**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”**

**Л А Б О Р А Т О Р Н А Р О Б О Т А № 1**

***ВИВЧЕННЯ СИСТЕМИ КОМАНД МІКРОКОНТРОЛЕРА МК48***

**з дисципліни:**

**«Архітектура комп’ютерів»**

Виконав:

Студент ФІОТ

Групи ІО – 31

Долинний Олександр

**Київ**

**2015**

**Лабораторна робота №1**

ВИВЧЕННЯ СИСТЕМИ КОМАНД МІКРОКОНТРОЛЕРА МК48

**Мета завдання:**

Вивчення структурної схеми МК48, системи та форматів команд. Отримання навиків складання алгоритмів та розробки програм виконання арифметичних операцій для мікроконтролера МК48.

Варіант завдання: 311010 = 110 000 100 1102

h2 h1 X1 X2 X3 X4 X5 X6 h4 h3 Порт

1 0 45 05 3С А0 22 12 1 0 BUS

h5 h1 Вираз для обчислення

0 0 F= 4(X1 + X2 – 1) – (X3 + X4) – (X5 & X6)/4

1. **Додаткові теоретичні відомості**

**Структура програми на мові асемблера MK48**

Текст програми на мові асемблера MK48 має певну структуру. Кожна команда (і псевдокоманда) записуються в рядок наступної конструкції:

МІТКА: ОПЕРАЦІЯ ОПЕРАНД(И) ; КОМЕНТАР

Поля можуть відділятися один від одного довільним числом пробілів.

**Мітка**

У полі мітки розміщується символічне ім'я комірки пам'яті, в якій зберігається відмічена команда або операнд. Мітка є буквеноцифровою комбінацією, що починається з букви. Використовуються тільки букви латинського алфавіту. Асемблер МК48 допускає використання в мітках символу підкреслення ( \_ ). Довжина мітки не повинна перевищувати шість символів. Мітка завжди завершується двокрапкою ( : ).

**Операція**

В полі операції записується мнемонічне позначення команди МК48 або псевдокоманди асемблера, яке являється скороченням (абревіатурою) повної англійської назви виконуючої дії.

Наприклад:

MOV – Move – перемістити;

JMP – Jump – перейти;

DB – Define Byte – визначити байт.

Для МК48 використовується строго визначений та обмежений набір мнемонічних кодів. Будь-який інший набір символів, що розміщений в полі операції, сприймається асемблером як помилковий.

**Операнди**

В цьому полі визначаються операнд (або операнди), які приймають участь в операції. Команди асемблера можуть не мати операндів або мати один чи два операнди. Операнди розділяються комою

( , ).

Операнд може бути заданий безпосередньо або його адресою (прямою або непрямою).

Використані в якості операндів символічні імена та мітки повинні бути визначені, а числові значення – подані з вказанням на систему числення. Для вказання системи числення використовується суфікс – буква, яка розміщується після числа. Число без суфіксу вважається десятковим. Застосовуються наступні суфікси:

В – двійкова система обчислення;

Q – вісімкова система обчислення;

D – десяткова система обчислення;

Н – шістнадцятирічна система обчислення.

**Обробка виразів в процесі трансляції**

Вираз представляє собою сукупність символічних імен та числових значень, що пов’язані операторами асемблера. Оператори асемблера забезпечують виконання в форматі двобайтових слів наступних операцій:

арифметичних операцій: логічних операцій:

+ – додавання; ORL – АБО;

– – віднімання; ANL – І;

\* – множення; XRL – виключне АБО;

/ – ділення; NOT – заперечення.

MOD ділення за модулем;

Наприклад, запис ADD A,#((NOT 13)+1) еквівалентний запису ADD A,#0F3H та забезпечує додавання до вмісту акумулятора числа 13, що подане доповнювальним машинним кодом.

**Коментар**

Поле коментаря застосовується для текстового чи символьного пояснення логічної організації прикладної програми. Поле коментаря ігнорується асемблером, тому тут допускається використання будь-яких символів. За правилами мови асемблера поле коментарію починається після символа крапки з комою ( ; ).

