Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Архітектура комп'ютерів-2. Процесори

Домашня

Модульна контрольна робота

Виконала:

студентка групи ІО-64

Бровченко А.В.

Залікова книжка №6403

Перевірила Ткаченко В.В.

Київ

2018 р.

**Завдання 1.**

***Мета роботи:*** Вивчення системи команд, форматів подання даних та способів адресації операндів; вивчення команд передачі управління, команд пересилки даних та команд переключення між банками регістрів та ПП; отримання навиків розробки програм на мові асемблеру МК51.

1. **Теоретичні відомості**

Мікроконтролер містить резидентну пам'ять програм (РПП) та резидентну пам'ять даних (РПД); пристрій управління і синхронізації, до складу якого входить лічильник команд, регістр команд і регістр ознак; арифметико-логічний пристрій, до складу якого входить АЛБ, акумулятор і регістри; блок таймерів-лічильників та блок послідовного інтерфейсу і переривань. Обмін даними здійснюється через чотири порти *Р*0*, Р*1*, Р*2*, Р*3, або через послідовний порт.

**Резидентна пам’ять програм**

Резидентна пам’ять програм, має ємність 4Кб. Призначена для зберігання команд, констант, управляючих слів ініціалізації, таблиць кодування вхідних і вихідних змінних. Резидентна пам’ять даних підключена до шістнадцятибітної шини адреси, що надходить з лічильника команд, або регістру покажчика даних.

**Резидентна пам’ять даних**

Резидентна пам’ять даних призначена для зберігання змінних у процесі виконання програми, адресується одним байтом і має ємність 128 байт. До адресного простору резидентної пам’яті данихналежать регістри спеціальних функцій.

**Система команд мікроконтролера КР1816ВЕ51**

Система команд мікроконтролера КР1816ВЕ51 значно ширша і потужніша ніж система команд КР1816ВЕ48, за рахунок команд множення, ділення, віднімання, операцій над бітами, операцій зі стеком, розширеного набору команд передачі управління.

Система команд мікроконтролера містить сто одинадцять команд. Відносно функціональних ознак команди класифікуються за наступними групами:

* команди передачі даних;
* команди виконання арифметичних операцій;
* команди виконання логічних операцій;
* команди виконання операцій з бітами;
* команди передачі управління.

Команди МК51 мають довжину один, два або три байти і виконуються відповідно за один, два або чотири машинні цикли. За тактової частоти генератора *fr* = 12Мгц тривалість циклу складає 1 мкс.

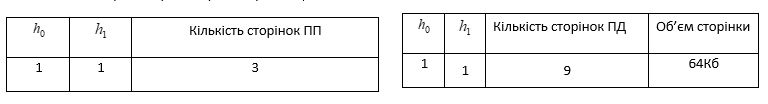
Можливі тринадцять форматів команд мікроконтролера. Перший байт містить код операції (КОП), другий і третій – адреси операндів або безпосередньо операнди.

Операнди можуть бути чотирьох типів:

* *однобітні операнди* (біти), в якості яких можливо застосовувати окремі біти регістрів спеціальних функції *SFR* та портів; для адресації бітів застосовується пряма восьмибітна адреса (*bit*); непряма адресація бітів неможлива;
* *чотирибітні операнди* (тетради) застосовуються під час виконання операцій обміну тетрадами *SWAP* та *XCHD*;
* *восьмибітні операнди* (байти), являють собою комірки пам’яті програм або даних, константи – безпосередні операнди, регістри спеціальних функцій, порти вводу/виводу; порти та регістри спеціальних функцій *SFR* адресуються тільки прямим способом; байти пам’яті можуть адресуватися і непрямим способом, за допомогою адресних регістрів *R*0, *R*1, *DPTR*, *PC*.
* *двобайтні операнди* це константи та прямі адреси, для подання яких застосовуються другий і третій байти команди.

