

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
"КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ"
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
Кафедра обчислювальної техніки

РОЗРАХУНКОВА РОБОТА

по курсу „Архітектура комп’ютерів - 2. Процесори”

Виконав: Долинний Олександр Валерійович

Група ІО-31, Факультет ІОТ,

Залікова книжка № 3110

Номер технічного завдання 110000100110

(підпис керівника)

Київ – 2015 р.

Зміст

1. Склад МПС та основні її функції	3
1.1. Опис мікроконтролеру.....	3
1.2. Підімкнення зовнішньої пам'яті програм	4
1.3. Підімкнення зовнішньої пам'яті даних	6
1.4. Контролер пріоритетних переривань (КПП).....	7
1.5. Контролер прямого доступу до пам'яті (КПДП).....	9
1.6. Додаткові порти і ПЗА	10
2. Програмна частина.....	12
2.1. Приклад переходу на підпрограму	12
2.2. Приклад формування часової затримки	15
2.3. Приклад виконання операції множення 1-им способом.....	15
2.4. Обчислення функції за варіантом	16
3. Структурна схема системи.	18
4. Висновки	19
5. Список літератури.	20

1. Склад МПС та основні її функції

У розрахунковій роботі розроблюється мікропроцесорна система (МПС), ядром якої є мікроконтролер МК1816BE48.

До складу розроблюваної МПС входять мікроконтролер(МК), зовнішня пам'ять програм (ЗПП), зовнішня пам'ять даних(ЗПД), 24 зовнішні пристрої (ЗП), контролер пріоритетних переривань (КПП), контролер прямого доступу до пам'яті (КПДП), додаткові порти P4, P6, P7 та програмований зв'язуючий адаптер KP580BV51.

Об'єм зовнішньої пам'яті програм 4Кб, зовнішньої пам'яті даних 8Кб. Шини адреси та даних розділені, КПП та КПДП децентралізовані. Кожен ЗП має фіксовану адресу в адресному просторі периферійних пристроїв.

1.1. Опис мікроконтролеру

Мікроконтролер МК1816BE48 має 8-розрядну операційну частину.

Обсяг резидентної ПП складає 1Кб, обсяг ПД складає 64 комірки по 1 байту.

Структурна схема МК48 наведена на рис. 1.1.



Рис. 1.1 – Структурна схема МК48

Резидентна ПД МК має в своєму складі 2 банки робочих регістрів (адреси 0-7 та 24-31), по 8 регістрів в кожному банку. Структура регістрів (карта адрес) пам'яті даних МК представлена на рис. 1.2:

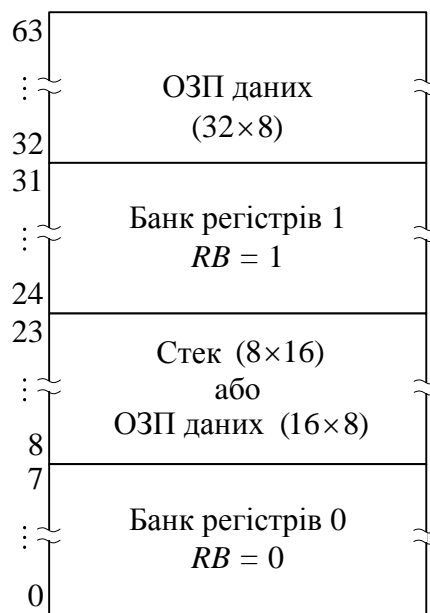


Рис. 1.2. Карта адрес пам'яті даних

1.2. Підімкнення зовнішньої пам'яті програм

Пам'ять програм розділяється на резидентну, розташовану всередині ІС, та зовнішню, для реалізації якої необхідні додаткові ІС пам'яті. Максимальний адресний простір пам'яті програм складає 4 Кб. Резидентна пам'ять програм займає адреси від 0 до 1023.

Пам'ять програм розглядається як два банки – нульовий банк програм і перший банк програм. Якщо встановлюється розряд лічильника команд $PC[11] = 0$, то вибір слів здійснюється із нульового банку пам'яті (адреси від 0 до 2047), і, якщо $PC[11] = 1$ – із першого банку пам'яті (адреси від 2047 до 4095). Вибір банку пам'яті здійснюється командами *SEL MBO* та *SEL MB1*, які встановлюють ознаку *MB* вибору банку пам'яті.

Карта розподілу адрес пам'яті програм показана на рис. 1.3.

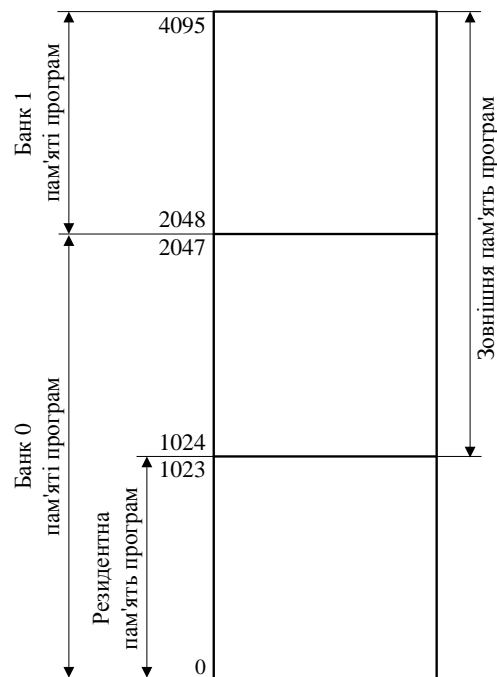


Рис. 1.3 Карта розподілу адрес пам'яті програм

Підключення ПП відбувається через порт P2 (на схемі сигнали 17...21), по розряду на вхід CS кожної сторінки.

