



## Вступ

Мікропроцесорами (МП) є автономні функціонально закінчені пристрої, що складаються з однієї або декількох програмно-керованих інтегральних мікросхем високого ступеня інтеграції, які включають усі засоби, необхідні для обробки інформації та управління даними, і розроблені для спільної роботи з пристроями пам'яті і введення-виведення інформації.

Структура мікропроцесора повинна задовольняти трьом основним вимогам: бути функціонально гнучкою, забезпечити достатньо високу швидкість і дозволити дешеву технологічну реалізацію. Висока функціональна гнучкість мікропроцесора, необхідна для створення ефективного програмного забезпечення, досягається мікропрограмним управлінням, за рахунок побайтової обробки і адресації даних, розвиненої системи переривань і великої кількості внутрішніх регістрів.

					ІАЛЦ 462619.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		2

## Розділ 1. Огляд існуючих рішень

### Intel

#### • Intel 8051

Intel 8051 – це однокристальний мікроконтролер зарвардської архітектури, який був вперше

вироблений Intel в 1980 році, для використання у вбудованих системах.

– Склад: процесорне ядро (CPU), ОЗП, ПЗП, послідовний порт, паралельний порт, логіка управління переривань, таймер, тощо

– Шина даних: 8 біт (обробка 8 біт даних за одну операцію).

– Шина адреси: 16 біт (доступ до 216 адресами пам'яті – 64 кБ адресний простір в ОЗП і ПЗП).

– Вбудоване ОЗП: 128 байт (Пам'ять даних)

– Вбудоване ПЗП: 4 КБ (Пам'ять програм)

– Чотири порти введення / виводу: один двонаправлений і три квазідвонаправлених.

– Послідовний інтерфейс UART (Універсальний асинхронний приймач).

– Два 16-бітових таймера

– Два рівня пріоритету переривання

– Порядку 60 тисяч транзисторів на кристалі площею 5,85 мм<sup>2</sup>

– Енергозберігаючий режим

#### • Intel 8058

Intel 8058 – це восьмибітний однокристальний мікроконтролер компанії Intel, що з'явився на ринку в 1980. Він належить до групи MCS-51, яка практично ідентична процесору 8051. Різниця з іншими версіями в наступних параметрах:

– Розмір пам'яті даних RAM: 256 байт.

– Розмір пам'яті програм ROM: до 32 кБ.

– Перемикання частоти: до 33 МГц.

					ІАЛЦ 462619.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		3

- Включає в себе 3 лічильника / таймера(він складається з чотирьох 8-розрядних портів).

### Atmel

- tinyAVR

- Пам'ять програм: 0,5-16 кБ
- Байти оперативної пам'яті: 32-102
- Байти енергонезалежної пам'яті EEPROM: 0-512 байтів
- 6-32-виводів на корпусі
- Обмежений набір внутрішніх периферійних пристроїв
- Напруга живлення: 1,8-5,5 В

- megaAVR

- Пам'ять програм: 4-256 кБ
- Оперативна пам'ять: 0,25-8 кБ
- Енергозалежна пам'ять даних EEPROM 0,25-4 кБ енергонезалежної пам'яті Даних EEPROM

- 28-100-вивідний корпус
- Розширений набір команд (команди множення, розширені методи адресації в командах LPM, SPM)

- Розширений набір внутрішніх периферійних пристроїв
- Напруга живлення: 1,8-5,5 В

- XMEGA

- Флеш-пам'ять: 384 Кб
- SRAM до 32 Кб
- EEPROM до 4 КБ
- Чотирьохканальний DMA-контролер
- Інноваційна система обробки подій.
- 44-64-100-вивідний корпус
- Напруга живлення: 1.6 - 3.6В.

