Дан массив из 10 беззнаковых слов (2 байта). Инвертировать биты старших байтов всех элементов массива. Найти сумму четных элементов.

Код программы:

mov 0, r0 ; r0 = 0 — временная переменная (будет переопределена)

mov 0, r1 ; r1 = 0 — временная переменная

mov FF00, r2 ; r2 = FF00h — маска для XOR

mov A, r3 ; r3 = 10 (в десятичной) — длина массива (10 элементов)

mov @array, r0 ; r0 указывает на начало массива

; ===Первый проход: XOR каждого элемента с FF00h ===

pr1:

mov (r0), r1 ; r1 = значение текущего элемента массива

xor r2, r1 ; r1 = r1 XOR FF00

mov r1, (r0) ; сохраняем обратно в массив

mov (r0)+, r1 ; r0 указывает на следующий элемент (инкремент), r1 читается, но не используется

sub 1, r3 ; уменьшаем счётчик элементов

cmp 0, r3 ; если ещё есть элементы

jnz pr1 ; повторяем, если не дошли до конца

; === Второй проход: сумма всех чётных чисел до встречи значения 60 ===

mov @array, r0 ; r0 снова указывает на начало массива

pr2:

cmp 60, r0 ; если указатель r0 указывает на адрес 60 (возможно, ошибка — см. ниже)

jz end ; если да — завершить

mov (r0)+, r1 ; загружаем значение текущего элемента, r0 переходит к следующему

mov r1, r2 ; копируем значение в r2

and 1, r2 ; проверка: чётное ли число (по последнему биту)

jnz pr2 ; если нечётное — пропускаем

jz sum ; если чётное — перейти к добавлению

sum:

add r1, r3 ; r3 = r3 + r1 (суммируем чётное число)

jmp pr2 ; продолжаем

end:

stop

; === Данные ===

array:

data FF01

data FF02

data FF03

data FF04

data FF05

data FF06

data FF07

data FF08

data FF09

data FF00  
  
2+4+6+8 =20 в регистре r3