Для підключення зовнішньої пам'яті програм використовуються виходи портів P2[3..0] та BUS[7..0]. Для зберігання адреси звернення до пам'яті використовується зовнішній регістр адреси PA. Схема підключення зображена на рис 1.4.

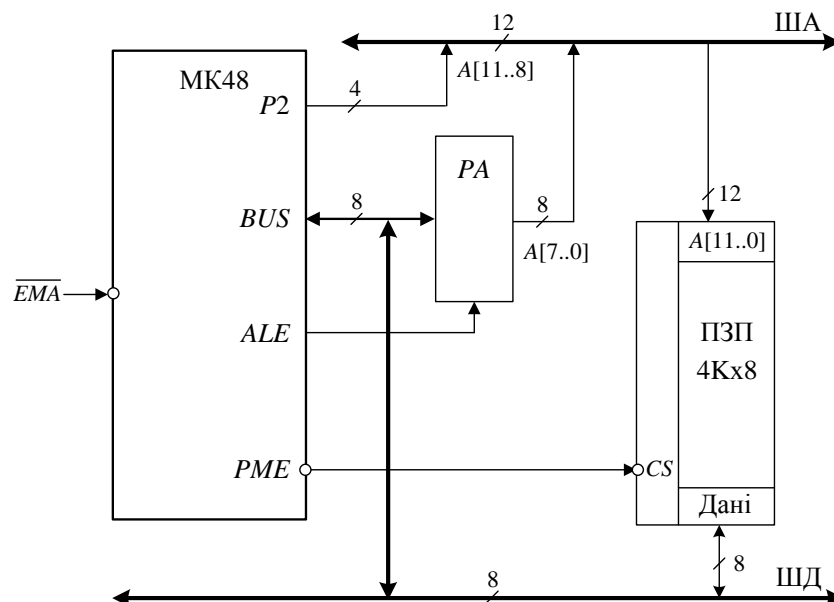


Рис. 1.4. Підключення зовнішньої пам'яті програм

У даній роботі до мікроконтролера підключається 4Кб зовнішньої пам'яті програм, 4 модулі по 1 Кб. Вибір сторінки здійснюється чотирма молодшими розрядами порту P2[3..0].

1.3. Підімкнення зовнішньої пам'яті даних

Підключення зовнішньої ПД відбувається через порт P1 з використанням дешифратора. Розряди 27...58 підходять до входів CS відповідних сторінок пам'яті. До ПД32 підключений селектор адрес для ЗП1 – ЗП24, та ППА. Карта розподілення пам'яті зображена на рис. 1.6. Загальна схема підключення зовнішньої пам'яті даних на рис. 1.5.

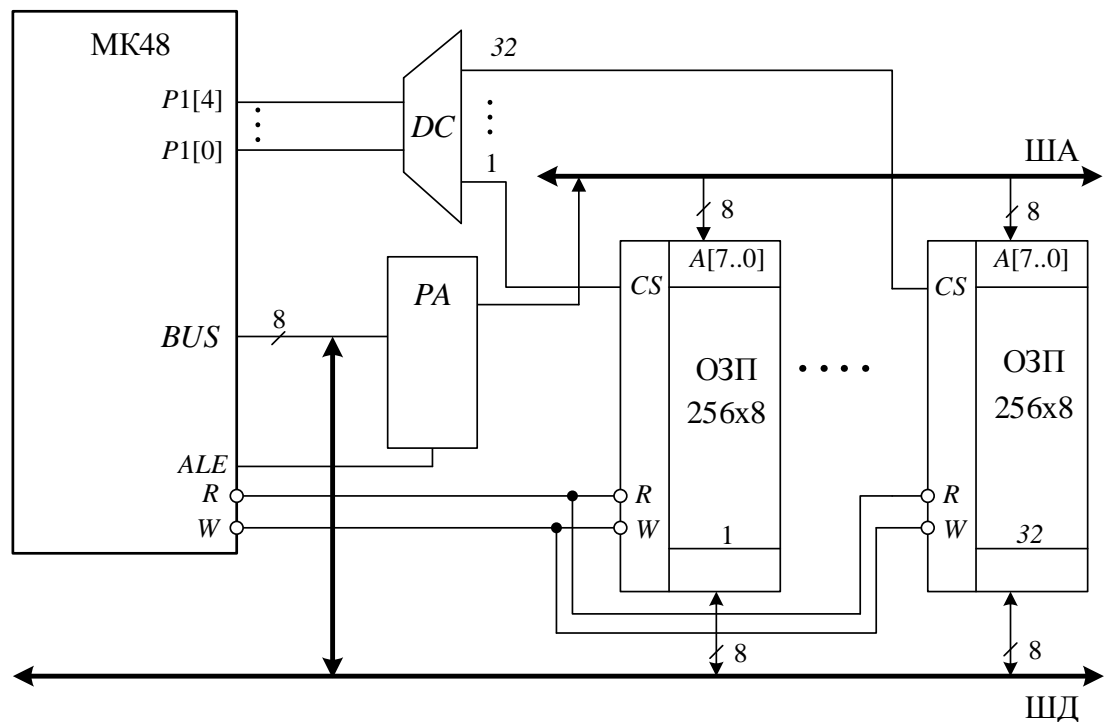


Рис. 1.5. Схема підключення зовнішньої пам'яті даних

Селектор адрес будується на вході CS сторінки зовнішньої пам'яті даних (в моєму випадку останньої). Його призначення – визначення того, з чим ми будемо працювати: власне з пам'яттю чи з підключеними ЗП або ПЗА. Селектор адрес, виходячи з карти розподілу пам'яті (рис. 1.6) зображений на рис. 1.7.