### Philips

- LPC2000

- 16/32-разр. мікроконтролер ARM7TDMI-S в корпусі LQFP144

					ІАЛЦ 462619.004 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дат		

- Вбудований статичний ОЗп: 16 кбайт
- Послідовний завантажувач, який використовує USART (дозволяє виконати внутрішньосистемне програмування)
- Вбудовані інтерфейси: ICE-RT, Trase (дозволяють відслідковувати виконання програми в реальному масштабі часу через наявну в мікроконтролері програму RealMonitor <sup>™</sup>)
- Два пов'язаних CAN-інтерфейсу з модернізованими прийомними фільтрами.
- Додаткові послідовні інтерфейси: два USART (16C550), високошвидкісний I2C (400 кбіт / с) і два SPI.8-канальний 10-бітний АЦП з часом перетворення 2,44 мкс.
- Два 32-разр. таймера (з 4 каналами захоплення і 4 каналами порівняння), модуль ШІМ (6 виходів), годинник реального часу і сторожовий таймер.
- Векторний контролер переривань з конфігуруючими пріоритетами і адресами векторів.
- Інтерфейс що конфігурується зовнішньою пам'ятю з 4 банками, кожен з яких має розмір 16 Мбіт з розрядністю даних 8/16/32.
- До 76 універсальних ліній введення-виведення (підтримують 5В рівні). З них 9 виводів можуть бути входами переривання по фронту або за рівнем.
- Максимальна тактова частота: 60 МГц, що виробляється вбудованим генератором ФАПЧ з часом стабілізації 100 мкс.

### SIEMENS [Siemens Semiconductor Group]

- × C166 <sup>™</sup> C161, C165, C166, C167, C16x from Infineon
- 80 нс Мінімальна тривалість циклу команди, при цьому більшість команд, які виконуються в 1 цикл
- 400 нс множення (16-біт \* 16-біт), 800 нс поділом (32-біт / 16-біт)
- 16 Мб лінійного адресного простору для коду та даних (архітектура фон Неймана)
- 2 кБайти внутрішнє ОЗП для змінних, системного стека і коду
- 2 кБайти високошвидкісної пам'яті XRAM для змінних, стека користувача і коду (не на всіх моделях)

					ІАЛЦ 462619.004 ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дат		

- 128 кбайт або 32 кбайт програмованої флеш-пам'яті (для мікроконтролерів без ПЗУ)
- Мультиплексні або демультіплексні конфігураційні шини
- Можливість сегментації і вибору мікросхеми генерації сигналу
- 8-бітові або 16-розрядну шину даних

### Motorola

- Сімейство HC08 – 8- розрядний АЛП. – Програмна сумісність “знизу вгору” на рівні початкового тексту з процесорним ядром сімейства HC05.
- Закрита архітектура. МК містять на кристалі резидентний ПЗП програм; режим адресації зовнішньої пам'яті у більшості моделей відсутня. – Перехід до FLASH-технології для ПЗП програм користувача.
- Інтеграція на кристалі МК трьох типів пам'яті: пам'яті програм (maskROM, EPROM, FLASH), оперативної пам'яті даних (статичний ОЗП) і незалежній пам'яті даних (EEPROM), яка програмується і стирається в робочому режимі МК
- Сімейства HC05
- Напруга живлення 3 або 5 В;
- Частота внутрішньої шини до 8 МГц;
- Об'єднана пам'ять програм і даних (архітектура фон Неймана);
- 16 режимів адресації; – 16-розрядні індексний регістр і покажчик стека;
- Пересилання з пам'яті в пам'ять; – апаратне множення 8.8 розрядів;
- Апаратне делення 16/8 розрядів;
- Апаратна підтримка двійково-десятькової арифметики.

					ІАЛЦ 462619.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		6

## Розділ 2. Розробка мікропроцесорної системи

За варіантом потрібно розробити МПС на базі контролера 1816BE48 з ШАД, підключити: децентралізовані КПП та КПДП; додаткові порти P4, P6; P7, BB51, 86 зовнішні пристрої.

### 2.1 Опис мікроконтролера 1816BE48

Мікроконтролер МК48 конструктивно виконаний в корпусі БІС з 40 зовнішніми виводами. Всі виводи електрично сумісні з елементами ТТЛ, входи є одиничним навантаженням, а виходи можуть бути навантажені одним ТТЛ-навантаженням. Основу структури МК утворює внутрішня двонаправлена 8-бітна шина, яка пов'язує між собою всі пристрої БІС: арифметично-логічний пристрій (АЛП), пристрій управління, пам'ять і порти введення / виводу інформації.

