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-3»

Виконав

Студент 3 курсу ФІОТ

групи ІВ-83

Денисюк Т.Ю.

Перевірив: Нікольський С.С.

Київ 2021

**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.

**Хід роботи**

Варіант:

830910 = 1 1111 1010 10012

**Завдання 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| h4 | h3 | h2 | h1 | Функція |
| 1 | 0 | 0 | 1 | https://lh3.googleusercontent.com/Ys2Urxpby25Y-Gv2xlKx2KHO5W3uysX4ue4xLST-6ZtikjKh8ShR91KlIs2eq-dgRJDMZKWjGmu3kgKQFxDzOnvDumjYGCOn2lqbu7x3imL4Vw6RSZEoxCnE0LHiVgovNagmeiE |

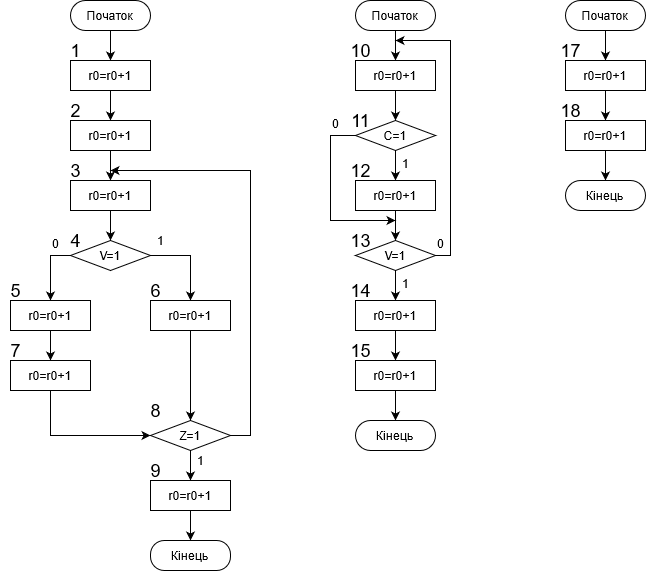
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| https://lh4.googleusercontent.com/MDimnfp6ltkXscYN9Y2SDlCmrUJylzNvbXaU7lkhMwbV2R_JaXkmqt8jwRA0olOAjgQzOYcwGvtY4Es9zK_gizjQltOp3o3Nx0-dLsXzkqFMtjDxhYWWD1KH8qsE3LpzEqsji7M | https://lh6.googleusercontent.com/tIQ8oRcqX9nCsi5St8ZyXU5NYTMVDauo_inu41_SrXyJZKYeG6uA1f0xTaG_gHF6hxLC5lY6sBOYu7-7QyQNthANvtYuEbivm72M_BpeLknxyLTztoZqIwfo3F2pHMf9p8HS7Cw | *X*1 | *X*2 | *X*3 | *X*4 |
| 0 | 1 | *12* | *2* | *–10* | *15* |

**Завдання 2**

|  |  |  |  |
| --- | --- | --- | --- |
| https://lh4.googleusercontent.com/Yvz_QFbMDLKrdvFw3KWvnKEhtiusRqdf2DqHgYMZ-dwKQ_hLiZ5rnDFZ48-2KooflyTiiUZBSd-RYkcEqf5TNsSNWRyAzeZD2bbwB5mnDiCklLm5dh61srDP245N62k3c5-_1KY | https://lh4.googleusercontent.com/mikDsqDW10DLpENgLPWWn8FYSR8w1-f_PcPpKarhM50-ZA9unqkdqXy99ySHRTC6VZjE0Jwh5B7zI1ZLBc91Obl72qjX06BNFarp7cdWRZO0e6M9LsD6IrpLEw3f_fnQIqSgX_g | https://lh5.googleusercontent.com/T_JV_0LwzXm92dQkUzczeOaph3rAgh8n3-yev3L3ZNveg587PKoZVbCVDjye0ENRnSQPkronaQITKHeLfM9jHphNLVMv4Rlq5j85-hDuOXA68v2SFI5C3LwRKC7l5KvtQB3D3i8 | Номер точки переходу на підпрограму |
| 1 | 0 | 1 | VI |