FFh	1111 1111	РД	3П8
FEh	1111 1110	РС	
...			
F1h	1111 0001	РД	3П1
F0h	1111 0000	РС	
EFh	1110 1111	РД	3П24
EEh	1110 1110	РС	
...			
D1h	1101 0001	РД	3П9
D0h	1101 0000	РС	
...			
21h	0010 0001	РД	BB51
20h	0010 0000	РЧС	

Рис. 1.6. Карта розподілу пам'яті

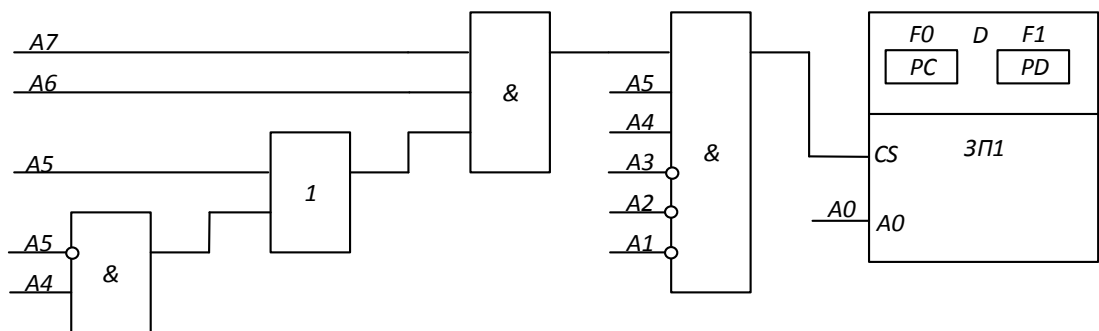


Рис. 1.7. Селектор адреси для ЗП1

1.4. Контролер пріоритетних переривань (КПП)

Під перериванням розуміють тимчасову при зупинку виконання програми і перехід на іншу підпрограму з можливістю повернення до перерваної.

Підключення контролеру пріоритетних переривань зображено на рис. 1.8.

В процесі ініціалізації системи процесор записує в регістр стану ЗП одиницю в біт дозволу переривання, якщо цей пристрій буде працювати в режимі переривання.

Коли ЗП уже готовий до обміну, встановлюється біт готовності в регістрі стану своїм контролером. При співпадінні сигналів готовності та дозволу переривання формується низьким рівнем сигнал вимоги переривання на спільній однопровідній шині.

У розрахунковій роботі використано децентралізований (розподілений) контролер ПП. Тому всі ЗП мають бути підключені до магістралі процесора.

Це обумовлено тим, що вектор переривання на шину даних в даному випадку видає власне активний ЗП.

До складу кожного ЗП включений блок контролера переривань БКП, який видає сигнал запиту IRQ на загальну лінію IRQ . У відповідь сигнал процесора $IACK$ розповсюджується послідовно через елементи БКП, створюючи так званий пріоритетний ланцюжок або "гірлянду" (daisy chain). Елементи ланцюжка в кожному БКП пропускають сигнал $IACK$ або розривають ланцюжок. Пріоритетний ланцюжок розривається на першому (по шляху розповсюдження сигналу) активному ЗП, який виставив сигнал запиту IRQ . Даний активний ЗП видає на шину даних вектор переривання, який приймається процесором.

Приклад побудови БКП показаний на рис. 1.9.

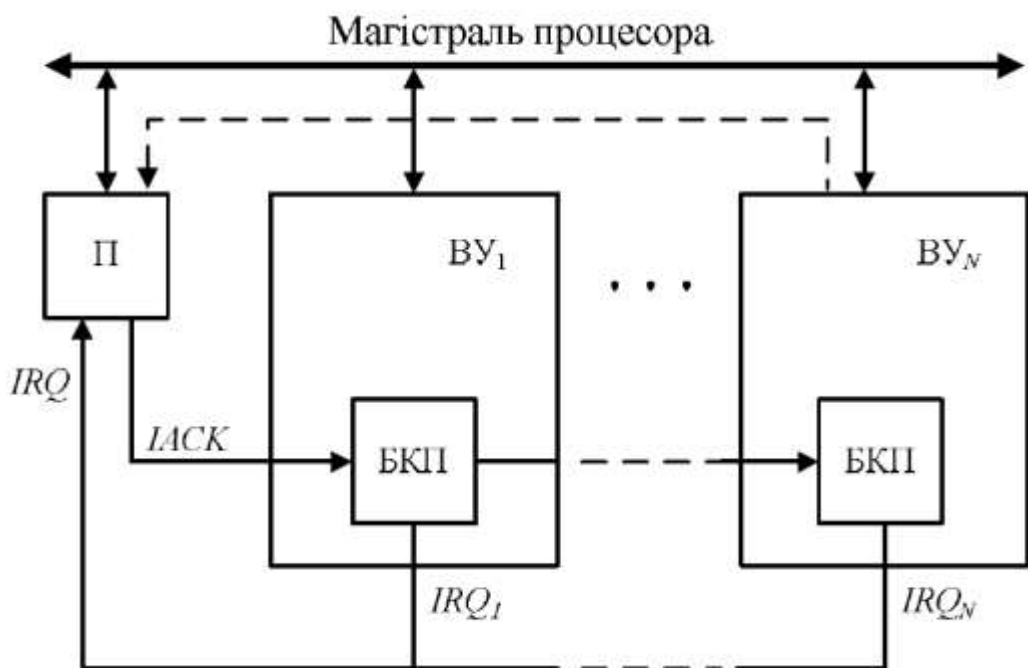


Рис. 1.8. Система з децентралізованим КПП

Блок БКП містить доступні для процесора регістр стану (РС) і регістр вектора (РВ). Під час ініціалізації режиму роботи системи процесор записує в регістр РВ вектор переривання, а в регістр РС – біт дозволу переривання (ДП). Якщо ЗП готовий до взаємодії з процесором, то в регістрі РС встановлюється біт готовності «Г». Це встановлення виконується засобами внутрішнього управління ЗП. За збігу сигналів «Г» і «ДП» формується запит IRQ , який через елемент узгодження поступає на лінію IRQ . Вхідний для кожного блоку сигнал $IACK_{in}$ передається на вихід $IACK_{out}$ або забезпечує видачу вектора через буфер вектора (БВ), що визначається значенням IRQ .