Структури ОЕОМ серії 1816 та їхніх команд такі, що в разі потреби функціонально-логічні можливості можуть бути розширені. З використанням зовнішніх додаткових БІС, постійної й оперативної пам'яті, адресний простір можна значно розширити, а шляхом підключення різних інтерфейсних БІС, число ліній зв'язку з об'єктом управління можна збільшити майже без обмежень.

ОЕОМ серії 1816 розсіюють потужність близько 1,5 Вт і працюють в діапазоні температур від 0 до 700С. ОЕОМ МК 48 може працювати в діапазоні частот синхронізації від 1 до 6 МГц, а мінімальний час виконання команди складає 2,5 мкс. Найпростіша в серії 1816 однокристальна ОЕОМ МК48 має на кристалі наступні апаратурні засоби:

- × процесор розрядністю 1 байт;
  - × перезаписуваний програмований ПЗП програм ємністю 1 Кбайт;
  - × ОЗП даних ємністю 64 байти; × програмований 8-бітний таймер / лічильник;
  - × програмовані схеми вводу / виводу;
  - × блок векторного переривання від двох джерел;
  - × Генератор тактових сигналів і систему синхронізації і управління.
- Всі ці елементи пов'язані між собою через системну магистраль.

Структурна схема мікроконтролера 1816BE48 представлена на Рис. 2.1.

Пам'ять програм і пам'ять даних в МК48 розділені. Максимальний адресний простір пам'яті програм становить 4Кб. Пам'ять програм поділяють на внутрішню і зовнішню.

РС – лічильник команд – має довжину 12 біт. Після вибірки чергового байту вміст РС збільшується на 1. Перенесення поширюється при цьому тільки від 0 до 10-ого розряду.

					ІАЛЦ 462619.004 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дат		

Пам'ять даних також поділяється на зовнішню і внутрішню. Внутрішня пам'ять даних (64 байти) містить 2 банки регістрів по 8 регістрів в кожному, кожен регістр містить 1 байт.

