

## **Практическое задание № 1. Разработка библиотеки mySimpleComputer. Оперативная память, регистр флагов, декодирование операций.**

### *Цель работы*

Изучить принципы работы оперативной памяти. Познакомиться с разрядными операциями языка Си. Разработать библиотеку mySimpleComputer, включающую функции по декодированию команд, управлению регистрами и взаимодействию с оперативной памятью.

### *Задание на лабораторную работу*

1. Прочитайте главу 4 практикума по курсу «Организация ЭВМ и систем». Изучите принципы работы разрядных операций в языке Си: как можно изменить значение указанного разряда целой переменной или получить его значение. Вспомните, как сохранять информацию в файл и считывать её оттуда в бинарном виде.
2. Разработайте функции по взаимодействию с оперативной памятью, управлению регистром флагов и кодированию/декодированию команд:
  - a. `int sc_memoryInit ()` – инициализирует оперативную память Simple Computer, задавая всем её ячейкам нулевые значения. В качестве «оперативной памяти» используется массив целых чисел, определенный статически в рамках библиотеки. Размер массива равен 100 элементам.
  - b. `int sc_memorySet (int address, int value)` – задает значение указанной ячейки памяти как value. Если адрес выходит за допустимые границы, то устанавливается флаг «выход за границы памяти» и работа функции прекращается с ошибкой;
  - c. `int sc_memoryGet (int address, int * value)` – возвращает значение указанной ячейки памяти в value. Если адрес выходит за допустимые границы, то устанавливается флаг «выход за границы памяти» и работа функции прекращается с ошибкой. Значение value в этом случае не изменяется.
  - d. `int sc_memorySave (char * filename)` – сохраняет содержимое памяти в файл в бинарном виде (используя функцию `write` или `fwrite`);
  - e. `int sc_memoryLoad (char * filename)` – загружает из указанного файла содержимое оперативной памяти (используя функцию `read` или `fread`);
  - f. `int sc_regInit (void)` – инициализирует регистр флагов нулевым значением;
  - g. `int sc_regSet (int register, int value)` – устанавливает значение указанного регистра флагов. Для номеров регистров флагов должны использоваться маски, задаваемые макросами (`#define`). Если указан недопустимый номер регистра или некорректное значение, то функция завершается с ошибкой.
  - h. `int sc_regGet (int register, int * value)` – возвращает значение указанного флага. Если указан недопустимый номер регистра, то функция завершается с ошибкой.
  - i. `int sc_commandEncode (int command, int operand, int * value)` – кодирует команду с указанным номером и операндом и помещает результат в value. Если указаны неправильные значения для команды или операнда, то функция завершается с ошибкой. В этом случае значение value не изменяется.
  - j. `int sc_commandDecode (int value, int * command, int * operand)` – декодирует значение как команду Simple Computer. Если декодирование невозможно, то устанавливается флаг «ошибочная команда» и функция завершается с ошибкой.
3. Оформите разработанные функции как статическую библиотеку. Подготовьте заголовочный файл для неё.

### *Защита лабораторной работы*

Для защиты лабораторной работы необходимо подготовить программу, демонстрирующую использование созданной библиотеки функций (сборка программы с библиотекой, использование заголовочного файла, примеры вызовов каждой функции, проверка корректности работы функций при различных входных значениях).

### *Контрольные вопросы*

1. Что такое вентиль? Какие значения он может принимать?
2. Сколько вентиляей необходимо, чтобы получить логические функции НЕ, ИЛИ-НЕ, И-НЕ, И, ИЛИ?
3. Что такое таблица истинности? Булева функция? Как они связаны между собой?
4. Как получить алгебраическую булеву функцию из таблицы истинности? И наоборот?
5. Каким образом можно синтезировать логическую схему по таблице истинности? По алгебраической формуле?
6. Что такое система счисления? Чем отличается позиционная система счисления от непозиционной?
7. Как получить качественный эквивалент числа в непозиционной системе счисления? В позиционной?
8. Как перевести числа из двоичной системы счисления в десятичную? Восьмеричную? Шестнадцатеричную? И наоборот?
9. Что такое двоично-десятичное число?
10. Как в ЭВМ представляются отрицательные числа и числа с плавающей запятой?
11. Что такое дополнительный код? Зачем он используется?
12. Как перевести десятичное число с плавающей запятой в двоичное?
13. Какие базовые типы данных используются для хранения переменных в языке СИ?
14. Что такое флаг? Зачем он используется? Каким образом можно манипулировать флагами? Что такое маска?