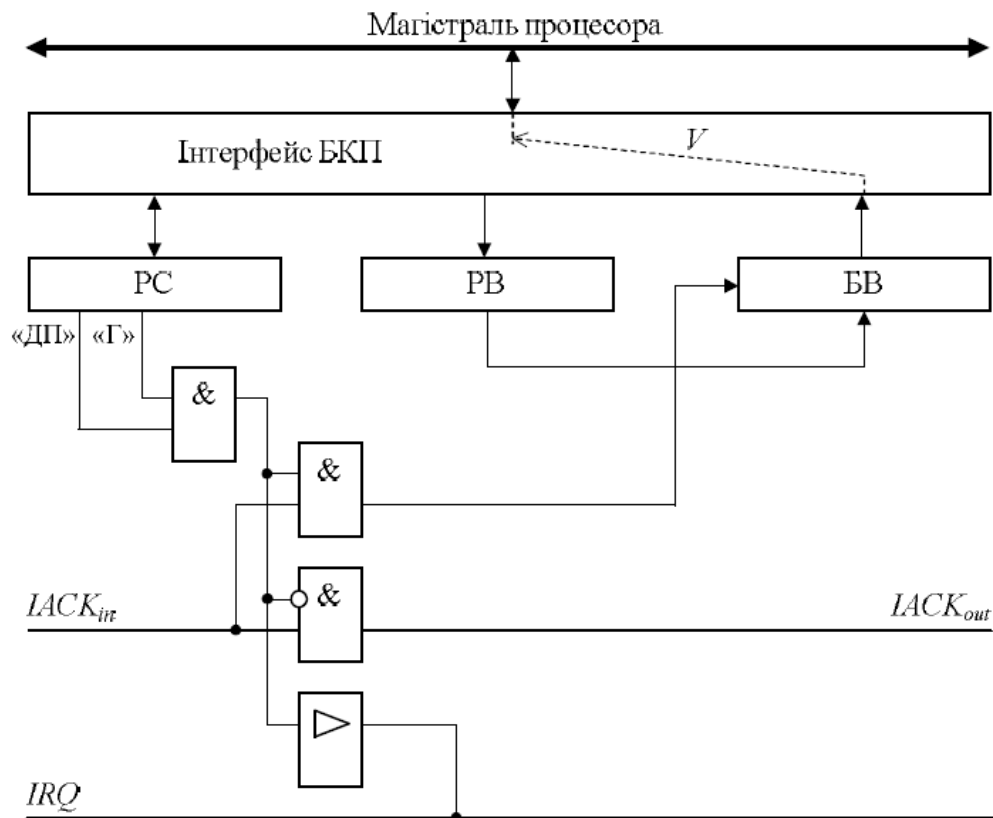


Рис. 1.9. Блок контролера переривань розподіленого КПП

1.5. Контролер прямого доступу до пам'яті (КПДП)

Режим ПДП є найбільш швидкісним способом обміну, якій реалізується за допомогою спеціальних апаратних засобів – контролерів ПДП без використання програмного забезпечення. Для здійснення режиму ПДП контролер має виконати ряд послідовних операцій (рис. 1.10):

- 1) прийняти запит DREQ на ПДП від ЗП;
- 2) сформувати запит HRQ на захоплення шин для ЦП;
- 3) прийняти сигнал HLDA, що підтверджує цей факт, після того, як ЦП увійде в стан захоплення (ШД, ША, ШУ в z-стані);
- 4) сформувати сигнал DACK, що повідомляє ЗП про початок виконання циклів ПДП;
- 5) сформувати на ША адрес комірки пам'яті, призначеної для обміну;
- 6) виробити сигнали MR, IOW и MW, IOR, що забезпечують керування обміном;
- 7) по закінченні ПДП або повторити цикл ПДП, змінивши адресу, або перервати ПДП, знявши запити на ПДП.