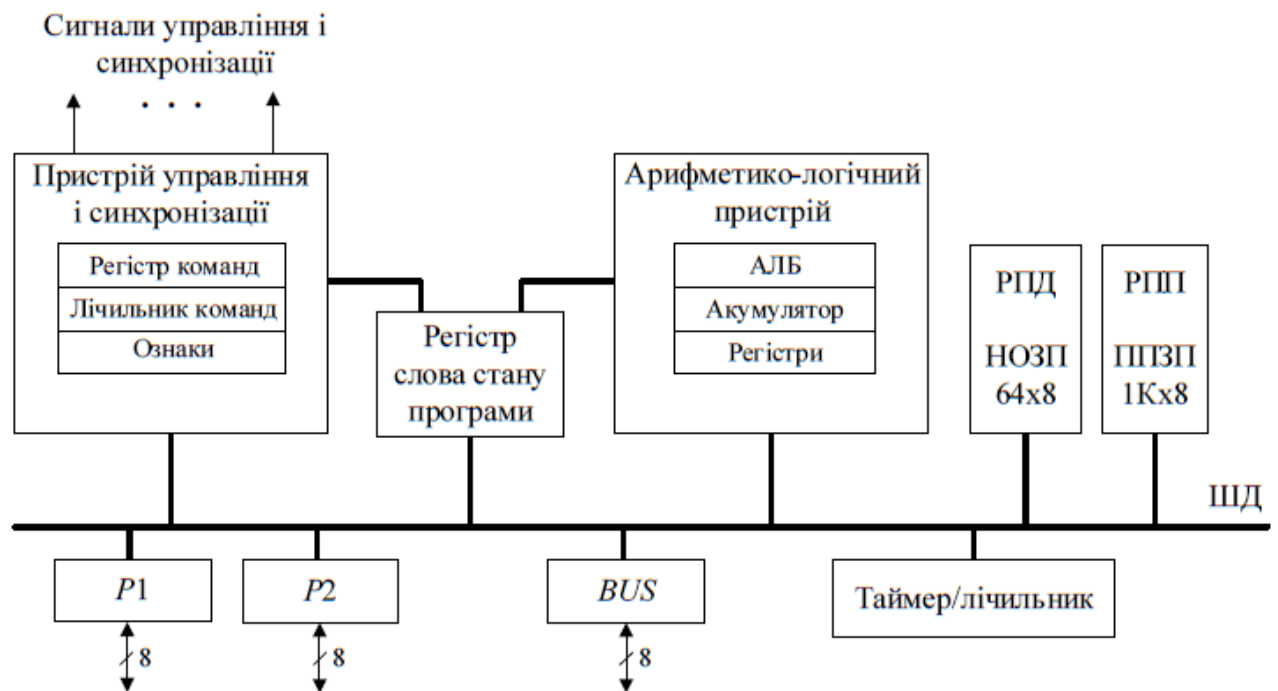


Рис. 2.1. Структурна схема мікроконтролеру 1816BE48

Арифметико-логічний пристрій працює з арифметико-логічним блоком, акумулятором і регістром стану.

А – акумулятор – 8-міразрядний регістр, який використовується в якості джерела і наступника результату, а також у непрямії адресації.

До вузла синхронізації можна підключити зовнішній генератор.

У мікроконтролері реалізована система переривання від 2 джерел: внутрішнього таймера / лічильника і зовнішнього. Внутрішній 8-мірівневий стек забезпечує автоматичне збереження і відновлення основних параметрів обчислювального процесу при запитах переривання і при поверненні після обслуговування переривання.

Для зв'язку з об'єктом управління у мікроконтролера є 3 восьми розрядних порту (24 лінії зв'язку). Порти P1 і P2 називають «квазідвупаправлені». Їх особливість полягає в тому, що при введенні над входними даними



і поточним станом порту (даними, які виводилися з порту останніми) виконується подібна логічна операція І. Вихідні дані в порту запам'ятовуються. При скиданні системи, кожному біту порту присвоюється значення 1. Порт BUS має звичайні двонаправлені виводи 3 станами.

## 2.2. Зовнішня пам'ять програм

Для реалізації більш складних програмно-апаратних способів доступу до зовнішньої пам'яті, її ємність може бути збільшена до необхідного об'єму за рахунок сторінкової організації, при цьому зовнішня пам'ять даних поділяється на сторінки по 256 байт в кожній, а зовнішня пам'ять програм — на сторінки по 4К байт. Для переключення між сторінками можна використовувати, вільні лінії портів  $P1$  та  $P2$ .

### Режим роботи з зовнішньою пам'яттю програм

Режим роботи МК48 з зовнішньою пам'яттю програм можливий за застосування додаткових мікросхем ПЗП. Якщо не використовувати сторінкову адресацію, об'єм пам'яті програм можна розширити до 4К байт. За встановлення сигналу  $EMA = 1$  доступні всі 4К байта зовнішньої пам'яті. Якщо сигнал  $EMA = 0$ , то адресація комірок зовнішньої пам'яті розпочинаються з адреси 1024. При цьому область пам'яті з адресами від 0 до 1023 належать резидентній пам'яті програм.

### Цикл отримання інструкції

Для отримання інструкції, адресація яких починається з 1024 або більше, виконуються наступна послідовність (рис 2.2):

1. Вміст 12-бітної програми лічильника буде виводитися на BUS і нижню половину порту  $P2$ .
2. Строб адреси ЗП ( $ALE$ ) вказує час при якому адреса є діющою. Задній край  $ALE$  використовується для фіксації зовнішньої адреси.
3.  $\overline{PSEN}$  вказує, що триває отримання інструкції і служить для утримування активності ЗПП.
4. Шина переходить в режим введення і процесор приймає її 8-бітний зміст в якості командного слова.