1. **Розробити структурну схему підключення до МК51 заданої кількості сторінок пам’яті даних та пам’яті програм. Кількість сторінок обрати з табл. 6.17 та табл. 6.18.**

**Номер залікової книжки: (6403)10 = (1100100000011)2**

**h0 = 1, h1 = 1, h2 = 0, h3 = 0, h4 = 0**

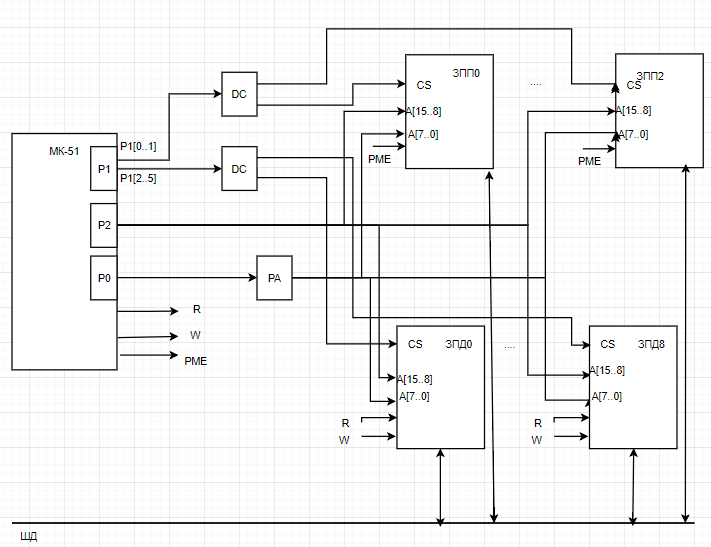


Рис. 1. Структурна схема підключення сторінок ПП і ЗПД

1. **Розробити алгоритм та програму на асемблері МК51 для пересилки масиву вихідних даних, що складається з заданої кількості байтів (табл. 6.20) у резидентну пам’ять даних МК51 розпочинаючи з комірки за адресою заданою у табл. 6.21. Розробити алгоритм обчислення функції та програму на асемблері МК51. Записати значення аргументів у заданий банк регістрів (табл. 6.19), в якому виконати обчислення функції. Результат обчислення розмісти у регістрах, заданих у табл. 6.22.**

****

**Код програми**

;Записуємо початкову адресу з якої ;почнеться прийом байтів

MOV R0, #4Ah

MOV R7,#1Eh ; 1Eh - 30 бітів

;Пересилаємо з порту Р0 масив

send: MOV A, P0

MOV @R0, A

INC R0

DJNZ R7, send

MOV PSW, #00000000b ;Скидуємо PSW

MOV R7, #6 ;Кількість байтів для ;збереження в регістри

MOV R0, #9h ;Адреса регістру R1 в ;БР1

MOV R1, #20h ;Початкова адреса ;пересилки

; Пересилаємо 6 байтів в регістри БР1

label2:

MOV A, @R1

MOV @R0, A

INC R0

INC R1

DJNZ R7, label2

Mov PSW, #00001000b ;Вибираємо БР1

;F2 = (8\*(X1 OR X2)\*(X3-X4)- ;X5/X6)/16

;R1 = X1 R2 = X2, X1 ORL X2

; Під час написання програми було ;виявлено, що R1 неможливо

; перемістити в А (компілятор дає ;помилку).

; Тому, щоб не перероблювати код ;програми було прийняте рішення

; Перемістити R1 в В, а потім B в R1

MOV B, R1

MOV A, B

ORL A, R2 ; Логічне АБО операндів Х1 ;та Х2

MOV R1, A

;R3 = X3 R4 = X4, X3-X4

CLR C

;Перевід в ДК, оскільки Х3 від’ємне

MOV A, R3

XRL A, #0FFH

INC A

SUBB A, R4; Віднімання операндів Х3 ;та Х4

; Переводимо результат в звичайний ;вигляд

XRL A, #0FFH

INC A

MOV R3, A; Результат в R3

;R5 = X5 R6 = X6, X5/X6

;Перевід в ДК, оскільки Х5 від’ємне

MOV A, R5

XRL A, #0FFH

INC A

;Кінець переводу в ДК A = Х5 В = Х6

MOV B, R6

DIV AB ; Ділення операндів X5 та Х6

; Переводимо результат в звичайний ;вигляд

XRL A, #0FFH

INC A

MOV R5, A

; R1 = X1 ORL X2 R3 = X3-X4

CLR C

; Та ж проблема, що була описана вище

MOV B, R1

MOV A, B

; Переводимо в ДК

XRL A, #0FFH

INC A

MOV B, R3

MUL AB ; Множення двох операндів R1 ;та R3

;Переводимо в нормальний вигляд

XRL A, #0FFH

XRL B, #0FFH

INC A

MOV R2, A ; Результат вже буде ;записуватись у два регістра

MOV R0, B ; R0 - Старші розряди, R2 - ;молодші

; Операція множення на 8(реалізована ;зсувами)

