Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Отчёт по рубежной работе

Выполнил

Макогон Ярослав Вадимович

Номер группы: P3118

Проверил

Ермаков М.К.

г. Санкт-Петербург 2025г.

Содержание

Задание

Задание №1. Разработать программу для работы с элементами массива M, в которой:
1. Массив имеет следующие характеристики:
- адрес начала массива в памяти БЭВМ - 0x6cf;
- число измерений исходного массива - 1;
- количество элементов исходного массива - 17;
- каждый элемент является знаковым числом с разрядностью 9 бит;
- нумерация элементов начинается с 4;
- элементы хранятся в массиве по границам слов, нет необходимости в плотной упаковке;
2. Для элементов массива необходимо вычислить 32-х битное значение функции:
- формула функции F(Mi) = 11 * Mi + 187;
- 32-битный результат необходимо поместить в другой массив по адресу 0x400
- Результатом является массив 32-х разрядных чисел равным количеству элементов исходного массива.
Примечание: все числа представлены в десятичной системе счисления, если явно не указано иное.

Решение

- ## Изначальная программа

```
ORG 0x400

RES:

WORD 0x0
WORD 0x0

WORD 0x0
WORD 0x0

WORD 0x0
WORD 0x0

WORD 0x0
WORD 0x0

WORD 0x0
WORD 0x0

WORD 0x0
WORD 0x0

WORD 0x0
WORD 0x0

WORD 0x0
WORD 0x0

WORD 0x0
WORD 0x0

WORD 0x0
WORD 0x0

WORD 0x0
WORD 0x0

WORD 0x0
WORD 0x0

WORD 0x0
WORD 0x0

WORD 0x0
WORD 0x0

WORD 0x0
WORD 0x0

WORD 0x0
WORD 0x0
```

```
WORD 0x0
WORD 0x0 ; 421

LENGTH: WORD 0x11
CNT: WORD ?
CURIND: WORD 0x4
CUR: WORD 0x6cf
CURRES: WORD 0x400
TMP1: WORD 0x0
TMP2: WORD 0x0
TRES1: WORD 0x0
TRES2: WORD 0x0
SMASK: WORD 0x0100
FILL: WORD 0xFE00
FULL: WORD 0xFFFF
CNST: WORD 0x00BB

START:
    CLA
    LD LENGTH
    ST CNT

CYCLE:
    LD (CUR)+
    ST TMP1

    LD TMP1
    AND SMASK

    BEQ SKIP
    LD TMP1
    OR FILL
    ST TMP1
    LD FULL
    ST TMP2
    ; SIGNED 32 - TMP2_TMP1 and result in TRES2_TRES1
SKIP:
    LD TRES1
    ADD TMP1
    ST TRES1
    LD TRES2
    ADC TMP2
    ST TRES2

    LD TRES1
    ADD TMP1
    ST TRES1
    LD TRES2
    ADC TMP2
    ST TRES2

    LD TRES1
    ADD TMP1
    ST TRES1
    LD TRES2
    ADC TMP2
    ST TRES2
```

```
    LD TRES1
    ADD TMP1
    ST TRES1
    LD TRES2
    ADC TMP2
    ST TRES2

    LD TRES1
    ADD TMP1
    ST TRES1
    LD TRES2
    ADC TMP2
    ST TRES2

    LD TRES1
    ADD TMP1
    ST TRES1
    LD TRES2
    ADC TMP2
    ST TRES2

    LD TRES1
    ADD TMP1
    ST TRES1
    LD TRES2
    ADC TMP2
    ST TRES2

    LD TRES1
    ADD TMP1
    ST TRES1
    LD TRES2
    ADC TMP2
    ST TRES2

    LD TRES1
    ADD TMP1
    ST TRES1
    LD TRES2
    ADC TMP2
    ST TRES2

    LD TRES1
    ADD TMP1
    ST TRES1
    LD TRES2
    ADC TMP2
    ST TRES2

    LD TRES1
    ADD TMP1
    ST TRES1
    LD TRES2
    ADC TMP2
    ST TRES2
```

```
    ; * 11 (2)
    LD TRES1
    ADD CNST
    ST TRES1
    LD TRES2
    ADC #0x0
    ST TRES2
    ; done formula

    LD TRES1
    ST (CURRES)+
    LD TRES2
    ST (CURRES)+
    LD (CURIND)+ ; просто на всякий храню тк начинается нумерация с 4
    LD #0x0
    ST TRES1
    ST TRES2
    ST TMP1
    ST TMP2
    LOOP CNT
    JUMP CYCLE
HLT;

ORG 0x6cf
WORD 0x01FF     ; -1 => -11 + 187 = 176 (00B0)
WORD 0x1        ;  1 =>  11 + 187 = 198 (00C6)
WORD 0x01F1     ;-15 => -15*11 + 187 = 22 (0016)
WORD 0x01EC     ;-20 => -20*11 + 187 = -33 (FFFF_FFFF_FFFF_FFDF)
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0

WORD 0x0
```

Эта программа полностью работает и выполняет поставленную задачу. Подправить ее можно лишь косметически: заменить повторяющиеся блоки кода на цикл и заменить выделение памяти под результирующий массив на выделение через DUP.

