Національний Технічний Університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет Інформатики та Обчислювальної Техніки

Кафедра обчислювальної техніки

**Лабораторна робота №4**

з курсу «Архітектура комп’ютерів»

Виконав:

студент 3 курсу

групи ІВ-83

Бойченко Дмитро

Перевірив:

Нікольський С. С.

Київ 2021

Залікова 8302 - 11111101101102

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 0 | 1 | 1 | 0 |
| *h*5 | *h*4 | *h*3 | *h*2 | *h*1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| h4 | h3 | h2 | h1 | Функція\* |
| 0 | 1 | 1 | 0 |  |

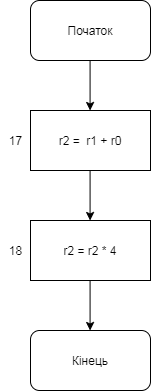
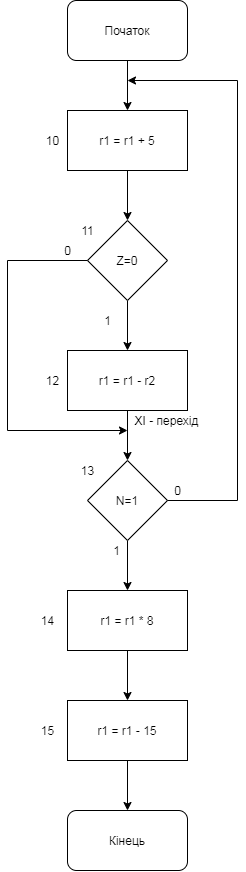
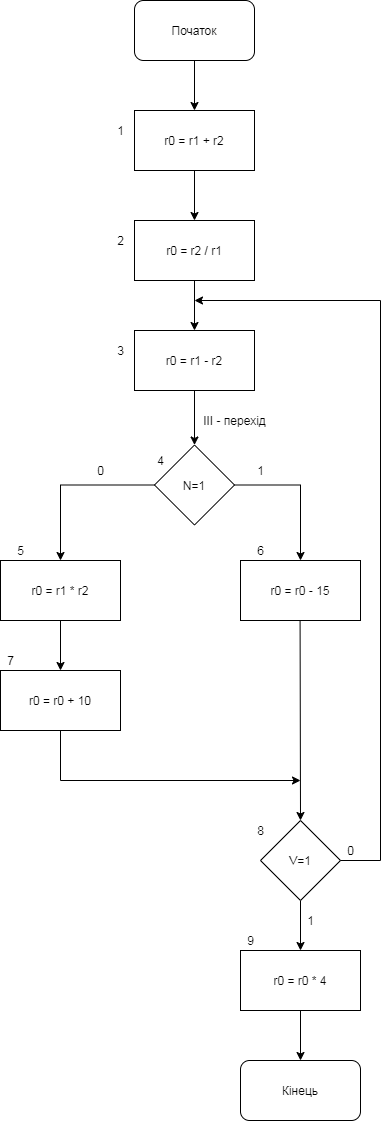
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | *X*1 | *X*2 | *X*3 | *X*4 |
| 1 | 0 | 18 | –9 | 23 | 11 |

|  |  |  |  |
| --- | --- | --- | --- |
| h4 | h5 | h1 | Номер точки переходу на підпрограму |
| 0 | 1 | 0 | ІІІ |

|  |  |  |
| --- | --- | --- |
| h2 | h1 | Номер точки переходу на підпрограму |
| 1 | 0 | XI |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Умови переходу (ознаки результату виконання операції) | | |
| h1 | h3 | CD1 | CD2 | CD3 |
| 0 | 1 | N=1 | Z=0 | V=1 |

Посилання на репозиторій: [marchiani/AK-3\_lab4 (github.com)](https://github.com/marchiani/AK-3_lab4)