MOV R3, #3h ; в R3 кількість зсувів - ;3

; Оскільки число від'ємне знову в ДК, ;інкрементуєм тільки молодші розряди

MOV A, R2

XRL A, #0FFH

INC A

MOV R2, A

MOV A, R0

XRL A, #0FFH

MOV R0, A

;Цикл зсуву числа вліво 3 рази

CYCLEMUL: MOV A, R2

RLC A

MOV R2, A

MOV A, R0

RLC A

MOV R0, A

DJNZ R3, CYCLEMUL

; Перетворюємо число до нормального ;вигляду

MOV A, R2

XRL A, #0FFH

INC A

MOV R2, A

MOV A, R0

XRL A, #0FFH

MOV R0, A

;Число знаходиться в R0.R2, R0.R2 = ;R0.R2 - R5

CLR C

MOV A, R2

SUBB A, R5 ; Робимо віднімання від ;молодших розрядів.

JNC WITHOUT ; Якщо йде недостача ;декрементуємо старший розряд

DEC R0

WITHOUT:

MOV R2, A ; Записуємо результат

; R0.R2 = R0.R2 / 16

CLR C

MOV R3, #4h ; /16 - 4 зсуви вправо

cycle:

MOV A, R0

RRC A

MOV R0, A

MOV A, R2

RRC A

MOV R2, A

DJNZ R3, cycle

; Перезаписуємо результат в БР2 в ;R4.R3 БР2

MOV A, R0

MOV 13H, A ; Адреса регістру R4 БР2

MOV A, R2

MOV 12H, A ; Адреса регістру R3 БР2

END

**Завдання 2.**

**Мета роботи:** Вивчення структури, режимів роботи, системи команд і отримання навиків розробки програм, що управляють, для мікроконтроллера КМ1816ВЕ451.

1. **Теоретичні відомості**

**Таймери / лічильники**

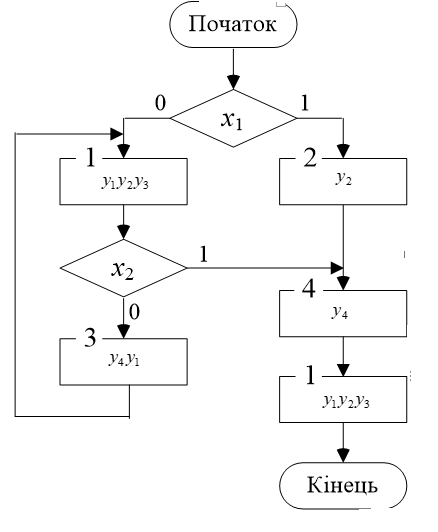
МК 51 має два 16-розрядних таймера / лічильника - Т / ЛØ, Т / Л1для формування тимчасових затримок на виходах портів або підрахунку зовнішніх подій. Під час роботи в якості таймера, в кожному машинному циклі виконується інкрементувания вмісту таймера / лічильника з частотою f\_ч / 12, де f\_ч - частота тактового генератора, оскільки машинний цикл складається з 12 періодів частоти синхронізації. Під час роботи в якості лічильника вміст таймера / лічильника інкрементується на кожному переході зовнішнього сигналу З «1» та «0». Для управління режимами роботи Т / Сч і для зв'язку з системою переривань використовуються (регістра спеціальних функцій) TMOD і TCON.

**Режим роботи «0»**

В цьому режимі Т / Сч схожий на Т / Сч ВЕ48 (8 бітний лічильник), на входи якого підключається тривалість частоти 32.В режимі «0» Т / Сч є тринадцятирозрядний лічильник, в якому послідовно з'єднані п’ятирозрядний регістр TL1 і восьмирозрядний регістрTH1. залежно від розряду С / Т1 регістра TMOD на вхід лічильника надходять зовнішні сигнали зі входу Т1 (лічильник) або сигнал таймер. Рахунок починається з установки біта TR регістра TCON. Управління рахунком ззовні здійснюється за допомогою біта GATE регістра TMOD. Рахунок дозволений при встановленні значення вхідного сигналу INT1 = 1 і заборонено при INT0 = 0. У момент переповнення Т / Сч встановлюється ознака TF1.Прі переповненні лічильник ТН змінює стан, при зміні стану лічильника З FFh на 00;

**Номер залікової книжки: (6403)10 = (1100100000011)2**

**h1 = 1, h2 = 1, h3 = 0, h4 = 0, h5 = 0**

****

**Код програми**

; Підготовка даних. В порт Р2 ;виставляємо ікси

MOV P2, #C0H

MOV TMOD, #00H ; Установка режиму "0" ;для таймера

MOV A, P2

CHECKX1:

