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

Виконав:

студент групи ІО-62

Копійка А.О.

Залікова книжка №6212

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: (6212)10 = (1100001000100)2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Кількість сторінок ПД | Об’єм сторінки | | |
| 0 | 0 | 10 | | 256б |

**h0 = 0, h1 = 0, h2 = 1, h3 = 0, h4 = 0, h5 = 0**

|  |  |  |
| --- | --- | --- |
|  |  | Кількість сторінок ПП |
| 0 | 0 | 4 |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Банк регистрів | Функція для обчислення |
| 1 | 0 | БР3 | *F*3 = |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | Адреса початкової комірки пам’яті |
| 0 | 0 | 1 | 48*h* |

|  |  |  |
| --- | --- | --- |
|  |  | Розмірність масиву |
| 0 | 0 | 30 |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Регістри | Банк регістрів |
| 0 | 1 | *R*5.*R*6 | БР2 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | *X*1 | *X*2 | *X*3 | *X*4 | *X*5 | *X*6 | *Х*7 | *X*8 | *X*9 … *X*30 |
| 0 | 0 | –7 | 12 | –17 | 3 | 5 | *FF* | \* | \* | \* … \* |

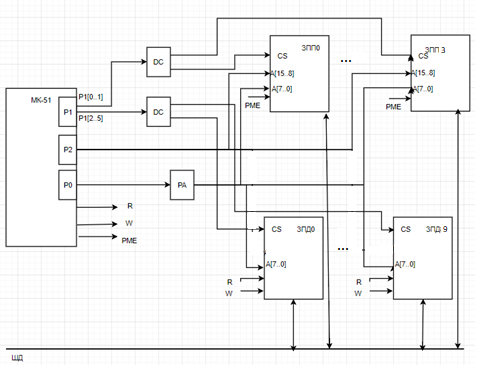


Рис. 1. Структурна схема підключення сторінок ПП і ЗПД

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

MOV R0, #48h

MOV R7,#1Eh – ;1Еh – 31 біт

;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, #18h ;Адреса регістру R1 в ;БР3

MOV R1, #48h ;Початкова адреса ;пересилки

label2:

MOV A, @R1

MOV @R0, A

INC R0

INC R1

DJNZ R7, label2

Mov PSW, #00001100b ;Вибираємо БР3

;F3 = (16\*(X1 -1)and X2 -(X3+X4)\* ;\*X5)/2\*X6

;R0 = X1, 16\*(X1 -1)

MOV A, R0

INC A ;X1-1

;Оскільки Х1 задане від'ємне для ;отримання правильного результату

;необхідно перевести його в ДК

;Перевід в ДК

MOV A, R0

XRL A, #0FFH

INC A

MOV R0, A

;для множення на 16 виконаємо

;зсув в ліво 4 рази

; R6. R0=16\*(X1 -1)

MOV R7, #4h

CYCLEMUL:

MOV А, R0

RLC A

MOV R0, A

MOV A, R6

RLC A

MOV R6, A

DJNZ R7, CYCLEMUL

;R1 = X2 , 16\*(X1 -1)AND X2

MOV А, R0

AND A, R1

MOV R1,A

; Результат в R1

;R2 = X3 R3 = X4, X3+X4

MOV A, R2

ADD A, R3

; Результат в R2

MOV R2, A

; R4 = X5

; R2. R4 = (X3+X4)\*X5

; X3+X4 від’ємне

MOV A, R2

; Переводимо в ДК

XRL A, #0FFH

INC A

MOV B, R4

MUL AB

; Множення двох операндів R2 та R4

;Переводимо в ПК

XRL A, #0FFH

XRL B, #0FFH

INC В

MOV R2, A

MOV R4, B

!!!!; 16\*(X1 -1)AND X2 -(X3+X4)\*X5

; (X3+X4)\*X5 - від’ємне

MOV A, R2

XRL A, #0FFH

MOV B, R4

XRL B, #0FFH

INC B

MOV R2, A

MOV R4, B

;виконаємо віднімання від R2.R4 R1

;отриманий результат інвертуємо

MOV A, R4

SUBB A, R1

; Робимо віднімання від ;молодших розрядів.

MOV R4, B

;оскільки нам потрібно буде ;повертатись до ПК, то робити ;інвертування двічі не будемо

INC В

MOV R4, B

; R2.R4/X6

; R5=X6

LOOPDIV:

MOV A, R4

RRC A

MOV R4, A

MOV A, R2

RRC A

MOV R2, A

DJNZ R5, LOOPDIV

; Перезаписуємо результат з БР3 в ;R5.R6 БР3

MOV A, R2

MOV 23H, A ; Адреса регістру R5 БР0

MOV A, R4

MOV 24H, A ; Адреса регістру R6 БР0

END

**Завдання 2.**

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

**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».

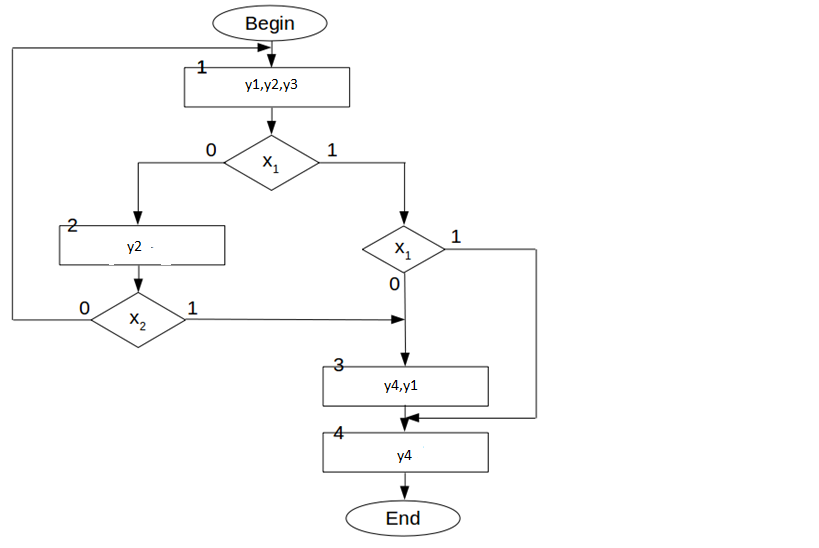
Для управління режимами роботи Т / Сч і для зв'язку з системою переривань використовуються (регістра спеціальних функцій) TMOD і TCON.

**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;

|  |  |  |
| --- | --- | --- |
| *h2* | *h1* | *Вихідний алгоритм* |
| 1 | 0 | Рис. 6.2, в |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | *h3*  *h2* | *Управляючі сигнали* | | | |
| *Номер операторної вершини* | | | |
| 1 | 2 | 3 | 4 |
| 1 | 0 |  |  |  |  |

****

**3. Розробити програму управління для МК51, що реалізує заданий алгоритм. Для вводу/виводу даних використати порти, задані в табл. 6.8. Часові параметри управляючих сигналів задані в табл. 6.9.**

|  |  |  |
| --- | --- | --- |
| *h2* | *h4* | *Порт* |
| 1 | 0 | Р2 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | *h5*  *h4* | *h1* | *Часові затримки* | | | | |
| *Управляючі сигнали* | | | | |
| *y1* | *y2* | *y3* | *y4* | *y5* |
| 0 | 0 | 0 | 720 | 25 | 400 | 10 | 800 |

**Код програми**

; Підготовка даних. В порт Р2 ;виставляємо ікси

MOV P2, #C0H

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

BLOCK1:

; підготовка таймеру

MOV 8DH, #00H

MOV 8BH, #F4H ; Для сигналу Y3 ;затримка буде становити (400-25)/32 ;~ 12

; (-12)дк = F4H

MOV R7, #DH ;Затримка сигналу Y2 ;становить 25, 25:2 ~ 13

;Виставляємо сигнали Y1Y2Y3

ORL P2, #С7

; Затримка для сигналу Y2

CYCLE1:

DJNZ R7, CYCLE1

; Обнуляємо Y2

ANL P2, #C5H

; Включаємо на таймері затримку для Y3

SETB TR1

CYCLE2:

JNB TF1, CYCLE2

; Обнуляємо сигнал Y3

ANL P2, #C1H

; Виставляємо таймер під сигнал Y1

MOV 8DH, #00H

MOV 8BH, #F6H ; Затримка Y3 = (720-400)/32 = 10 (-10)дк=F6H

; Включаємо таймер

SETB TR1

CYCLE3:

JNB TF1, CYCLE3

;Обнуляємо останній сигнал

ANL P2, #C0H

MOV A, P2 ;

CHECKX1\_first:

JB ACC.6, CHECKX1\_second; Перевірка Х1 вперше

BLOCK2:

; Виставляємо затримку в 25 для Y2, а ;також ставим його на порті

MOV R7, #DH

MOV P2, #C2H

; Затримка для Y2

CYCLEY2:

DJNZ R7, CYCLEY2

; Обнулюємо сигнал Y2

ANL P2, #C0H

;Перевірка Х2

CHECKX2:

JB P2.7, BLOCK1 ; Перевірку можна ;проводити і не записуючи в А

CHECKX1\_second:

JB P2.6, BLOCK4 ; Перевірка Х1 вдруге

BLOCK3:

; Встановлення таймеру для Y1

MOV 8DH, #00H

MOV 8BH, #E9H ; Затримка на Y1 = ;(720-10)/32 ~ (-23)дк = E9H

MOV R7, #DH ;Затримка сигналу Y4 ;становить 10, 10:2 ~ 5

MOV P2, #C9H ; Вмикаємо сигнали Y1, Y4

;Вмикаємо таймер для Y4

CYCLE1Y4:

DJNZ R7, CYCLE1Y4

; Обнуляємо Y4

ANL P2, #C1H

; Включаємо на таймері затримку для Y1

SETB TR1

CYCLE2Y1:

JNB TF1, CYCLE2Y1

; Обнуляємо сигнали

ANL P2, #C0H

BLOCK4:

; Виставляємо затримку в 10 для Y4, а ;також ставим його на порті

MOV R7, #5H

MOV P2, #C8H; Виставляємо Y4 на ;порті

; Затримка для Y4

CYCLEY4:

DJNZ R7, CYCLEY4

; Обнулюємо сигнал Y4

ANL P2, #C0H

END