Прошу заметить, что все изменения никак не влияют на логическую часть программы. И отредактированный код и исходный одинаково хорошо выполняют поставленную задачу.

- Отредактированный код

```
ORG 0x400

RES:
WORD 34 DUP(0)


LENGTH: WORD 0x11
CNT: WORD ?
CURIND: WORD 0x4
CUR: WORD 0x6cf
CURRES: WORD 0x400
TMP1: WORD 0x0
TMP2: WORD 0x0
TRES1: WORD 0x0
TRES2: WORD 0x0
SMASK: WORD 0x0100
FILL: WORD 0xFE00
FULL: WORD 0xFFFF
CNST: WORD 0x00BB

SUMCNT: WORD 0xB

START:
    CLA
    LD LENGTH
    ST CNT

CYCLE:
    LD (CUR)+
    ST TMP1

    LD TMP1
    AND SMASK

    BEQ SKIP
    LD TMP1
    OR FILL
    ST TMP1
    LD FULL
    ST TMP2
    ; SIGNED 32 - TMP2_TMP1 and result in TRES2_TRES1
SKIP:

    SUMLOOP:
        LD TRES1
        ADD TMP1
        ST TRES1
        LD TRES2
        ADC TMP2
        ST TRES2
    LOOP SUMCNT
    JUMP SUMLOOP
    LD #0xB
```

```
    ST SUMCNT

    ; * 11 (2)
    LD TRES1
    ADD CNST
    ST TRES1
    LD TRES2
    ADC #0x0
    ST TRES2
    ; done formula

    LD TRES1
    ST (CURRES)+
    LD TRES2
    ST (CURRES)+
    LD (CURIND)+ ; просто на всякий храню тк начинается нумерация с 4
    LD #0x0
    ST TRES1
    ST TRES2
    ST TMP1
    ST TMP2
    LOOP CNT
    JUMP CYCLE
HLT;

ORG 0x6cf
WORD 0x01FF     ; -1 => -11 + 187 = 176 (00B0)
WORD 0x1        ;  1 =>  11 + 187 = 198 (00C6)
WORD 0x01F1     ;-15 => -15*11 + 187 = 22 (0016)
WORD 0x01EC     ;-20 => -20*11 + 187 = -33 (FFFF_FFFF_FFFF_FFDF)
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0

WORD 0x0
```

; Желтым отметил изменения

Тесты

(1)

```
IN:
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
```

```
OUT:
00BB
0000
00BB
0000
00BB
0000
00BB
0000
00BB
0000
00BB
0000
00BB
0000
00BB
0000
00BB
0000
00BB
0000
00BB
0000
00BB
0000
00BB
0000
00BB
0000
00BB
0000
00BB
0000
00BB
0000
00BB
0000
00BB
0000
00BB
0000
```

```
; 00BB
; 0000
; x17 база
; 0000_00BB(16) = 0*11 + 187 = 187
```

(2)

```
IN:
WORD 0x01FF     ; -1 => -11 + 187 = 176 (00B0)
WORD 0x1        ;  1 =>  11 + 187 = 198 (00C6)
WORD 0x01F1     ;-15 => -15*11 + 187 = 22 (0016)
WORD 0x01EC     ;-20 => -20*11 + 187 = -33 (FFFF_FFFF_FFFF_FFDF)
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0
WORD 0x0

OUT:
00B0
0000
00C6
0000
0016
0000
FFDF
FFFF
00BB
0000
00BB
0000
00BB
0000
00BB
0000
00BB
0000
00BB
0000
00BB
0000
00BB
0000
```

```
00BB
0000
00BB
0000
00BB
0000
00BB
0000
00BB
0000
```

Запуск тестов и на исходном, и на отредактированном коде выдают одинаковый и верный результат.

Выводы

- Исходная программа корректно выполняет поставленную задачу.
- Изменения носят лишь косметический характер (красота кода) и никак не влияют на логику выполнения.