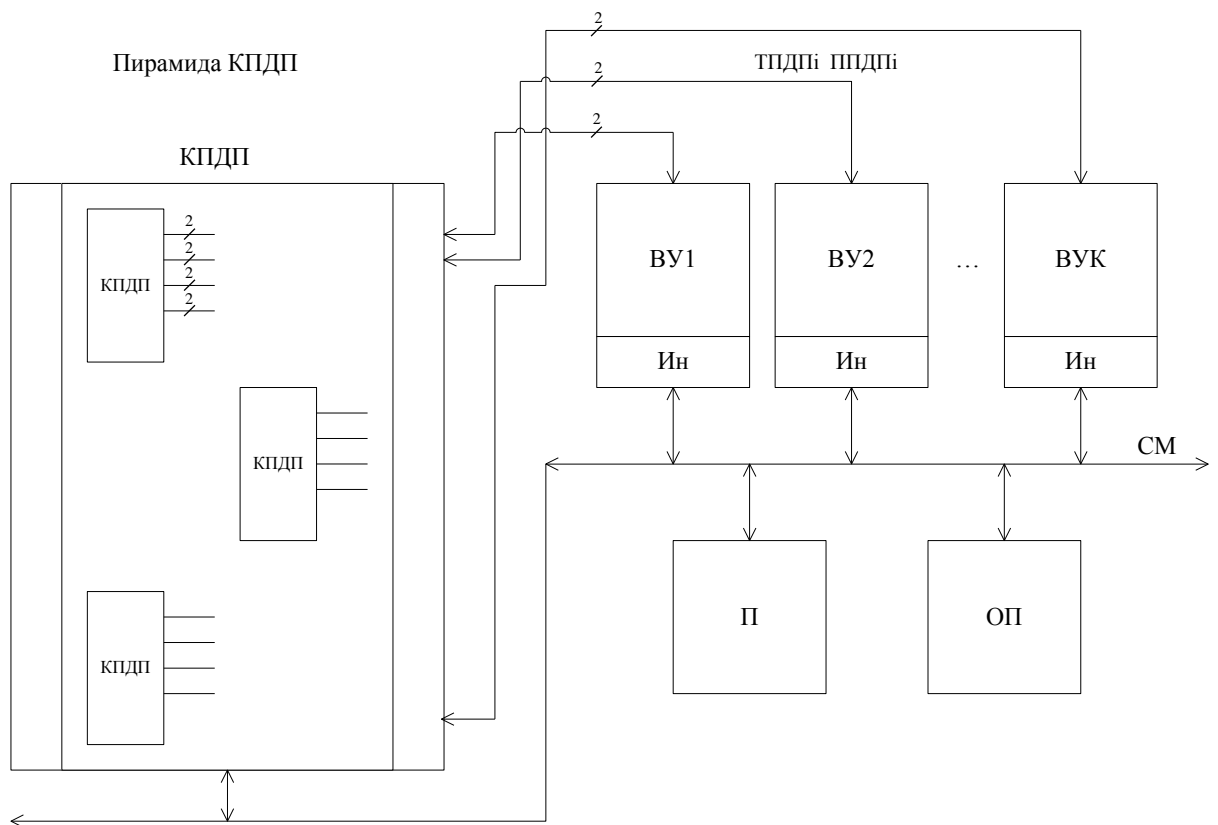


Рис 1.10. Структурна схема децентралізованого КПДП

1.6. Додаткові порти і ПЗА

Для збільшення кількості ліній зв'язку МК48 з об'єктом управління можна підключати додаткові чотирирозрядні порти P4, P5, P6, P7. Найбільш просто це здійснюється за використання спеціальної ІС КР580 ВР43, спосіб підключення якої до МК48 показаний на рис. 1.11. В цьому випадку забезпечується виконання всіх чотирьох команд роботи з додатковими портами – MOVD A, Pp; MOVD Pp, A; ANLD Pp, A та ORLD Pp, A, причому кожний вихід порту може бути налаштований як на введення, так і на виведення інформації. У нашому випадку підімкнено порти P4, P6, P7.

Команди передачі інформації між МК48 та додатковими портами виконуються за два цикли. В першому циклі на виходах P2[3..0] встановлюється управляюче слово, в другому циклі – через зазначені виходи здійснюється обмін інформацією між МК48 та одним з додаткових портів. Формат управляючого слова показаний на рис. 1.12.

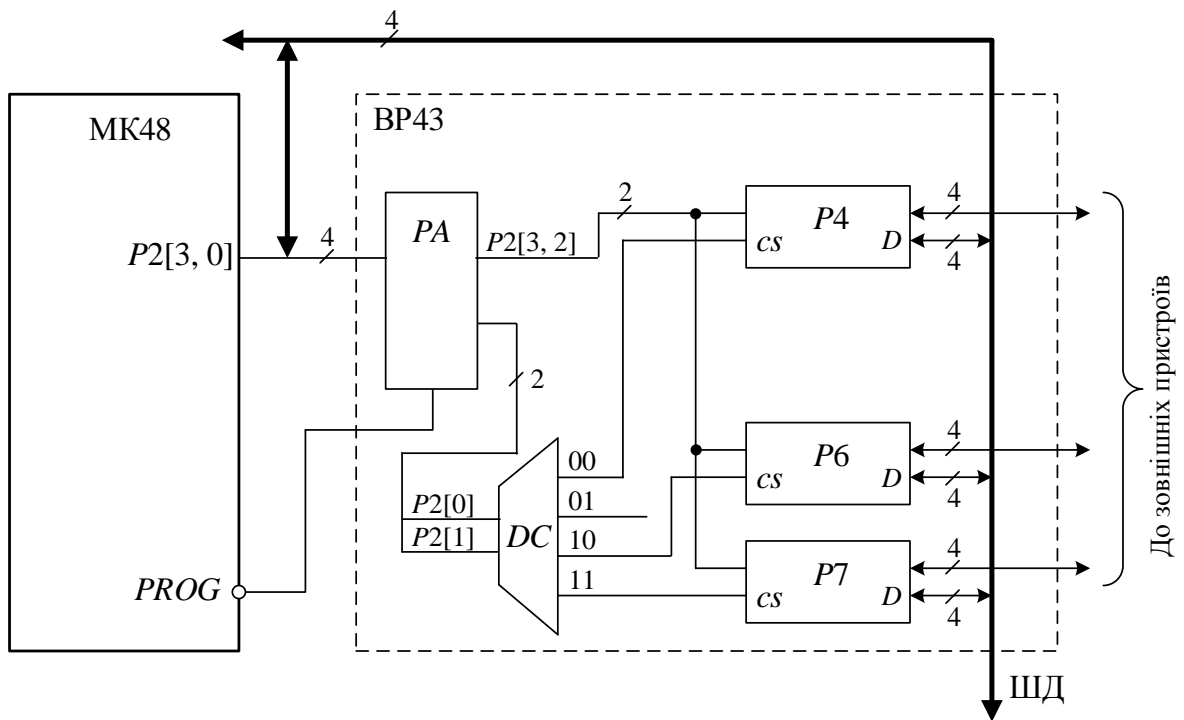


Рис. 1.11. Схема з'єднання виходів МК та IC KP580BP43

	3	2	1	0	
MOV D A, Pp	0	0	0	0	P4
MOV D Pp, A	0	1	0	1	-
ANLD Pp, A	1	0	1	0	P6
ORLD Pp, A	1	1	1	1	P7

Рис. 1.12. Структура управляючого слова

Структурна схема підключення програмованого зв'язуючого адаптера KP580BB51 до мікроконтролера МК48 приведена на рис. 1.13.