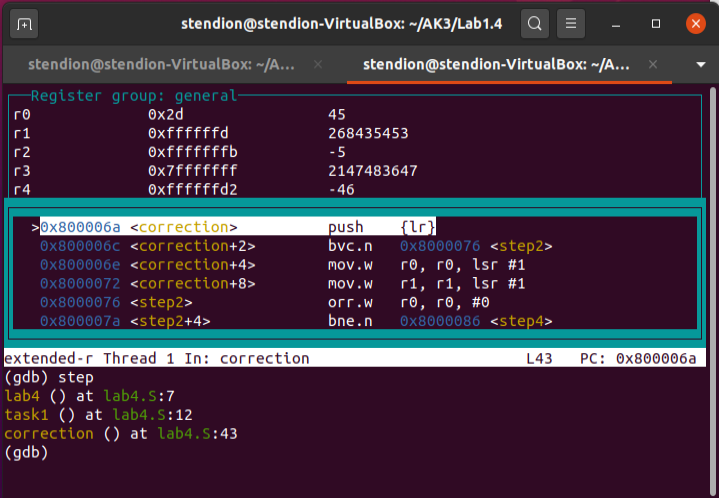


Результат

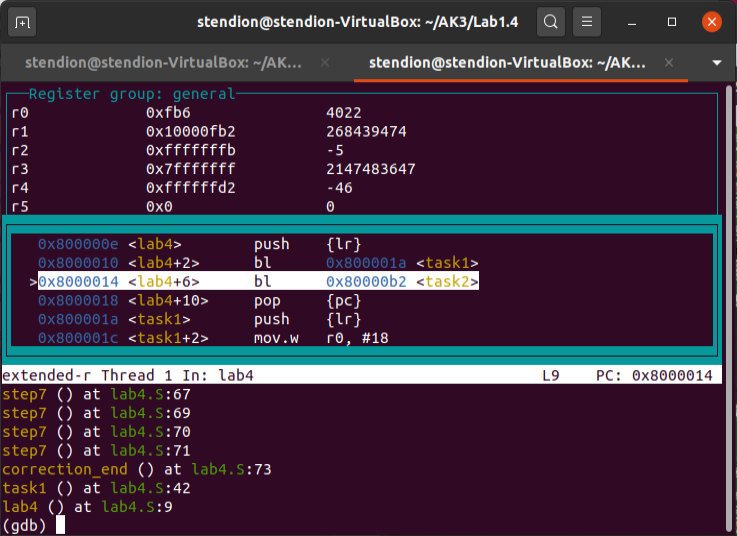


X1 = 18, X2 = -9, X3 = 23

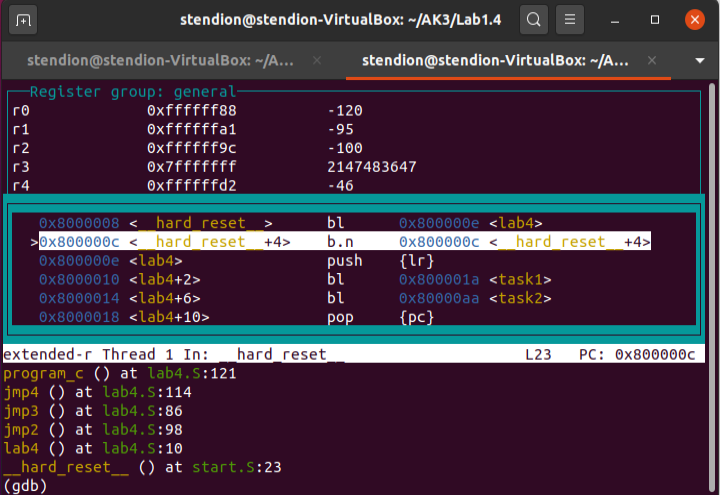
F = (18 v (-9)/2)/8 - 2(~23 ^ 1) = (18 v (-5))/8 - 2((-24) ^ 1) = (-5) / 8 - 2 \*(-23) = -1 + 46 = 45