**Псевдокоманди асемблера**

Спеціальна програма асемблер транслює вихідну програму в об’єктні коди. Асемблер бере на себе багато рутинних задач програміста, таких як присвоєння дійсних адрес, перетворення чисел, присвоєння дійсних значень символьним змінним і таке інше. Програміст вказує асемблеру наступні параметри: початкову адресу прикладної програми, кінець програми асемблера, формати даних. Ця інформація додається у вихідний текст прикладної програми у вигляді псевдокоманд (директив) асемблера, які управляють процесом трансляції і не перетворюються в коди об’єктної програми.

**Застосування псевдокоманд:**

– псевдокоманда ORG 10Н задає асемблеру адресу комірки пам’яті 10h, в якій розміщується наступна команда прикладної програми;

– псевдокоманда EQU будь-якому символічному імені, яке використовується в програмі, ставить у відповідність певний операнд; наприклад, запис PET EQU 13, означає, що під час асемблювання символічне ім’я PET, буде замінено числом 13.

– псевдокоманда END дає асемблеру вказівку закінчити трансляцію.

1. **Лістинг:**

;Function = 4\*(X1+X2-1)-(X3+X4)-(X5&X6)/4

Sel Rb0

Mov R0, #01H

Mov R7, #06H

Input:

In A,BUS

Mov @R0, A

Inc R0

Djnz R7, Input

;R1=X1+X2-1

Clr C

Sel Rb0

Mov A, R1

Addc A, R2

Dec A

Mov R1, A

Sel Rb1

Mov A, R1

Addc A, R2

Mov R1, A

;R1=2\*R1=2\*(X1+X2-1)

Clr C

Sel Rb0

Mov A, R1

Rlc A

Mov R1, A

Sel Rb1

Mov A, R1

Rlc A

Mov R1, A

;R1=2\*R1=4\*(X1+X2-1)

Clr C

Sel Rb0

Mov A, R1

Rlc A

Mov R1, A

Sel Rb1

Mov A, R1

Rlc A

Mov R1, A

;R3=X3+X4

Clr C

Sel Rb0

Mov A, R3

Addc A, R4

Mov R3, A

Sel Rb1

Mov A, R3

Addc A, R4

Mov R3, A

;R5=X5&X6

Clr C

Sel Rb0

Mov A, R5

Anl A, R6

Mov R5, A

Sel Rb1

Mov A, R5

Anl A, R6

Mov R5, A

; R5=R5/2=(X5&X6)/2

Clr C

Sel Rb1

Mov A, R5

Rrc A

Mov R5, A

Sel Rb0

Mov A, R5

Rrc A

Mov R5, A

; R5=R5/2=(X5&X6)/4

Clr C

Sel Rb1

Mov A, R5

Rrc A

Mov R5, A

Sel Rb0

Mov A, R5

Rrc A

Mov R5, A

;R3=-R3=-(X3+X4)

Clr C

Sel Rb0

Mov A, R3

Cpl A

Addc A, #1

Mov R3, A

Sel Rb1

Mov A, R3

Cpl A

Addc A, #0

Mov R3, A

;R5=-R5=-((X5&X6)/4)

Clr C

Sel Rb0

Mov A, R5

Cpl A

Addc A, #1

Mov R5, A

Sel Rb1

Mov A, R5

Cpl A

Addc A, #0

Mov R5, A

;R1=R1+R3 = 4\*(X1+X2-1)-(X3+X4)

Clr C

Sel Rb0

Mov A, R1

Addc A, R3

Mov R1, A

Sel Rb1

Mov A, R1

Addc A, R3

Mov R1, A

;R1=R1+R5 = 4\*(X1+X2-1)-(X3+X4)-(X5&X6)/4

Clr C

Sel Rb0

Mov A, R1

Addc A, R5

Mov R1, A

Sel Rb1

Mov A, R1

Addc A, R5

Mov R1, A

Nop

End

**Висновки:**

В ході виконання лабораторної роботи була вивчена структурна схема МК48, а також система команд контролера. Також на прикладі обчислення функції були закріплені навички складання алгоритмів, а також розробки програм виконання арифметичних операцій для контролера МК48.