Адреси портів ПЗА входять у загальний адресний простір зовнішньої пам'яті даних. Для уникнення перетину адрес загального адресного простору ЗПД та ПЗА застосовано селектор адреси СА. Доступ до портів під час запису та читання здійснюється за застосування команд $\text{MOVX A}, @Rr$; $\text{MOVX } @Rr, A$ (де, $r = 1, 0$).

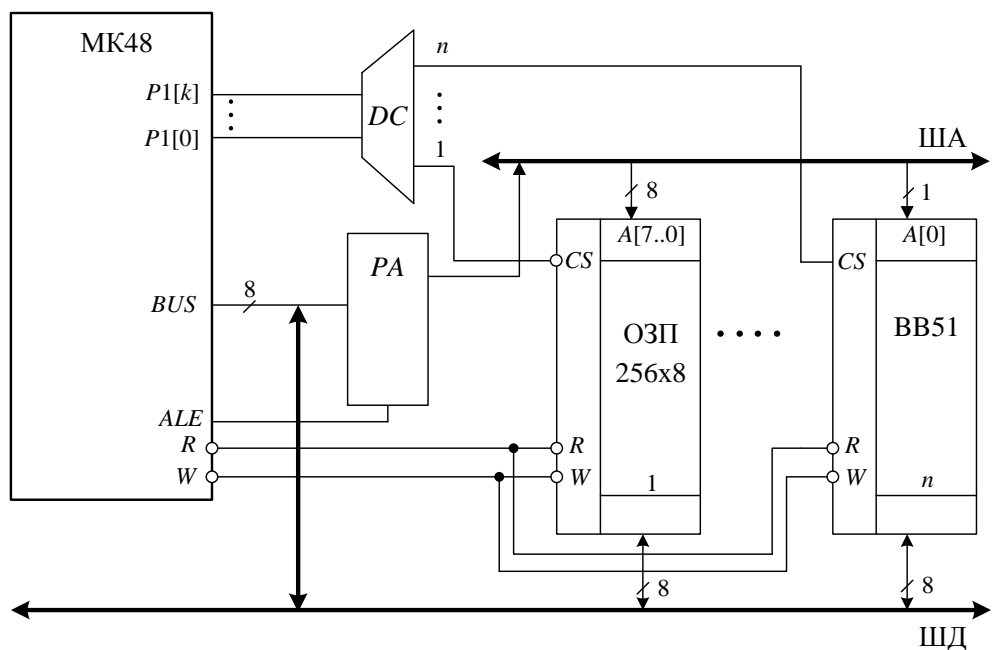


Рис. 1.13. Структурна схема підключення програмованого зв'язуючого адаптера KP580BB51 до мікроконтролера МК48

