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 ІО-81

Мисак Олександр

Перевірив:

Каплунов А. В.

Київ 2021

## Завдання

Залікова 8119 - 11111101101112

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

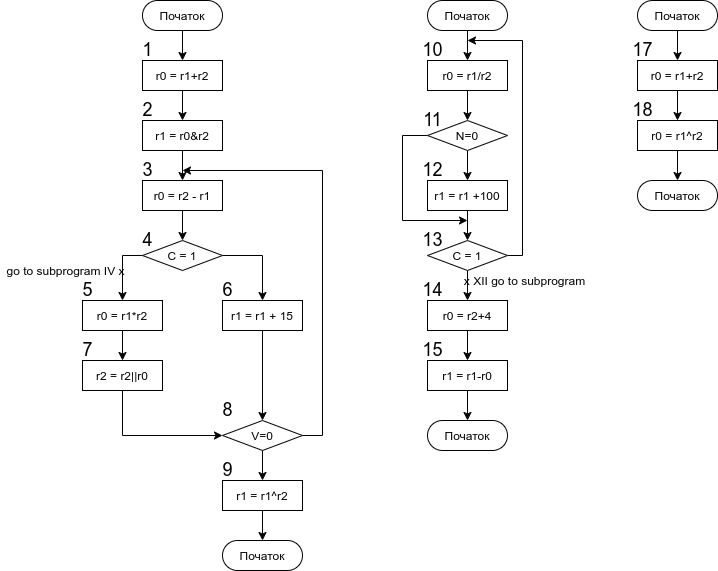
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | *X*1 | *X*2 | *X*3 | *X*4 |
| 1 | 1 | -9 | 10 | 31 | -21 |

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

|  |  |  |
| --- | --- | --- |
| h2 | h1 | Номер точки переходу на підпрограму |
| 1 | 1 | XII |