|  |  |  |
| --- | --- | --- |
| https://lh6.googleusercontent.com/cCFIT3QwJUUGLVwiTqWJkk4GUqpjG4YSKW484X9FE1P_e7FHdRYMQpJKB_V8rrPL9uq-kNlqoXOCvmW4M7UKt4jGMsxTMI_FhyYXHYowxGY3ZxbYyIoM1znpynVInHG-oymm7X0 | https://lh6.googleusercontent.com/wY9rUXjMuRvmbHUeqq0ofXGJ-yjC2E-9Ly4PpY5rdtD53qyjLnJb9vFPrSMTPFLW5vSSgPgbPcnlElRHl0QGPsK5yv0KfzgiJnIEZ8lt0ATi6F9BjlkOYGdzu40tMl23g2n2Rc4 | Номер точки переходу на підпрограму |
| 0 | 1 | X |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Умови переходу (ознаки результату виконання операції) | | |
| https://lh6.googleusercontent.com/kPoamK0KDIbYQQb8KQC_ty7g2oEhaVHsniCZFnrcz0iDf9qwEtWcbui1NgcUjK3jOVUdH2HZMsQwW1Xaj5NV1TRFr5F1D2I9hfm4HPJ31SJhKq2s88y99gj-7M2v8kvA8lPPxDo | https://lh3.googleusercontent.com/xXE5xGzwxIgdrDKo4TiV6JAPKgQLVcDT7oGyWzFL34jUNKNLalL_NZIZkYFRasIJhj4FiZGVf2th01TDnNXpLmNq1SLyEP9bzCTv5nGRIoVV4jrLwHpuVYtYzzj-BDHpNsjXmK8 | *CD*1 | *CD*2 | *CD*3 |
| 1 | 0 | *V=1* | *C=1* | *Z=1* |

****

**Лістинг**

.global lab4

.syntax unified

#define X1 #12

#define X2 #2

#define X3 #-10

#define X4 #15

lab4:

push {lr}

bl task1

bl task2

pop {pc}

task1:

push {lr}

mov r0, X1

mov r1, #0

mov r2, X2

mov r3, #0

mov r4, X3

mov r5, #-1

mov r6, X4

mov r7, #0

// Calculation (X3+1-X4)/4∨4(X1/2+X2)

// (X3+1)

add r4, #1

adc r5, #0

// (X3+1-X4)

sub r4, r6

sbc r5, r7

// (X3+1-X4)/4

asr r5, #2

lsr r4, #2

// x1/2

asr r1, #1

lsr r0, #1

// x1/2+x2

add r0, r2

adc r1, r3

//4(X1/2+X2)

lsl r0, #2

lsl r1, #2

// (X3+1-X4)/4∨4(X1/2+X2)

orr r0, r4

orr r1, r5

bl correction

pop {pc}

// Result correction

correction:

push {lr}

bvc step2

lsr r0, #1

lsr r1, #1

step2:

orr r0, #0 // Update flag reg

bne step4

orr r1, #0 // Update flag reg

bne step4

mov r0, #0b1111110101001

step4:

mov r11, #0b1111110101001

add r1, r11

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}

// task 2

task2:

push {lr}

// declaration

mov r0, #1

program\_1:

// 1. r0 = r0 + 1

add r0, #1

// 2. r0 = r0 + 1

add r0, #1

point1:

// 3. r0 = r0 + 1

add r0, #1

// 4. jump1

bvc point2

// 6. r0 = r0 + 1

add r0, #1

b point3

point2:

// 5. r0 = r0 + 1

add r0, #1

// 7. r0 = r0 + 1

adds r0, #1

point3:

// 8. jump3 beq | bne for Z

beq point1

bl program\_2

// 9. r0 = r0 + 1

add r0, #1

pop {pc}

program\_2:

push {lr}

point4:

// 10. r0 = r0 + 1

adds r0, #1

// 11. jump4 bcs | bcc for C

bcs point5

bl program\_3

// 12. r0 = r0 + 1

add r0, #1

point5:

// 13. jump 5 bvs | bvc for V

bvs point4

// 14. r0 = r0 + 1

add r0, #1

// 15. r0 = r0 + 1

add r0, #1

pop {pc}

program\_3:

push {lr}

// 17. r0 = r0 + 1

add r0, #1

// 18. r0 = r0 + 1

add r0, #1

pop {pc}

