

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

**ІКНІ  
Кафедра ПЗ**

**ЗВІТ**

до лабораторної роботи № 9

**на тему:** “Складення та відлагодження циклічної програми мовою асемблера  
процесорів Cortex- M3/M4”

**з дисципліни:** “Архітектура комп’ютера”

**Лектор:**  
доц. каф. ПЗ  
Крук О. Г.

**Виконав:**  
ст. гр. ПЗ-22  
Солтисюк Д.А.

**Прийняв:**  
доц. каф. ПЗ  
Крук О. Г.

« \_\_\_\_ » \_\_\_\_\_ 2022р.  
Σ = \_\_\_\_ \_

Тема: Складення та відлагодження циклічної програми мовою асемблера процесорів Cortex- M3/M4

Мета: Ознайомитись на приладі циклічної програми з основними командами асемблера процесорів Cortex- M3/M4; розвинути навички складання програми з вкладеними циклами; відтранслювати і виконати покроково в режимі відлагодження програму, складену відповідно до свого варіанту; перевірити виконання тесту.

### Індивідуальне завдання

1. В середовищі Keil  $\mu$ Vision створіть проект Lab11. Відтранслюйте програми StartUp\_1.s та LR\_11.s (скопіюйте ці програми з методичних вказівок).
2. Виконайте компонування проекту.
3. Запустіть процес відлагодження.
4. Виконайте проект в режимі відлагодження покроково, слідкуйте за змінами регістрів.
5. В середовищі Keil  $\mu$ Vision створіть новий проект.
6. Для розміщення матриці опишіть та ініціалізуйте двовимірний масив з розмірами, відповідними до свого варіанту. Елементи матриці задавайте довільними різними дворозрядними цілими додатними або від'ємними числами. Значення елементів рядка чи стовпця, які необхідно перевіряти на виконання умови відповідно до індивідуального завдання, виберіть довільно, але вони мають бути і меншими, і рівними, і більшими за b та c.
7. Напишіть фрагмент програми для транспонування матриці ( $b_{j,i} = a_{i,j}$ ), яку збережіть в іншому масиві.
8. В програмі реалізуйте вказані у своєму варіанті операції оброблення матриці в першому масиві.
9. Всі результати розміщуйте в пам'яті (копіюйте з регістрів в пам'ять).
10. Відтранслюйте програму StartUp\_1.s та розроблену програму.
11. Виконайте компонування проекту.
12. Запустіть процес відлагодження.
13. Виконайте проект в режимі відлагодження, слідкуйте за змінами регістрів.
14. Перевірте результат роботи програми. Наведіть розгорнутий розрахунок скалярного добутку та обчислення кількості та суми елементів, що задовільняють вказаній умові.
15. Збережіть проект.
16. У звіті наведіть текст програми та копії вікон з регістрами і всіма змінними.
17. Зробіть висновки про виконану роботу.

22	(6 × 8)	1. Обчисліть скалярний добуток 5-го і 1-го стовпців. 2. Обчисліть кількість і суму елементів 4-го рядка, які задовільняють вказаній умові.	-4 9	73	$a_i < b$ або $a_i \geq c$
----	---------	--	---------	----	----------------------------