2. Програмна частина

2.1. Приклад переходу на підпрограму

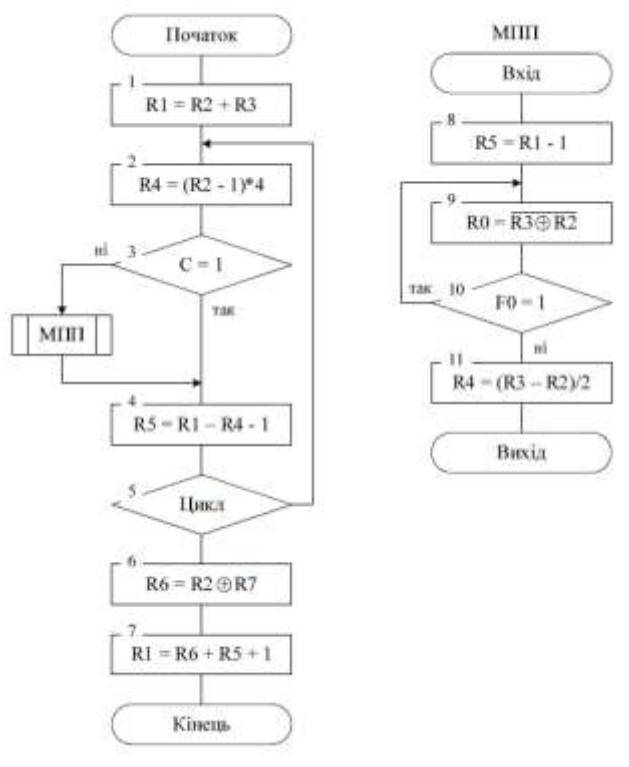


Рисунок. 2.1 Алгоритм обчислення функції з передачею управління


```

; Block 7
; R1=R6+R5+1
    Clr    C
    Sel    Rb0
    Mov    A, R5
    Addc   A, R6
    Mov    R1, A

    Sel    Rb1
    Mov    A, R5
    Addc   A, R6
    Mov    R1, A

    Clr    C
    Sel    Rb0
    Mov    A, R1
    Addc   A, #1
    Mov    R1, A

    Sel    Rb1
    Mov    A, R1
    Addc   A, #0
    Mov    R1, A

    Jmp    Ending

; Block 8
; R5=R1-1
MPP:
    Clr    C
    Sel    Rb0
    Mov    A, R1
    Addc   A, #FFh
    Mov    R5, A

    Sel    Rb1
    Mov    A, R1
    Addc   A, #FFh
    Mov    R5, A

; Block 9
; R0=NOT(R3 XOR R2)
MPP_cont:
    Sel    Rb0
    Mov    A, R3
    Xrl    A, R2
    Mov    R0, A

    Sel    Rb1
    Mov    A, R3
    Xrl    A, R2
    Mov    R0, A

    Sel    Rb0
    Mov    A, R0
    Cpl    A
    Mov    R0, A

; Block 10
; IF (F0=1)
    Clr    F0
    Cpl    F0
    Jc     MPP_cont

; Block 11
; R4=(R3-R2)/2
    Clr    C
    Sel    Rb0
    Mov    A, R2
    Cpl    A
    Addc   A, #1h
    Mov    R4, A

    Sel    Rb1
    Mov    A, R2
    Cpl    A
    Addc   A, #0
    Mov    R4, A

    Clr    C
    Sel    Rb0
    Mov    A, R4
    Addc   A, R3
    Mov    R4, A

    Sel    Rb1
    Mov    A, R4
    Addc   A, R3
    Mov    R4, A

    Clr    C
    Sel    Rb1
    Mov    A, R4
    Rrc    A
    Mov    R4, A

    Sel    Rb0
    Mov    A, R4
    Rrc    A
    Mov    R4, A

    Nop
    Ret

Ending:
    Nop
    End

```

2.2. Приклад формування часової затримки

```
;Задержки Y1=720мкс; Y2=25мкс; Y5=800мкс.
;Формируем задержку 25 мкс = 5мкс * 5 для Y1, Y2, Y5.
    ANL P2, #11000000b; Y3 OFF
    ORL P2, #00100110b; Y1, Y2, Y5 ON
    MOV R7, #5
label1: DJNZ R7, label1
    ANL P2, #11100010b; Y2 OFF

;Формируем задержку 55мкс = 5мкс * 11 для Y1, Y5.
    MOV R7, #11
label2: DJNZ R7, label2

;Формируем задержку 640мкс = 80мкс * 8 для Y1, Y5.
    MOV A, #11111000b; (-8)ДК
    MOV T, A
    STRT T
label3: JTF label4
    JMP label3

;Формируем задержку 80мкс = 80мкс * 1 для Y5.
label4: ANL P2, #11100000b; Y1 OFF
    MOV A, #11111111b; (-1)ДК
    MOV T, A
    STRT T
label5: JTF label6
    JMP label5
label6: ANL P2, #11000000b; Y5 OFF
    NOP
    END
```

2.3. Приклад виконання операції множення 1-им способом

```
;Ввод
JBO label2
JMP label3

SEL RB0
IN A, P1
MOV R3, A
IN A, P1
MOV R4, A
IN A, P1
MOV R5, A
MOV R6, #24

SEL RB1
IN A, P1
MOV R0, A
IN A, P1
MOV R1, A
IN A, P1
MOV R2, A

; Блок 2
; Суммирование Z=Z+X
Label2: SEL RB0
        CLR C
        MOV A, R2
        ADDC A, R5
        MOV R2, A
        MOV A, R1
        ADDC A, R4
        MOV R1, A
        MOV A, R0
        ADDC A, R3
        MOV R0, A

; Блок 3
; Сдвиг Z, Dec счетчика
Label3: SEL RB0
        CLR C
        MOV A, R0
        RRC A
        MOV R0, A

; Блок 1
; Проверка младшего бита Y
Label1: SEL RB1
        MOV A, R2
```

```

MOV    A, R1
RRC    A
MOV    R1, A
MOV    A, R2
RRC    A
MOV    R2, A

SEL    RB1
MOV    A, R0
RRC    A
MOV    R0, A
MOV    A, R1
RRC    A

```

```

MOV    R1, A
MOV    A, R2
RRC    A
MOV    R2, A

SEL    RB0
DJNZ   R6, label1

Nop
End