JB ACC.6, BLOCK2 ; Перевірка Х1

BLOCK1:

; підготовка таймеру

MOV 8DH, #00H

MOV 8BH, #F0H ; Для сигналу Y1 ;затримка буде становити (560-12)/32 ;~ 17

; (-17)дк = F0H

MOV R7, #FFH ;Затримка сигналу Y2 ;становить 12, 12:2 ~ 6

;Виставляємо сигнали Y1Y2Y3

ORL P2, #6H

; Затримка для сигналу Y2

CYCLE1:

DJNZ R7, CYCLE1

; Обнуляємо Y2

ANL P2, #C5H

; Включаємо на таймері затримку для Y1

SETB TR1

CYCLE2:

JNB TF1, CYCLE2

; Обнуляємо сигнал Y1

ANL P2, #C4H

; Виставляємо таймер під сигнал Y3

MOV 8DH, #00H

MOV 8BH, #FAH ; Затримка Y3 = (700-560)/32 = 7 (-7)дк=FAH

; Включаємо таймер

SETB TR1

CYCLE3:

JNB TF1, CYCLE3

;Обнуляємо останній сигнал

ANL P2, #C0H

;Перевірка Х2

CHECKX2:

JB P2.7, BLOCK4 ; Перевірку можна ;проводити і не записуючи в А

BLOCK3:

; Встановлення таймеру для Y4

MOV 8DH, #00H

MOV 8BH, #F7H ; Затримка на Y4 = ;330/32 ~ 10 (-10)дк = F7H

MOV P2, #C9H ; Вмикаємо сигнали Y1, Y4

;Вмикаємо таймер для Y4

SETB TR1

CYCLE1Y4:

JNB TF1, CYCLE1Y4

;Обнулення сигналу Y4

ANL P2, #C1H

; Встановлення таймеру для Y1

MOV 8DH, #00H

MOV 8BH, #FAH ; Затримка на Y1 = (560-330)/32 ~ 7 (-7)дк = FAH

; Вмикаємо таймер

SETB TR1

CYCLE2Y1:

JNB TF1, CYCLE2Y1

;Обнуляємо Y1

ANL P2, #C0H

; Оскільки ми взамкнутому циклі ;переходимо в BLOCK1

SJMP BLOCK1

BLOCK2:

; Виставляємо затримку в 10 для Y2, а ;також ставим його на порті

MOV R7, #6H

MOV P2, #C2H

; Затримка для Y2

CYCLEY2:

DJNZ R7, CYCLEY2

; Обнулюємо сигнал Y2

ANL P2, #C0H

BLOCK4:

; Встановлення таймеру для Y4

MOV 8DH, #00H

MOV 8BH, #EFH ; Затримка = 330/32 ~ 10 (-10)дк = F7H

MOV P2, #C8H ; Виставляємо Y4 на ;порті

;Вмикаємо таймер

SETB TR1

CYCLEY4:

JNB TF1, CYCLEY4

;Обнулюємо Y4

ANL P2, #C0H

;Код, який реалізований нижче є ;копією коду з BLOCK1

; оскільки нам потрібно знову ;викликати затримку сигналів Y1Y2Y3

; Якщо я не помиляюсь в одному з ;прикладів я бачив, що даний код

; можна було би просто замінити на CALL BLOCK1 а в BLOCK1 додати

; в кінці RET, але у нас немає стеку ;тому користуємось звичайним

; кодом.

; підготовка таймеру

MOV 8DH, #00H

MOV 8BH, #F0H ; Для сигналу Y1 ;затримка буде становити

;(-17)дк = F0H

MOV R7, #FFH ;Затримка сигналу Y2 ;становить 12, 12:2 ~ 6

;Виставляємо сигнали Y1Y2Y3

ORL P2, #6H

; Затримка для сигналу Y2

CYCLEL1:

DJNZ R7, CYCLEL1

; Обнуляємо Y2

ANL P2, #C5H

; Включаємо затримку для Y1

SETB TR1

CYCLEL2:

JNB TF1, CYCLEL2

; Обнуляємо сигнал Y1

ANL P2, #C4H

; Виставляємо таймер під сигнал Y3

MOV 8DH, #00H

MOV 8BH, #FAH ;Затримка Y3=(-7)дк=FAH

; Включаємо таймер

SETB TR1

CYCLEL3:

JNB TF1, CYCLEL3

;Обнуляємо останній сигнал

ANL P2, #C0H

END