					ІАЛЦ 462619.004 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дат		

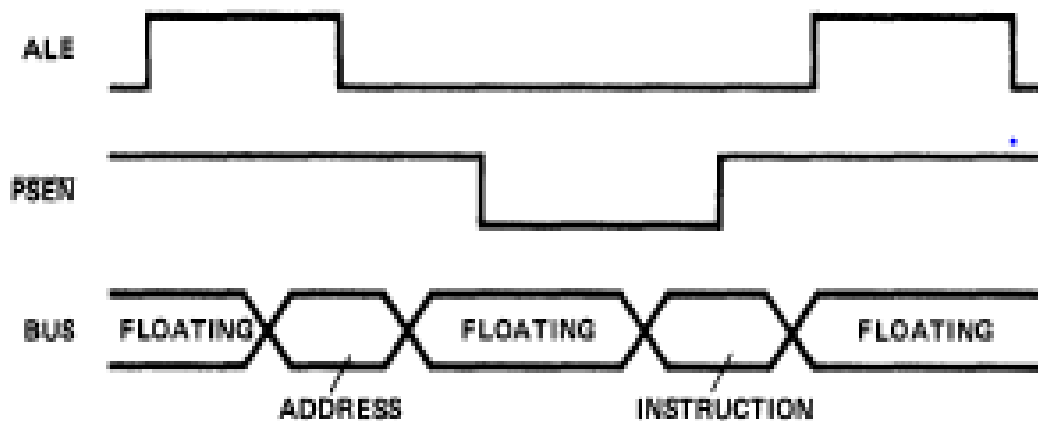


Рис 2.2. Часова діаграма тримання інструкції з ЗПП.

### Підключення ЗПП

На рис 2.3. показано підключення трьох пін-сумісних 2308 ПЗП для загального обсягу 4К слів пам'яті програм.

BUS підключається безпосередньо до порту виводу інформації на ЗП. Молодші 8 біт адреси ЗП підключаються використовуючи ALE, як строб. Молодші 4 біти порту P2 забезпечують підключення старших чотирьох бітів адреси. Два старших адресних біти підключаються безпосередньо до адресного входу пам'яті, в той час як два найбільш значущих біти декодуються і забезпечують вибір з трьох мікросхем необхідної.

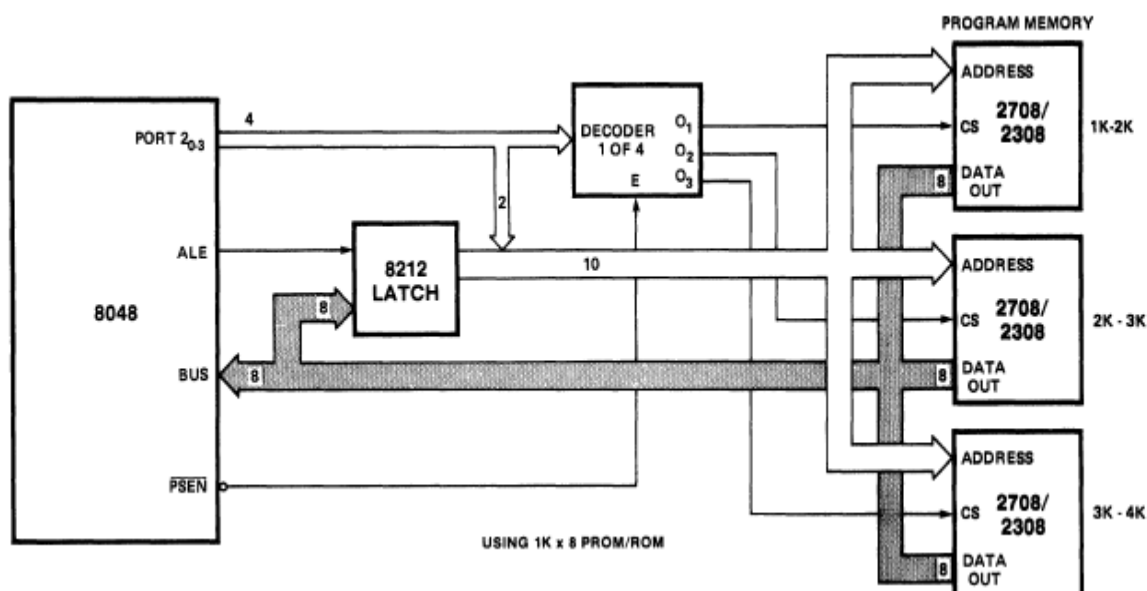


Рис 2.3. Підключення 3-х зовнішніх запам'ятовувачих пристроїв.