```

2.4 Обчислення функції за варіантом

$X=(X1-X2)+(X5+X6)*5+(X9+X10)/16$, операнди 16-розрядні

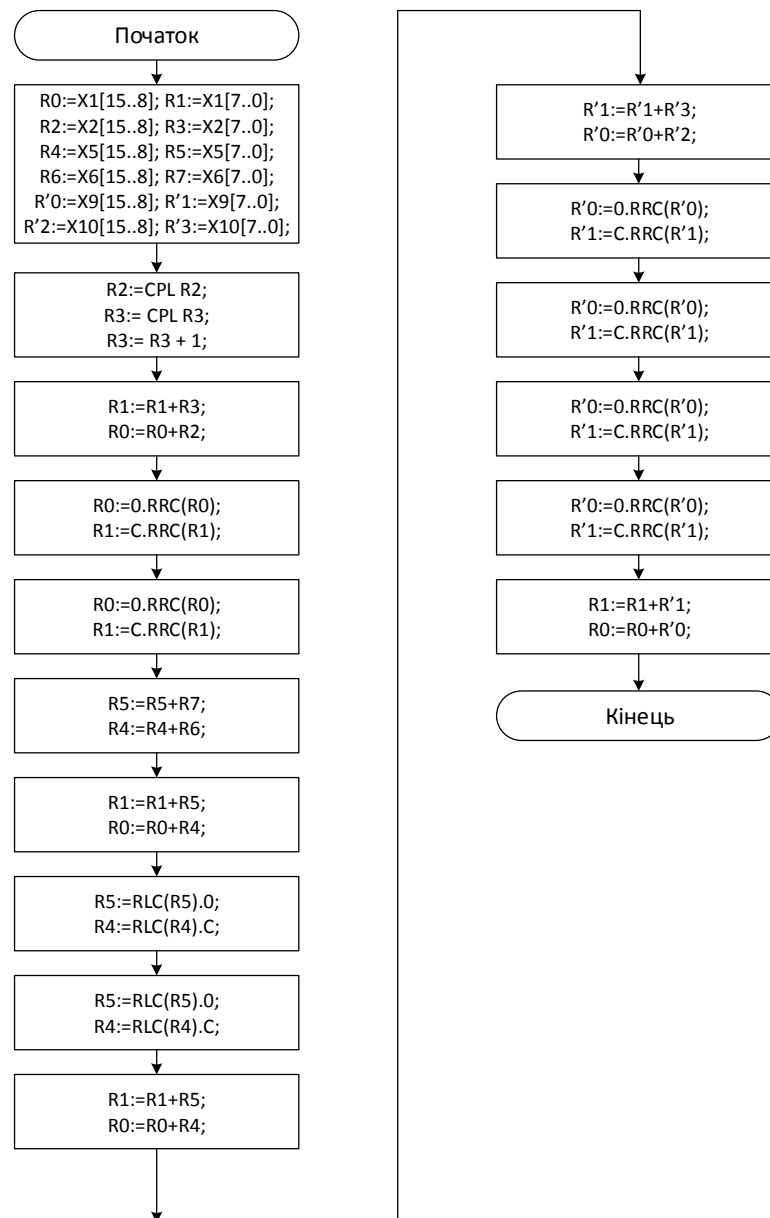


Рис. 2.2 Блок схема виконання функції

Код програми:

```

SEL RB0
R0 = #5h; X1[15..8]
R1 = #5h; X1[7..0]
R2 = #5h; X2[15..8]
R3 = #5h; X2[7..0]
R4 = #5h; X5[15..8]
R5 = #5h; X5[7..0]
R6 = #5h; X6[15..8]
R7 = #5h; X6[7..0]
SEL RB1
R0 = #5h; X9[15..8]
R1 = #5h; X9[7..0]
R2 = #5h; X10[15..8]
R3 = #5h; X10[7..0]

;X2=-X2
SEL RB0
MOV A, R3
CPL A
MOV R3, A
MOV A, R2
CPL A
MOV R2, A
CLR C
MOV A, R3
ADDC A, #1h
MOV R3, A
MOV A, R2
ADDC A, #0h
MOV R2, A

;X1=X1-X2
CLR C
MOV A, R3
ADDC A, R1
MOV R1, A
MOV A, R2
ADDC A, R0
MOV R0, A

;X1=(X1-X2)/2
CLR C
MOV A, R0
RRC A
MOV R0, A
MOV A, R1
RRC A
MOV R1, A

;X1=(X1-X2)/4
CLR C
MOV A, R0
RRC A
MOV R0, A
MOV A, R1
RRC A
MOV R1, A

;X5=X5+X6
CLR C
MOV A, R5
ADDC A, R7
MOV R5, A
MOV A, R4
ADDC A, R6
MOV R4, A

;X1=(X1-X2)/4+(X5+X6)
CLR C
MOV A, R4
ADDC A, R1
MOV R1, A
MOV A, R5
ADDC A, R0
MOV R0, A

;X5=(X5+X6)*2
CLR C
MOV A, R5
RLC A
MOV R5, A
MOV A, R4
RLC A
MOV R4, A

;X5=(X5+X6)*4
CLR C
MOV A, R5
RLC A
MOV R5, A
MOV A, R4
RLC A
MOV R4, A

;X1=(X1-X2)/4+(X5+X6)*5
CLR C
MOV A, R4
ADDC A, R1
MOV R1, A
MOV A, R5
ADDC A, R0
MOV R0, A

;X9=X9+X10
SEL RB1
CLR C
MOV A, R3
ADDC A, R1
MOV R1, A
MOV A, R2
ADDC A, R0
MOV R0, A

;X9=(X9+X10)/2
CLR C
MOV A, R0

RRC A
MOV R0, A
MOV A, R1
RRC A
MOV R1, A

;X9=(X9+X10)/4
CLR C
MOV A, R0
RRC A
MOV R0, A
MOV A, R1
RRC A
MOV R1, A

;X9=(X9+X10)/8
CLR C
MOV A, R0
RRC A
MOV R0, A
MOV A, R1
RRC A
MOV R1, A

;X9=(X9+X10)/16
CLR C
MOV A, R0
RRC A
MOV R0, A
MOV A, R1
RRC A
MOV R1, A

;X9=(X1-X2)/4+(X5+X6)*5
+ (X9+X10)/16
CLR C
SEL RB0
MOV A, R1
SEL RB1
ADDC A, R1
SEL RB0
MOV R1, A
MOV A, R0
SEL RB1
ADDC A, R0
SEL RB0
MOV R0, A

NOP
END

```

3. Структурна схема системи.

Структурна схема МПС приведена на кресленні НТУУ КПІ 15 3110 004. Е1. Її ядром є мікроконтролер КР1816ВЕ48.

До складу МПС входять такі основні функціональні частини:

1. Мікроконтролер МК-48.
2. Зовнішня ПД 8К, зовнішня ПП 4К.
3. Децентралізований КПП та КПДП..
4. Зовнішні пристрої - 24 одиниці.
5. Додаткові порти Р4, Р6, Р7.
6. Зв'язувальний адаптер ВВ51 для підключення ЗП

4. Висновки

Розроблена МПС на основі МК-48 з підключенням зовнішньої пам'яті даних, зовнішньої пам'яті програм, зв'язувального адаптера та зовнішніх пристроїв.

5. Список літератури.

1. Бояринов А.Е., Дьяков И.А. – Архитектура микроконтроллеров MCS-51 – Тамбов: “Издательство ТГТУ”, 2005.
2. Конспект лекцій по курсу “Архітектура ЕОМ”.
3. Жабин В.И., Ткаченко В.В., Макаров В.В., Зайцев А.А. – Архитектура однокристальных ЭВМ. – Киев, “Век”, 1997.