Корекція результату



ob111110110110 = 4022



r0 = 10, r1 =5, r2 = 20

Step 1: r0 = r1 + r2 = 25

Step 2: r0 = r2 / r1 = 4

Step 3: r0 = r1 - r2 = -15

Step 4: transition to program\_b: r1 = r1 + 5 = 10

Step 5: Z = 0 - True: r1 = r1 - r2 = 10 - 20 = -10

Step 6: transition to program\_c: r2 = r1 + r0 = -10 -15 = -25

Step 7: r2 = r2 \* 4 = -100

Step 8: N = 1 - True: r1 = r1 \* 8 = -80

Step 9: r1 = r1 - 15 = -95

Step 10: N = 1: True: r0 = r0 -15 = -30

Step 11: V = 1: True: r0 = r0 \* 4 = -120

End results: r0 = -120, r1 = -95, r2 = - 100

Код програми

lab4.S

.global lab4

.syntax unified

#define x1 #18

#define x2 #-9

#define x3 #23

lab4:

push {lr}

bl task1

bl task2

pop {pc}

task1:

push {lr}

mov r0, x1

mov r1, #0

mov r2, x2

mov r3, #0xFFFFFFFF

// ~x3

mvn r4, x3

mov r5, #0

// F = (x1 | x2 / 2)/8 - 2(~x3 ^ 1)

// x2 / 2

asr r2, #1

asr r3, #1

// x1 | x2

orr r0, r2

orr r1, r3

// (x1 | x2 / 2)/8

asr r0, #3

asr r1, #1

asr r1, #1

asr r1, #1

// ~x3 ^ 1

eor r4, #1

eor r5, #1

// 2(~x3 ^ 1)

lsl r4, #1

lsl r5, #1

// (x1 | x2 / 2)/8 - 2(~x3 ^ 1)

sub r0, r4

sbc r1, r5

bl correction

pop {pc}

correction:

push {lr}

bvc step2

lsr r0, #1

lsr r1, #1

step2:

orr r0, #0

bne step4

orr r1, #0

bne step4

step3:

add r0, #0b111110110110

step4:

add r1, #0b111110110110

step5:

and r5, r1, #0xc0000000

beq step7

eor r5, #0xc0000000

beq step7

step6:

lsr r0, #1

lsr r1, #1

step7:

bvs correction\_end

push {r0}

push {r1}

pop {r1}

pop {r0}

correction\_end:

pop {pc}

task2:

push {lr}

mov r0, #10

mov r1, #5

mov r2, #20

program\_a:

// 1: r0 = r1 + r2

add r0, r1, r2

// 2: r0 = r2 / r1

udiv r0, r2, r1

jmp3:

// 3: r0 = r1 - r2

sub r0, r1, r2

bl program\_b

// 4: jmp1

bmi jmp1

// 6: r0 = r0 - 15

sub r0, #15

b jmp2

jmp1:

// 5: r0 = r1 \* r2

mul r0, r1, r2

// 7: r0 = r0 + 10

add r0, #10

jmp2:

// 8: jmp3

bvs jmp3

// 9: r0 = r0 \* 4

lsl r0, #2

pop {pc}

program\_b:

push {lr}

jmp5:

// 10: r1 = r1 + 5

add r1, #5

// 11: jmp4

bne jmp4

// 12: r1 = r1 - r2

sub r1, r2

bl program\_c

jmp4:

// 13: jmp5

bmi jmp5

// 14: r1 = r1 \* 8

lsl r1, #3

// 15: r1 = r1 - 15

sub r1, #15

pop {pc}

program\_c:

push {lr}

// 17: r2 = r1 + r0

add r2, r1, r0

// 18: r2 = r2 \* 4

lsl r2, #2

pop {pc}

**Висновок:** в результаті виконання роботи було розроблено алгоритм та програму обчислення функції у кодах асемблеру, що задана за варіантом, а також виконано нормалізацію результату. У 2 частині роботи було розроблено програму згідно із заданим алгоритмом, забезпечено звернення з основної програми до першої підпрограми та звернення з першої підпрограми до другої.

Роботу програми було протестовано у відлагоджувачі gdb, на виході маємо правильний результат, який збігається з результатами ручного обрахування.