### 2.3. Зовнішня пам'ять даних

#### Режим роботи з зовнішньою пам'яттю даних

В режимі роботи МК48 з зовнішньою пам'яттю даних використовуються додаткові мікросхеми ОЗП об'ємом 256 байт. Якщо адресний простір має об'єм більш ніж 256 байт, то необхідна сторінкова організація зовнішньої пам'яті даних. При роботі з адресами в межах однієї сторінки застосовуються команди  $\text{MOVX } A, @R_r$ ,  $\text{MOVX } @R_r, A$  (де  $r = 1, 0$ ). Обмін інформацією здійснюється між акумулятором  $A$  і коміркою ОЗП, яка непрямо адресується через регістр  $R0$  або  $R1$ . Переключення між сторінками потребує використання додаткових команд вибору сторінки пам'яті даних. Інтерфейс зовнішньої пам'яті даних зображено на рис. 2.4.

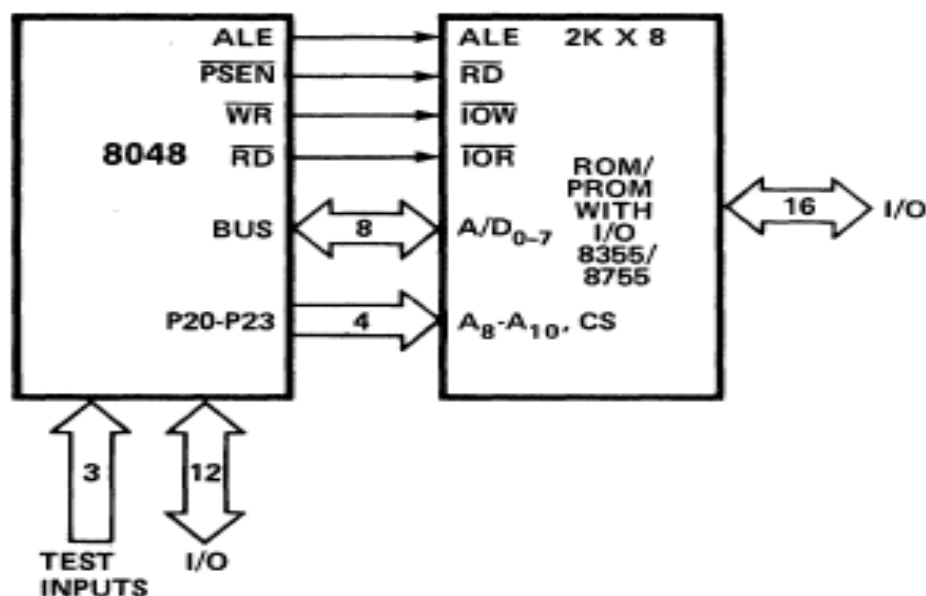


Рис.2.4. Інтерфейс зовнішньої пам'яті даних

#### Цикл читання/запису

Всі адреси та дані передаються через 8 ліній BUS. Цикл читання (часова діаграма рис 2.5) або цикл запису(часова діаграма рис. 2.6) відбувається наступним чином:

1. Зміст регістру  $R0$  або  $R1$  виводиться на шину BUS.
2.  $ALE$  перевіряє валідність адрес.
3. Імпульс читання ( $\overline{RD}$ ) або запису ( $\overline{WR}$ ) подається на відповідні контакти мікроконтролера, і вказує що доступ до пам'яті даних в процесі.

4. Дані(8 біт) передаються в систему або на шину.

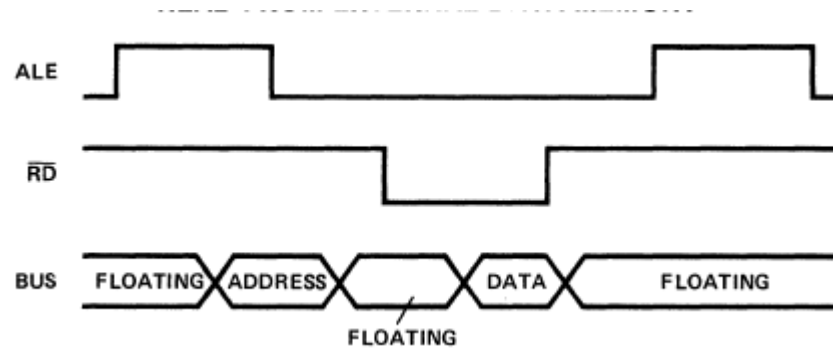


Рис 2.5. Часова діаграма читання з ЗПД.