## Код програми

```
        AREA    MyCode, CODE, ReadOnly
ENTRY
EXPORT MyProg
MyProg
    ;          6*8, 5 and 1 th columns, 4th row, a < -49, a >= 73
TRANPOSE
    LDR r2, = 6; rows
    LDR r3, = 8; cols
    LDR r7, = 4; DCD size
    LDR r4, = 0; 0..cols counter - i

OUTER
    LDR r5, = 0; 0..rows counter - j

INNER
    LDR r0, = arr
    LDR r1, = res

    ;  arr[i][j] => arr[4*(i*cols)+j]
    MUL r6, r5, r3
    ADD r6, r4
    MUL r6, r6, r7
    ADD r0, r6

    ;  res[j][i] => res[4*(j*rows)+i]
    MUL r6, r4, r2
    ADD r6, r5
    MUL r6, r6, r7
    ADD r1, r6

    LDR r6, [r0]; mov [r0], [r1]
    STR r6, [r1]

    ADD r5, #1; if r5 < r2 goto INNER
    CMP r5, r2
    BLO INNER

    ADD r4, #1; if r4 < r3 goto OUTER
    CMP r4, r3
    BLO OUTER

SCALAR
    LDR r0, = arr; 1 col
    LDR r1, = arr + 16; 5 col
    LDR r4, = 6; rows
    LDR r5, = 0; 0..rows
    LDR r6, = 0; scalar
    LDR r7, = scalar

LOOP
    LDR r2, [r0]; load el from 1 col
    LDR r3, [r1]; load el from 5 col
    MUL r2, r2, r3
    ADD r6, r2
    ADD r0, #32; 8*4 next cell of 1 col
    ADD r1, #32; 8*4 next cell of 5 col
    ADD r5, #1
    CMP r5, r4
    BLO LOOP
    STR r6, [r7]; answer
```

```

    LDR r0, = arr + 96; (8*4)*3 4 row
    LDR r3, = 0; 0..cols
    LDR r4, = 0; sum
    LDR r5, = 0; count
    LDR r6, = sum
    LDR r7, = count

COUNT_AND_SUM
    LDR r1, [r0]
    ADD r0, #4

    CMP r3, #8; >= cols
    BGE DONE

    ADD r3, #1
    CMP r1, #-49
    BLO SUM
    CMP r1, #73
    BGE SUM
    B COUNT_AND_SUM

SUM
    ADD r4, r1
    ADD r5, #1
    B COUNT_AND_SUM

DONE
    STR r4, [r6]
    STR r5, [r7]

STOP B STOP
    ALIGN
    AREA InputData, Data, ReadOnly
    EXPORT arr
arr    DCD 10, 64, -94, 77, 99, 18, 52, -11
    DCD -23, -77, -45, 65, 77, 66, -24, 69
    DCD -81, -78, -82, -39, -90, -78, 24, 95
    DCD -18, -64, -74, -28, -16, -40, 91, 42
    DCD 56, -19, 86, 34, -83, -99, -31, -51
    DCD -70, -58, 13, 98, 90, 46, -77, 37; -37 <= a < 69 => 13+46+37 = 96

    AREA OutputData, Data, ReadWrite
    EXPORT res
    EXPORT scalar
    EXPORT sum
    EXPORT count

res    SPACE 6 * 8 * 4
scalar SPACE 4
sum    SPACE 4
count  SPACE 4
    END

```

## Результат

Memory 1										Watch 1		
Address: 0x000000E4										Name	Value	Type
0x000000E4:	0000000010	0000000064	-0000000094	0000000077	0000000099	0000000018	0000000052	-0000000011		scalar*-1	4151	uint
0x00000104:	-0000000023	-0000000077	-0000000045	0000000065	0000000077	0000000066	-0000000024	0000000069		(long)r4	-5	long
0x00000124:	-0000000081	-0000000078	-0000000082	-0000000039	-0000000090	-0000000078	0000000024	0000000095		(long)r5	0x00000004	long
0x00000144:	-0000000018	-0000000064	-0000000074	-0000000028	-0000000016	-0000000040	0000000091	0000000042		<Enter expression>		
0x00000164:	0000000056	-0000000019	0000000086	0000000034	-0000000083	-0000000099	-0000000031	-0000000051				
0x00000184:	-0000000070	-0000000058	0000000013	0000000098	0000000090	0000000046	-0000000077	0000000037				

Результати роботи програми - транспонування матриці, обчислення скалярного добутку 1 та 5 стовпців та сума та к-сть елементів 4того рядку, які задовільняють умовам

Скалярний добуток -  $(10*99)+(-23*77)+(-81*-90)+(-18*-16)+(56*(-83))+(-70*90)$   
= -4151

## Висновки

Під час виконання лабораторної роботи я ознайомивсь на приладі циклічної програми з основними командами асемблера процесорів Cortex- M3/M4; розвинув навички складання програми з вкладеними циклами; відтранслявав і виконав покроково в режимі відлагодження програму, складену відповідно до свого варіанту; перевінив виконання тесту.