Лабораторная работа №4 (использование макросредств)

Во всех задачах, где это применимо, макросы, блоки повторений и директивы условного ассемблирования должны самостоятельно определять корректность введённых данных, в том числе проверяя существование переданных в качестве фактических параметров переменных, а также не портить значения ни в каких регистрах, если не указано обратное.

1. Описать с помощью подходящих блоков повторений решение следующих задач:
   а) записать в регистр AH (изначально равный нулю) сумму чисел из регистров AL, BL, CL и DH;
   б) обнулить переменные A, B и C размером в 64 бита (массивы из 8 байт или 4 слов);
   в) зарезервировать директивой DB место в памяти для 40 байтов, задав им в качестве начальных значений первые 40 нечётных чисел.
2. Решить следующие задачи:
   а) Пусть A, B, C, D, E, F — переменные (байты или слова). Написать блок повторения, который увеличивает значение переменной на 1, если это — байтовая переменная, и на 2 – в противном случае.
   б) Пусть A, B, C, D, E, F — переменные (байты или слова). Написать блок повторения, который записывает число −1 в байтовые переменные. Другие переменные не менять.
   в) Описать в виде макроса DEF X, T, N, V определение массива X из N величин V, тип которых задается параметром T: значение параметра T=B соответствует типу byte, значение W — типу word. V может быть пустым (тогда массив не инициализируется).
3. Описать в виде макроса указанное действие над знаковыми числами размером в байт.
   а) ABSOL R, X; R:=abs(X), где R — регистр, X — переменная
   б) SUM X; AX:=сумма элементов массива X
   в) MAX X; AL: =максимум элементов массива X
   Разрешается пользоваться операторами length и size ассемблера, однако стоит учесть, что для массивов корректно они работают только в том случае, когда массив описан единичным оператором dup (например, A dw 8 dup (?) -> length A = 8, size A = 16), поэтому для их использования определить массивы необходимо так, а инициализировать программно.

4. Описать в виде макросов указанные команды, предполагая, что таких команд в ассемблере нет (PUSH и POP использовать можно, т.к. их разновидности в 8086 работают только с операндами размером в слово; также в качестве вспомогательных можно использовать регистры EAX и EBX):
   а) PUSHM X (X — переменная размером в двойное слово);
   б) POPM X (X — переменная размером в двойное слово);
   в) CALLM P (P — имя процедуры);
   г) RETM N (N — константа);
   д) LOOPM L (L — метка).
5. Решить следующие задачи:
   а) Описать в виде макроса MAX2 M, X, Y вычисление M=max(X, Y) и на его основе описать макрос MAX3 M,X,Y,Z для вычисления M=max(X, Y, Z), где M, X, Y и Z — знаковые байтовые переменные.
   б) Пусть RUN — константа, которая своим значением 1 или 0 указывает режим текущего прогона программы: счёт или отладка. Используя средства условного ассемблирования, выписать фрагмент программы, в котором с помощью алгоритма Евклида находится наибольший общий делитель двух положительных чисел с записью его в регистр AX, при условии, что в режиме счёта эти два числа должны браться из переменных в сегменте данных, а в режиме отладки эти числа равны 45 и 30.
6. Описать макрос SHIFT X, K, где X — переменная (слово), K — явно заданное число. Макрос сдвигает значение X на |K| разрядов вправо (при K>0) или влево (при K <0).
7. Описать в виде макроса OUTF без параметров вывод текущих значений флагов CF, OF, SF и ZF в виде четверки из 0 и 1, причём значения всех флагов и используемых макросом регистров должны быть сохранены.
8. Описать макрос VOLUME p1, p2, p3, p4, p5, p6, p7, который подсчитывает и печатает общее количество байтов, занимаемое переданными параметрами в сегменте данных. Параметры — имена переменных (или массивов, описанных единичным оператором dup) или константы. Учесть, что константы в памяти не хранятся.