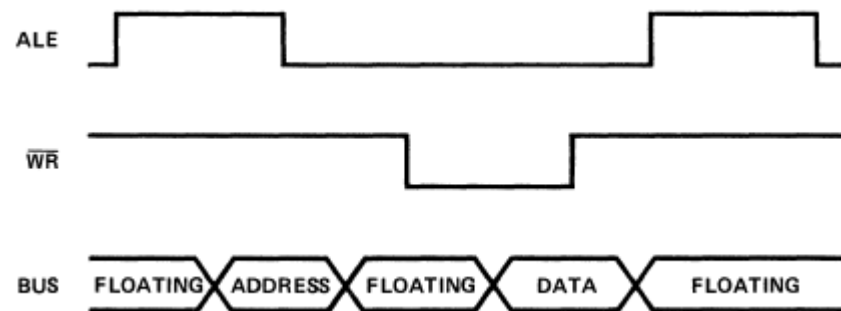


Рис. 2.6. Часова діаграма запису в ЗПД

### Підключення ЗПД

Обмін інформацією між МК48 та зовнішньою пам'яттю (як з пам'яттю програми, так і з пам'яттю даних) здійснюється в синхронному режимі, тобто на обмін виділяється фіксований проміжок часу, причому сигнали квитирування (зворотного зв'язку), узгодженні за часом роботи МК48 і пам'яті, не передбачаються. В зв'язку з цим таке узгодження можливо реалізувати тільки за рахунок вибору тактуючої частоти роботи МК48.

Для реалізації більш складних програмно-апаратних способів доступу до зовнішньої пам'яті, її ємність може бути збільшена до необхідного об'єму за рахунок сторінкової організації, при цьому зовнішня пам'ять даних поділяється на сторінки по 256 байт в кожній, а зовнішня пам'ять програм — на сторінки по 4К байт. Для переключення між сторінками можна використовувати, вільні лінії портів P1 та P2. Розглянемо підключення 8Кбайт зовнішньої пам'яті даних(Рис.2.7.).