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

## Алгоритм для другої частини



# Результати

## Змоделюємо результат:

x1 = -9

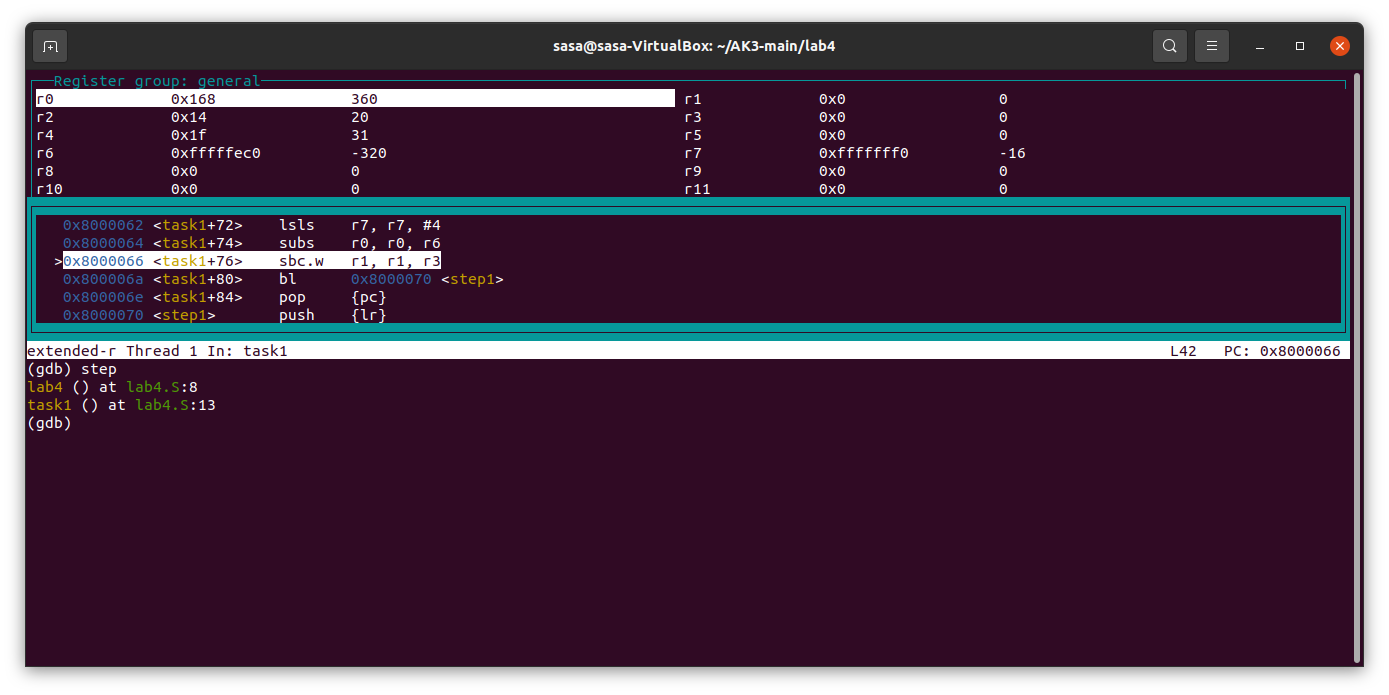
x2 = 10

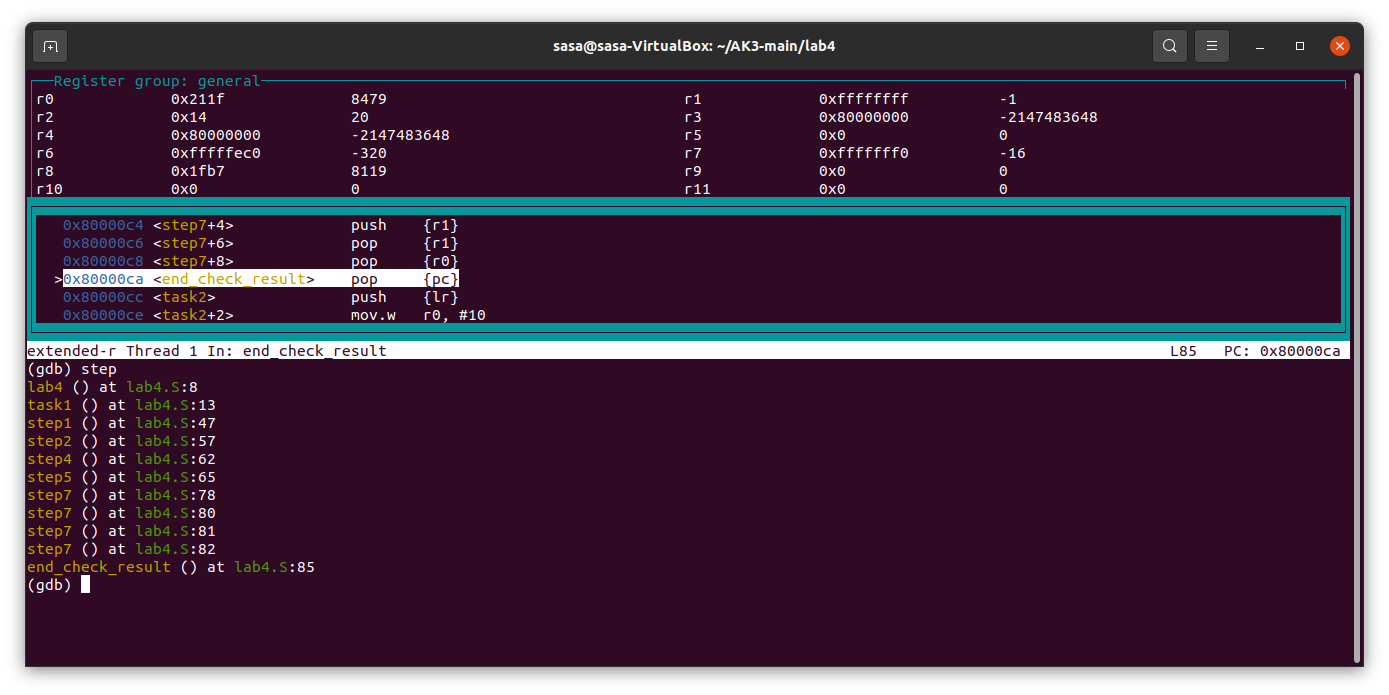
x3 = 31

x4 = -21

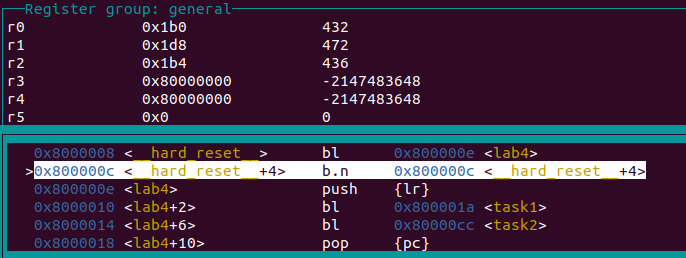
f(x1,x2,x3,x4) = 360

## Результати виконання

Корекція результату



Результат другої програми



r0 = 10, r1 =5, r2 = 20

Step 1: r0 = r1 + r2 = 25

Step 2: r1= r0 and r2 = 16

Step 3: r0 = r2 - r1 = 4

Step 4: C = 1 - False: go to program\_b: r2 = r1 / r0 = 4

Step 5: N= 0 - True: r1 = r1 +100 = 116

Step 6: C = 1 : True: go to program\_c: r0 = r1 + r2 = 120

Step 7: r0 = r1 ^ r2 = 112

Step 8: r0 = r2+4 = 8

Step 9: r1 = r1 - r0= 108

Step 10: r0 = r1 \* r2= 432

Step 11: r2 = r2 || r0 = 436

Step 12: V = 0: True: r1 = r1 ^ r2= 472

End results: r0 = 432, r1 = 472, r2 = 436

# Код програми

lab4.S

.global lab4

.syntax unified

#define x1 #-9

#define x2 #10

#define x3 #31

#define x4 #-21

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

// F = 2(X1 & ~(X2) + X3) - 16(X4 + 1)

mvn r2, r2

mvn r3, r3

adds r2,r4

adc r3, r5

and r0, r2

and r1, r3

lsls r0, #1

lsl r1, #1

adds r6, #1

adc r7, #0

lsl r6, 4

lsls r7, 4

subs r0, r6

sbc r1, r3

bl step1

pop {pc}

step1:

push {lr}

bvc step2

mov r8, 0

lsrs r1, #1

adc r8, 0

lsl r8, 31

lsr r0, #1

orr r0, r8

step2:

orr r0, #0 // check if 0

bne step4 // if != 0 go to step4

