*Національний технічний університет України*

*‘Київський політехнічний інститут’*

*Факультет інформатики та обчислювальної техніки*

*Кафедра обчислювальної техніки*

***Лабораторна робота №2***

*З архітектури комп’ютерів - 2*

«Вивчення команд передачі управлення МК48»

*Виконав: Кириченко Ю.  
 Група: ІО-23*

*Номер зк: №2310*

***Київ — 2014***

***Вхідні дані:***

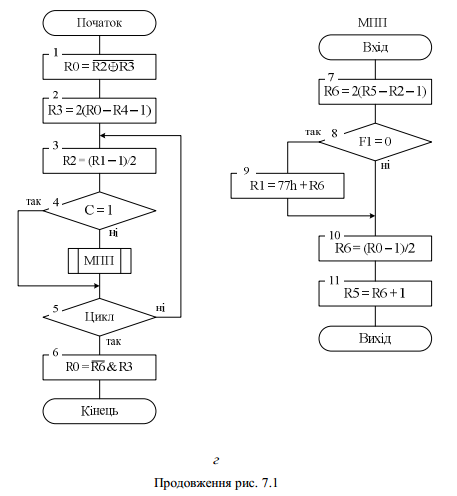
**Залікова книжка №** 231010= 1001 0000 01102

**Варіант завдання:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***h4*** | ***h1*** | ***R0*** | ***R1*** | ***R2*** | ***R3*** | ***R4*** | ***R5*** | ***R6*** | ***Кількість ітерацій(R7)*** |
| *0* | *0* | *4A* | *D3* | *E5* | *24* | *06* | *20* | *05* | *3* |

|  |  |
| --- | --- |
| ***h5 h2*** | ***Порт*** |
| *0 1* | *Р2* |

|  |  |
| --- | --- |
| ***h2 h3*** | ***Алгоритм*** |
| *1 1* | *рис. 7.1, г* |



***Виконання роботи:***

**Програма на символічному асемблері:**

Sel Rb0 ;банк регістрів №1

;---------------------Ініціалізація--------------------------------

In A, P2 ;Зчитування з порту Р2 в акамулятор

Mov R0, A ;Занесення даних з акамулятора в регістр

In A, P2

Mov R1, A

In A, P2

Mov R2, A

In A, P2

Mov R3, A

In A, P2

Mov R4, A

In A, P2

Mov R5, A

In A, P2

Mov R6, A

Mov R7, #3 ;Кількість ітерацій

;-------------------Обчислення NOT (R2 xor R3)-----------------------

Clr A ;Очистка акамулятора

Add A, R2 ;Підсумовування акамулятора і регістра R2

Xrl A, R3 ;Виключне АБО акамулятора і R3

Cpl A ;Інверсія акамулятора

Mov R0, A ;Повернення результату в R0

;-------------------Обчислення 2\*(R0-R4-1)---------------------------

Mov A, R4 ;Запис R4 в акамулятор

Cpl A ;Інверсія акамулятора

Add A, #1 ;А:=А+1

Add A, R0 ;А:=А+R0

Dec A ;Декримент акамулятора

Rl A ;Зсув акамулятора

Mov R3, A ;Запис результату в R3

;-------------------Обчислення (R1-1)/2------------------------------

Cycle:;Загальний цикл

Mov A, R1 ;A:=R1

Dec A ;A:=R-1

Clr C ;Очистка ознаки переносу С

Rr A ;Ділення на 2 (зсув вправо)

Mov R2, A ;R2:=(R1-1)/2

;---------------------------------------------------------------------

Jnc L1 ;Перевірка ознаки С на 0

Call Prog ;ВИКЛИК ПІДПРОГРАМИ, ЯКЩО С=1

L1: Djnz R7, Cycle ;Декримент регістра R7 і його перевірка на 0 --кінець загального циклу

;-------------------Обчислення Not R6 And R3---------------------------

Mov A, R6 ;A:=R6

Cpl A ;Інверсія А (NOT R6)

Anl A, R3 ;Логічне І акамулятора і регістра R3

Mov R0, A ;R0:=Not R6 And R3

Jmp Theend ;Перехід в кінець програми

Theend:;Мітка кінця програми

End ;Кінець програми

;-------------------------ПІДПРОГРАМА----------------------------------

Prog:;Мітка початку підпрограми

;----------------------Обчислення R6:=2\*(R5-R2-1)----------------------

;Перетворення R2 в ДК

Mov A, R2 ;A:=R2

Cpl A ;Інверсія А

Inc A ;А:=A+1 --кінець перетворення в ДК

Add A, R5 ;A:=R5-R2

Dec A ;A:=R5-R2-1

Clr C ;Очистка ознаки переносу С

Rlc A ;Множення на 2(зсув вліво)

Mov R6, A ;R6:=2\*(R5-R2-1)

;----------------------------------------------------------------------

Jf0 L2 ;Перевірка ознаки F0 на 0

Mov A, R6 ;A:=R6

Add A, #77H ;A:=R6+77h

Mov R1, A ;R1:=R6+77h

L2:;Мітка переходу, якщо f0=0

Mov A, R0 ;A:=R0

Dec A ;A:=R0-1

Clr C ;Очистка ознаки переносу С

Rrc A ;Зсув акамулятора вправо (ділення на 2)

Mov R6, A ;R6=(R0-1)/2

Clr A ;Очистка А

Add A, R6 ;Копіювання R6 в А

Inc A ;A:=R6+1

Mov R5, A ;R5:=R6+1

Ret ;Кінець підпрограми