

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий Фізико-технічний інститут

Архітектура комп'ютерних систем
Комп'ютерний практикум
Робота №3

Виконав:
студент групи ФІ-12
Завалій Олександр
Перевірив:
Козленко О.В.

Робота №3.

Основи побудови програми на асемблері в архітектурі ARM

Мета:

Ознайомитися з створенням базової програми на мові асемблер для архітектури ARM

Варіант №4

Зміст індивідуального завдання:

1. Завантажити VisUAL2.
2. Ознайомитись з теоретичними положеннями.
3. Визначити змінні, занести відповідні значення у регістри та організувати цикл роботи для архітектури ARM згідно свого варіанту:
Дано натуральні числа n, a_1, \dots, a_n . Отримати середнє арифметичне тих членів послідовності a_1, \dots, a_n які при діленні на 5 дають залишок 1 чи 2.
4. Підготувати звіт і представити на захист.

Code

```
ENTRY
    LDR    R0, =ASize
    LDR    R1, [R0]
    LDR    R2, =Array
    MOV    R3, #0
    MOV    R4, #0
    MOV    R5, #0

compareLoop
    CMP    R3, R1
    BEQ    endLoop

    LDR    R6, [R2, R3, LSL #2]
    MOV    R7, R6

calculateRemainder
    CMP    R7, #5
    BLT    checkRemainder
    SUB    R7, R7, #5
    B      calculateRemainder

checkRemainder
    CMP    R7, #1
    BEQ    addToSum
    CMP    R7, #2
    BEQ    addToSum

nextElement
    ADD    R3, R3, #1
    B      compareLoop

addToSum
    ADD    R4, R4, R6
    ADD    R5, R5, #1
    B      nextElement

endLoop
    CMP    R5, #0
    BEQ    storeZero

    MOV    R0, R4
    MOV    R1, R5
    MOV    R2, #0
    MOV    R3, #0
```

```

calculateAverage
    CMP    R1, #0
    BEQ    storeAverage

divisionLoop
    CMP    R0, R1
    BLT    storeAverage
    SUB    R0, R0, R1
    ADD    R2, R2, #1
    B      divisionLoop

storeAverage
    LDR    R0, =Result
    STR    R2, [R0]
    B      stopExecution

storeZero
    MOV    R2, #0
    LDR    R0, =Result
    STR    R2, [R0]
    B      stopExecution

stopExecution
    B      stopExecution

ASize      DCD    6
Array      DCD    3, 7, 12, 21, 26, 31
Result     DCD    0

```

```

array = [3, 7, 12, 21, 26, 31]
resut = []

[7] ✓ 0.0s

for i in array:
    if i % 5 == 1 or i % 5 == 2:
        resut.append(i)

[8] ✓ 0.0s

print(f'Result array: {resut}')

[9] ✓ 0.0s
... Result array: [7, 12, 21, 26, 31]

▷ ✓
print(f'Arithmetic mean: {sum(resut)/len(resut)}')

[10] ✓ 0.0s
... Arithmetic mean: 19.4

```

Results

The screenshot shows the VisUAL2 debugger interface. The main window displays assembly code for a program. The code starts with an ENTRY point, followed by loading registers R0, R1, R2, R3, R4, and R5. It then enters a loop labeled 'compareLoop' which compares R3 and R1, and branches to 'endLoop' if below equal. The 'calculateRemainder' block calculates the remainder of R7 divided by R5. The 'checkRemainder' block checks if the remainder is zero and branches to 'nextElement' if not below equal. The 'nextElement' block increments R3 and branches back to 'compareLoop'. The 'addToSum' block increments R4 and R5, and branches to 'nextElement'. The 'endLoop' block compares R5 with zero and branches to 'storeZero' if below equal. Finally, it moves R4 to R0.

Symbol	Address	Value
ASize	0x200	6
Array	0x204	3
	0x208	7
	0x20C	12
	0x210	21
	0x214	26
	0x218	31
Result	0x21C	19

Registers: N 1 Z 0 C 0 V 0

The screenshot shows the VisUAL2 debugger interface. The main window displays assembly code for a program. The code continues from the previous screenshot, starting with adding R5 to R5 and branching to 'nextElement'. It then enters a loop labeled 'endLoop' which compares R5 with zero and branches to 'storeZero' if below equal. The 'calculateAverage' block calculates the average of R0 and R1. The 'divisionLoop' block divides R0 by R1 and branches to 'storeAverage' if below equal. The 'storeAverage' block stores the result in R2 and branches to 'stopExecution'. The 'storeZero' block stores zero in R2 and branches to 'stopExecution'. Finally, it moves R4 to R0.

Symbol	Address	Value
ASize	0x200	6
Array	0x204	3
	0x208	7
	0x20C	12
	0x210	21
	0x214	26
	0x218	31
Result	0x21C	19

Registers: N 1 Z 0 C 0 V 0

Результати роботи програм співпадають.