Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**ЛАБОРАТОРНА РОБОТА №4**

**З ДИСЦИПЛІНИ “** **АРХІТЕКТУРА КОМП’ЮТЕРІВ - 3”**

**Виконав:**

Студент ІІІ курсу ФІОТ

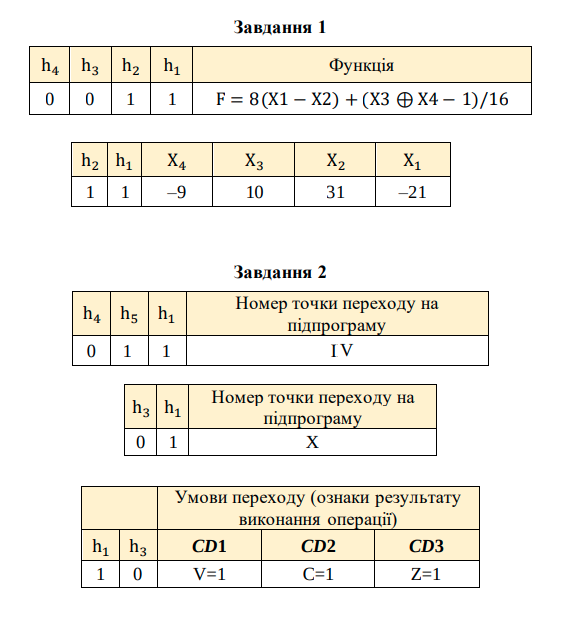
групи ІВ-83

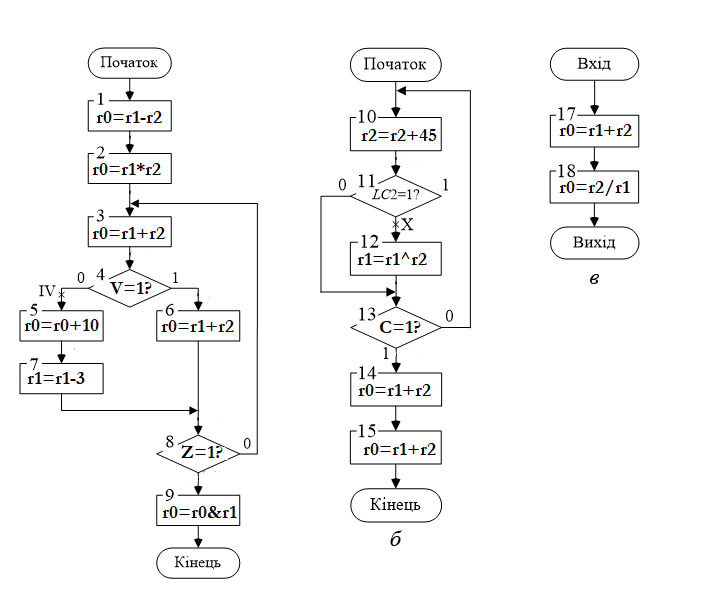
**Литвиненко Олександр**

м. Київ – 2021 р.

**Тема:** РОЗРОБЛЕННЯ ПРОГРАМ ОБРОБКИ ДАНИХ ДЛЯ ПРОЦЕСОРНОГО ЯДРА CORTEX М4

**Мета:** ВИВЧЕННЯ АРХІТЕКТУРНИХ ОСОБЛИВОСТЕЙ, СИСТЕМИ КОМАНД, ПРИНЦИПІВ ОРГАНІЗАЦІЇ КОМАНД УМОВНИХ ТА БЕЗУМОВНИХ ПЕРЕХОДІВ ТА ПЕРЕХОДІВ НА ПІДПРОГРАМИ, КОМАНД РОБОТИ З ПАМ’ЯТТЮ ТА СПОСОБІВ АДРЕСАЦІЇ ОПЕРАНДІВ.





**Код програми**

lab4.S

.global lab4

.syntax unified

#define X1 #-21

#define X2 #31

#define X3 #10

#define X4 #-9

lab4:

push {lr}

bl task1

bl task2

pop {pc}

task1:

push {lr}

mov r0, X1

mov r1, #0xFFFFFFFF

mov r2, X2

mov r3, #0

mov r4, X3

mov r5, #0

mov r6, X4

mov r7, #0xFFFFFFFF

// Calculation: 8 \* (X1 - X2) + (X3 ^ X4 - 1) / 16

// X1 - X2

sub r0, r2

sbc r1, r3

// 8 \* (X1 - X2)

lsl r0, #3

lsl r1, #3

// X3 ^ X4

eor r4, r6

eor r5, r7

// X3 ^ X4 - 1

sub r4, #1

sbc r5, #1

// (X3 ^ X4 - 1) / 16

asr r4, #4

lsr r5, #4

// 8 \* (X1 - X2) + (X3 ^ X4 - 1) / 16

add r0, r4

adc r1, r5

bl result\_correction

pop {pc}

// Result correction

result\_correction:

push {lr}

bvc step2

lsr r0, #1

lsr r1, #1

step2:

orr r0, #0

bne step4

orr r1, #0

bne step4

mov r0, #0b10110011

step4:

add r1, #0b10110011

and r5, r1, #0xc0000000

beq step7

eor r5, #0xc0000000

beq step7

lsr r0, #1

lsr r1, #1

step7:

bvs end\_correction

push {r0}

push {r1}

pop {r1}

pop {r0}

end\_correction:

pop {pc}

task2:

push {lr}

mov r1, #25

mov r2, #5

program\_a:

// 1: r0 = r1 - r2

sub r0, r1, r2

// 2: r0 = r1 \* r2

mul r0, r1, r2

bl program\_b

jump3:

// 3: r0 = r1 + r2

add r0, r1, r2

// 4: jump1

bvs jump1

// 6: r0 = r1 + r2

add r0, r1, r2

b jump2

jump1:

// 5: r0 = r0 + 10

add r0, #10

// 7: r1 = r1 - 3

sub r1, #3

jump2:

// 8: jump3

beq jump3

// 9: r0 = r0 & r1

and r0, r1

pop {pc}

program\_b:

push {lr}

jump5:

// 10: r2 = r2 + 45

add r2, #45

// 11: jump4

bcs jump4

bl program\_c

// 12: r1 = r1 ^ r2

eor r1, r2

jump4:

// 13: jump5

bvs jump5

// 14: r0 = r1 + r2

add r0, r1, r2

// 15: r0 = r1 - r2

sub r0, r1, r2

pop {pc}

program\_c:

push {lr}

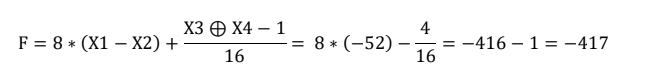
// 17: r0 = r1 + r2

adds r0, r1, r2

// 18: r0 = r2 / r1

udiv r0, r2, r1

pop {pc}



**Висновок:** в результаті виконання роботи було розроблено алгоритм та

програму обчислення функції у кодах асемблеру, що задана за варіантом.

Виконано нормалізацію результату.