step3:

mov r1, #0b1111110110111

step4:

mov r8, #0b1111110110111

add r0, r8

step5:

and r3, r1, #0b10000000000000000000000000000000

and r4, r1, #0b01000000000000000000000000000000

lsl r4, #1

cmp r3, r4

beq step7

step6:

mov r8, 0

lsrs r1, #1

adc r8, 0

lsl r8, 31

lsr r0, #1

orr r0, r8

step7:

bvs end\_check\_result

push {r0}

push {r1}

pop {r1}

pop {r0}

end\_check\_result:

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: r1 = r0 & r2

and r1, r0, r2

jmp3:

// 3: r0 = r2 - r1

sub r0, r2, r1

// 4: jmp1

bcs jmp1

// 6: r1 += 15

adds r1, #15

b jmp2

jmp1:

bl program\_b

// 5: r0 = r1 \* r2

mul r0, r1, r2

// 7: r2 = r2||r0

orr r2,r0

jmp2:

// 8: jmp3

bvs jmp3

// 9: r1 = r1^r2

eor r1,r2

pop {pc}

program\_b:

push {lr}

jmp5:

// 10: r2 = r1/r0

udiv r2, r1, r0

// 11: jmp4

bmi jmp4

// 12: r1 = r1 +100

adds r1, #100

jmp4:

// 13: jmp5

bcs jmp5

bl program\_c

// 14: r0 = r2 +4

add r0, r2, #4

// 15: r1 -= r0

sub r1, r0

pop {pc}

program\_c:

push {lr}

// 17: r0 = r1 + r2

add r0, r1, r2

// 18: r0 = r1^r2

eor r0, r1, r2

pop {pc}

# **Висновок:**

В результаті виконання роботи було розроблено алгоритм та програму обчислення функції, що задана за варіантом, за допомогою асемблеру, 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, в кінцевому результаті маємо повністю функціонуючий програмний код, який повертає коректне значення, що збігається з результатами ручного обчислення.