2.6. Разработка генератора кода

Последним этапом компиляции является генерация исполняемого кода за счет перевода ассемблерной команды в 16 формат и запись полученных цифр в файл hex формата. Для осуществления этих действий необходимо разработать соответствующие алгоритмы, выполнить программную реализацию и провести тестирование.

2.6.1. Разработка алгоритмов

Для перевода ассемблерной команды необходимо разработать алгоритм преобразования. Кроме того, для записи чисел в файл также необходимо разработать алгоритм.

Алгоритм перевода ассемблерной команды в 16 формат представлен на рисунке 21.

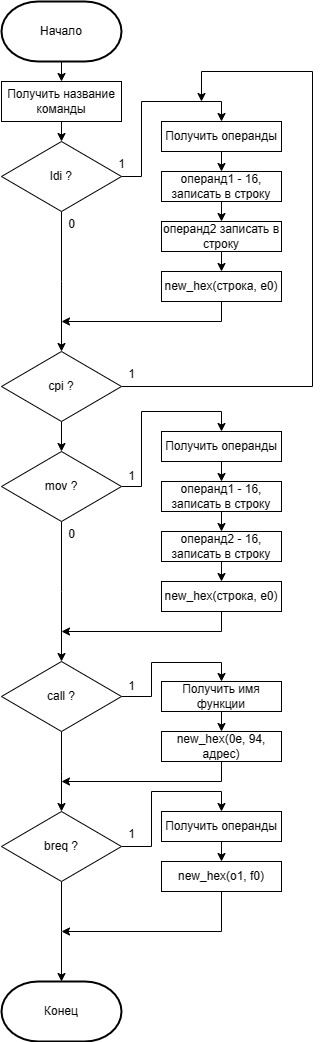


Рисунок 21 – Схема алгоритма перевода команды

Команда ldi выполняет запись числа в регистр. Для преобразования ее в 16 формат необходимо получить первый операнд, отвечающий за номер регистра, вычесть из него 16, так как команда может работать только с РОН 16-31, полученный результат записать в строку. Далее получить второй операнд, преобразовать его в строку и записать в строку с первым регистром. Это и будет 1-ой частью 16-чной команды. Вторая часть – е0 – код операции.

Аналогично для команды cpi. Единственное код операции – 30.

Что касается команды mov, то она подразумевает работу с 2мя регистрами. Следовательно, вычислять первую часть 16-чной команды необходимо вычитанием 16 из значения операндов и запись полученных значений в одну строку.

Формирование команды call происходит при помощи преобразования адреса функции. Форма для преобразования команды в 16-чный формат представлена на рисунке 22.

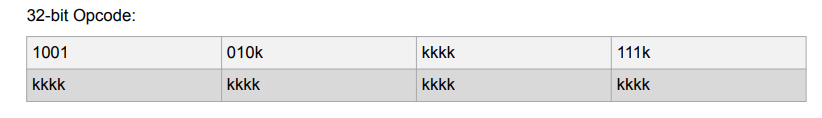


Рисунок 22 – Форма для преобразования команды call

Например, вызов функции по адресу 0х1еа запишется, как 0e 94 f5 00. Аналогично этому методу формируется команда для операции breq.

Алгоритм записи значений в файл представлен на рисунке 23.

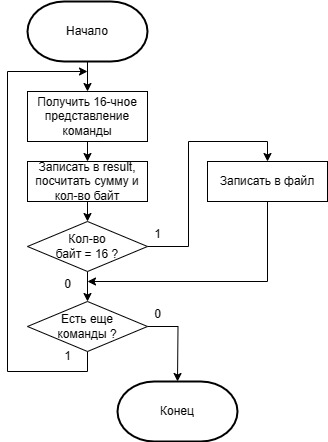


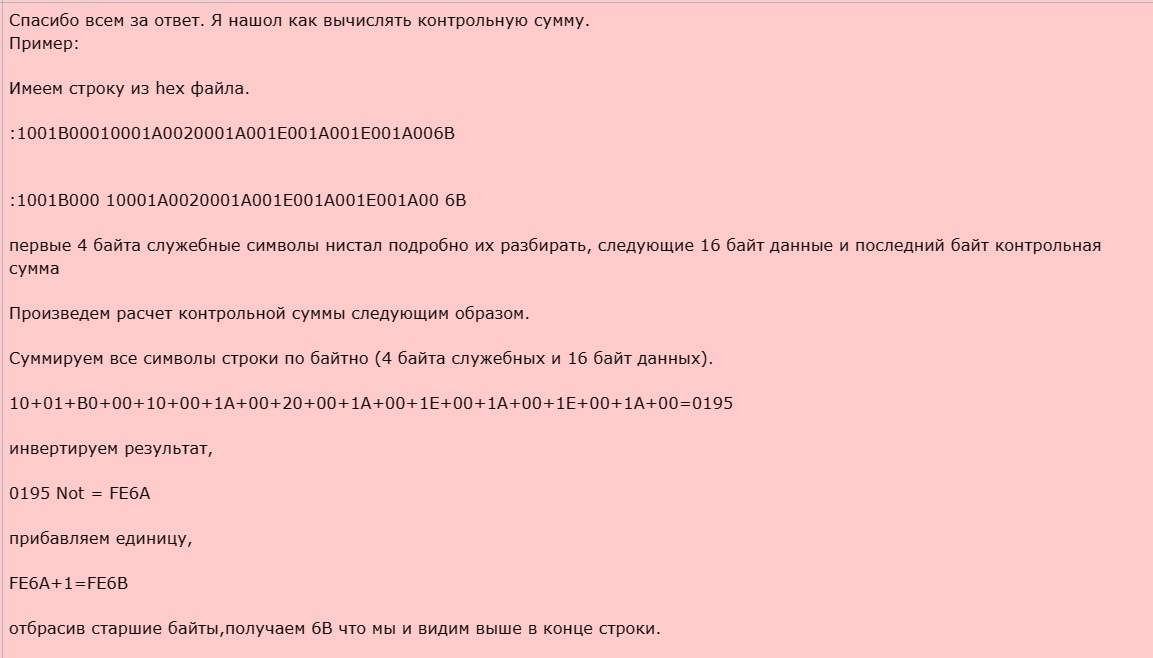
Рисунок 23 – Алгоритм записи значений в файл

Сначала получаем команду, потом записываем ее в результирующую строку. Одновременно происходит подсчет кол-ва символов (байт) и контрольной суммы. Если символов 16, то происходит запись результирующей строки в файл вместе с контрольной суммой. Далее происходит проверка на наличие команд. Если команд больше нет, то происходит запись содержимого результирующей строки и символа конца в файл.

2.6.2. Программная реализация

При разработке программы было принято решение о выполнении алгоритма генерации кода только для главной части программы, которая была разработана пользователем, так как она может иметь различную структуру. Что касается пользовательских функций, они имеют стандартное описание, которое постоянно. Следовательно, 16 код для каждой функции можно задать в программе, что и было реализовано.

Кроме того, стоит отметить, что вычисление контрольной суммы происходит следующим образом:



Для того, чтобы получить последние 2 цифры контрольной суммы она первоначально записывается в файл, потом считываются 2 последние цифры, дописываются в результирующую строку, которая потом записывается в исполняемый файл.

Листинг кода приведен в приложении Г.