

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Архітектура комп'ютера - 3

Лабораторна робота №4

Виконав:
студент групи ІВ-83
Грисюк М. О.

Перевірів: Нікольський С. С.

Київ - 2021 р.

Варіант

Номер варіанта 8306 (у двійковій системі числення 0010000001110010)
 $h_1=0, h_2=1, h_3=0, h_4=0, h_5=1$.

h4	h3	h2	h1	Функція
0	0	1	0	$F = 16[(X1 - 1) \& X2] - (X3 + X4) / 4$

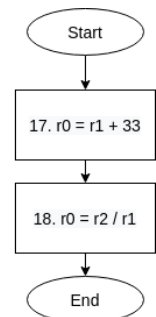
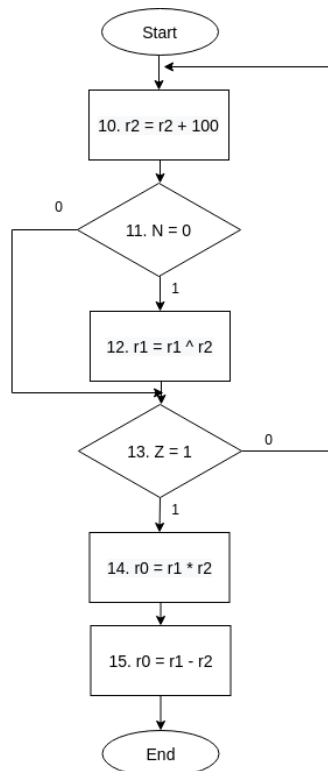
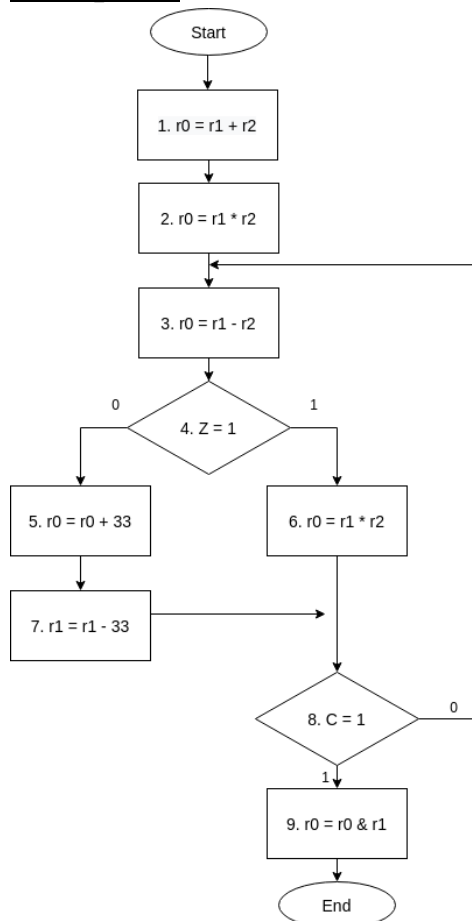
h_2	h_1	X1	X2	X3	X4
1	0	18	-9	23	11

h_4	h_5	h_1	Номер точки переходу на підпрограму
0	1	0	III

h_2	h_1	Номер точки переходу на підпрограму
1	0	XI

		Умови переходу (ознаки результату виконання операції)		
h_1	h_3	CD1	CD2	CD3
0	0	Z=1	N=0	C=1

Алгоритм



Код програми

lab4.S

```
.global lab4
.syntax unified

#define X1 #18
#define X2 #-9
#define X3 #23
#define X4 #11

#define bigger #0xFD100000
#define less #0b10000001110010

lab4:
    push {lr}
    bl task1
    bl task2
    pop {pc}

task1:
    push {lr}
    mov r0, X1
    mov r1, #0

    mov r2, X2
    mov r3, #0xFFFFFFFF

    mov r4, X3
    mov r5, #0

    mov r6, X4
    mov r7, #0

    // Calculation: 16 * [(X1 - 1) & X2] - (X3 + X4) / 4

    // X1 - 1
    sub r0, #1
    sbc r1, #1

    // (X1 - 1) & X2
    and r0, r2
    and r1, r3

    // 16 * [(X1 - 1) & X2]
    lsl r0, #4
    lsl r1, #4

    // X3 + X4
    add r4, r6
    adc r5, r7
```

```

    // (X3 + X4) / 4
    asr r4, #2
    lsr r5, #2

    // 16 * [(X1 - 1) & X2] - (X3 + X4) / 4
    sub r0, r4
    sbc 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
    ldr r7, =bigger
    mov r0, r7
step4:
    ldr r7, =less
    add r1, r7
    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
    add r0, r1, r2

    // 2: r0 = r1 * r2
    mul r0, r1, r2

bl program_b

jump3:
    // 3: r0 = r1 - r2
    sub r0, r1, r2

    // 4: jump1
    beq jump1

    // 6: r0 = r1 * r2
    mul r0, r1, r2

    b jump2
jump1:
    // 5: r0 = r0 + 33
    add r0, #33

    // 7: r1 = r1 - 33
    sub r1, #33
jump2:
    // 8: jump3
    //bls jump3

    // 9: r0 = r0 & r1
    and r0, r1
    pop {pc}

program_b:
    push {lr}
jump5:
    // 10: r2 = r2 + 100
    add r2, #100

    // 11: jump4
    bpl jump4

    bl program_c

    // 12: r1 = r1 ^ r2
    eor r1, r2
jump4:
    // 13: jump5
    //beq jump5

```

```

// 14: r0 = r1 * r2
mul r0, r1, r2

// 15: r0 = r1 - r2
sub r0, r1, r2

pop {pc}

program_c:
push {lr}

// 17: r0 = r1 + 33
adds r0, r1, 33

// 18: r0 = r2 / r1
udiv r0, r2, r1

pop {pc}

```

Результати виконання лабораторної роботи:

The screenshot shows a GDB debugger window titled "misha_hrysiuk@not-my-computer: ~/2_semestr/4_Lab". The "Register group: general" section displays the following values:

Register	Value	Decimal
r0	0x108	264
r1	0xffffffffdf	-33
r2	0xffffffff7	-9
r3	0xfffffffff	-1
r4	0x8	8
r5	0x0	0

The assembly code window shows the following instructions:

```

>0x8000072 <result_correction> push    {lr}
0x8000074 <result_correction+2> bvc.n   0x800007e <step2>
0x8000076 <result_correction+4> mov.w   r0, r0, lsr #1
0x800007a <result_correction+8> mov.w   r1, r1, lsr #1
0x800007e <step2>               orr.w   r0, r0, #0
0x8000082 <step2+4>             bne.n   0x800008e <step4>

```

The status bar indicates "extended-r Thread 1 In: result_correction L63 PC: 0x8000072". The command window shows the following commands and output:

```

(gdb) s
lab4 () at lab4.S:13
task1 () at lab4.S:19
result_correction () at lab4.S:63
(gdb)

```

```

>>> X1 = 18
>>> X2 = -9
>>> X3 = 23
>>> X4 = 11
>>> 16*((X1-1)&X2)-(X3+X4)/4
263.5
>>>

```

