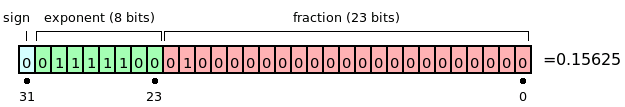


Рисунок 4.

В завершении описания представления вещественных чисел нужно отметить, что для стандартных типов данных (*float* и *double*) имеют место следующие значения:

Таблица 1.

|  |  |  |  |
| --- | --- | --- | --- |
| Тип | Минимальный порядок\* | Максимальный порядок | Число значащих знаков |
| *float* | – 45 | 38 | 7 |
| *double* | – 323 | 308 | 15 |

(\*) Стоит отметить, что мантисса может быть ненормализованной, что и приводит к таким значениям минимального порядка.

**4. Идентификация оборудования и программного окружения**

Средствами операционной системы Windows можно узнать достаточно много информации об оборудовании, памяти (функция *GlobalMemoryStatus*), жёстких дисках (функция *GetDiskFreeSpace*), сети и мониторе (функция *GetSystemMetrics* при различных параметрах), программном окружении (функции *GetComputerName* и *GetUserName*) и о многом другом. Подробное описание функций и примеры их использования можно найти в справочной системе MSDN.

Для определения таких параметров процессора, как фирма производитель, наличие расширений, количества и параметров кэшей команд и данных, TLB и других параметров в случае архитектур x86 используется инструкция процессора *cpuid*, которая имеет интерфейс на языке С/С++ *\_\_cpuid*. Так для определения идентификатора процессора имеет место следующий код:

#include <intrin.h> // подключение описания функции *\_\_cpuid*

…

int CPUInfo[4];

char CPUString[32];

\_\_cpuid(CPUInfo, 0);

memset(CPUString, 0, sizeof(CPUString));

\*((int\*)CPUString) = CPUInfo[1];

\*((int\*)(CPUString+4)) = CPUInfo[3];

\*((int\*)(CPUString+8)) = CPUInfo[2];

printf(" CPU vendor: %s\n",CPUString);

…

Первый параметр функции *\_\_cpuid* – 4-х элементный целочисленный массив, который соответствует регистрам eax, ebx, ecx, edx после выполнения инструкции. Второй параметр функции – номер функции инструкции. Подробная информация о номерах функций инструкции *cpuid* и содержимом регистров приведена в документах [1,2] для процессоров Intel и AMD. Так например с помощью функций 0x80000002, 0x80000003, 0x80000004 можно узнать полное название процессора.

1. Спецификация инструкции cpuid для процессоров Intel <http://www.intel.com/Assets/PDF/appnote/241618.pdf>
2. Спецификация инструкции cpuid для процессоров AMD <https://www.amd.com/en/support/tech-docs>

***Задание.***

1. В соответствии с вариантом задания записать представление целого числа в типе *char* и вещественного числа в типе *float*.
2. С помощью функций WinAPI определить информацию об оперативной памяти.
3. С помощью функций WinAPI определить информацию о памяти на одном из жёстких дисков.
4. С помощью инструкции cpuid определить название процессора.

***Варианты.***

1. Целое число –12, вещественное число 12.5.
2. Целое число –23, вещественное число 12.125.
3. Целое число –56, вещественное число 12.25.
4. Целое число –78, вещественное число 12.75.
5. Целое число –89, вещественное число 12.625.
6. Целое число –90, вещественное число 24.5.
7. Целое число –21, вещественное число 24.125.
8. Целое число –45, вещественное число 24.25.
9. Целое число –78, вещественное число 24.75.
10. Целое число –86, вещественное число 24.625.
11. Целое число –25, вещественное число 48.125.
12. Целое число –88, вещественное число 48.5.
13. Целое число –44, вещественное число 48.25.
14. Целое число –34, вещественное число 48.75.
15. Целое число –39, вещественное число 48.625.
16. Целое число –33, вещественное число 96.125.
17. Целое число –49, вещественное число 96.5.
18. Целое число –51, вещественное число 96.25.
19. Целое число –69, вещественное число 96.75.
20. Целое число –28, вещественное число 96.625.
21. Целое число –74, вещественное число 36.125.
22. Целое число –46, вещественное число 36.5.
23. Целое число –53, вещественное число 36.25.
24. Целое число –62, вещественное число 36.75.
25. Целое число –98, вещественное число 36.625.