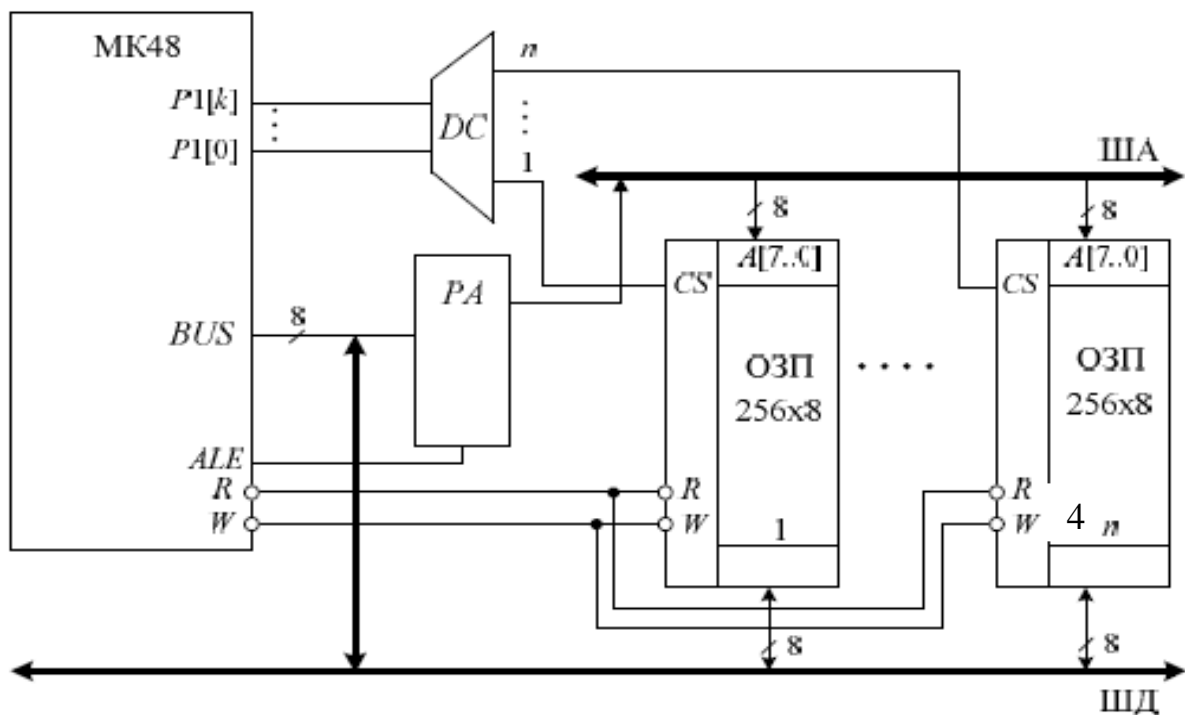


Рис 2.7. Підключення 8 Кбайт ЗПД

## 2.4. Додаткові порти

Для збільшення кількості ліній зв'язку МК48 з об'єктом управління підключають 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. 2.8.

В цьому випадку забезпечується виконання всіх чотирьох команд роботи з додатковими портами – MOVD A,P<sub>r</sub>; MOVD P<sub>r</sub>,A; ANLD P<sub>r</sub>,A та ORLD P<sub>r</sub>,A

(де

= 4, 7 ), причому кожний вихід порту може бути налаштований як на введення так і на виведення інформації.

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. 2.9. Для стробування даних в режимі підключення додаткових портів використовується сигнал PROG. Відмітимо, що логічні операції І та АБО виконуються безпосередньо в ІС ВР43. Це необхідно враховувати при побудові додаткових портів з використанням інших апаратних засобів.

Часова діаграма роботи з додатковими портами P4, P5, P6, P7 показана на рис. 2.10.

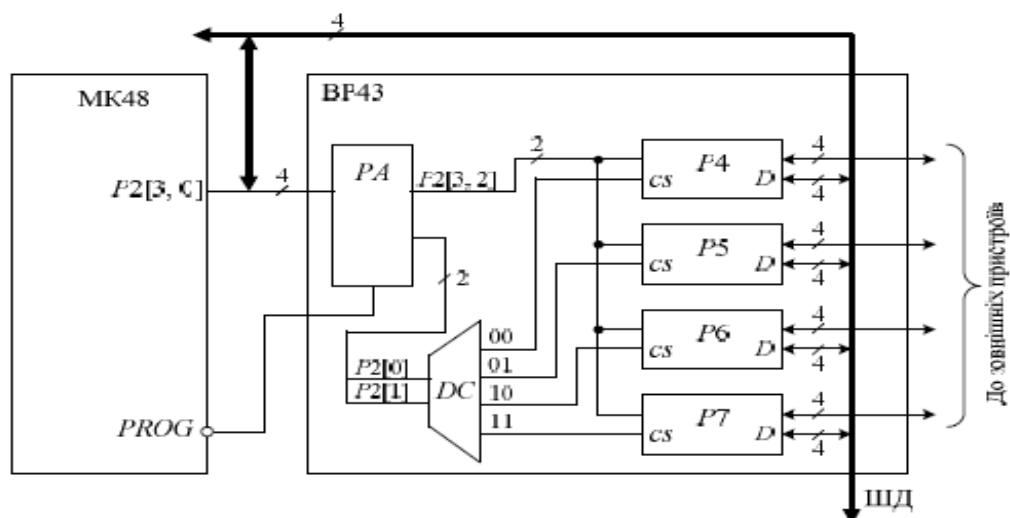


Рис 2.8. Схема з'єднання виходів МК48 та ІС КР580 ВР43

	3	2	1	0	
MOV D A <sub>r</sub> P <sub>p</sub>	0	0	0	0	P4
MOV D P <sub>p</sub> A	0	1	0	1	P5
ANLD P <sub>p</sub> A	1	0	1	0	P6
ORLD P <sub>p</sub> A	1	1	1	1	P7

Рис. 2.9. Структура управляючого слова

Для розширення функціональних можливостей системи до МК48 можна підключати різні ІС, наприклад, адаптери КР580ВВ51, КР580ВВ55 тощо.

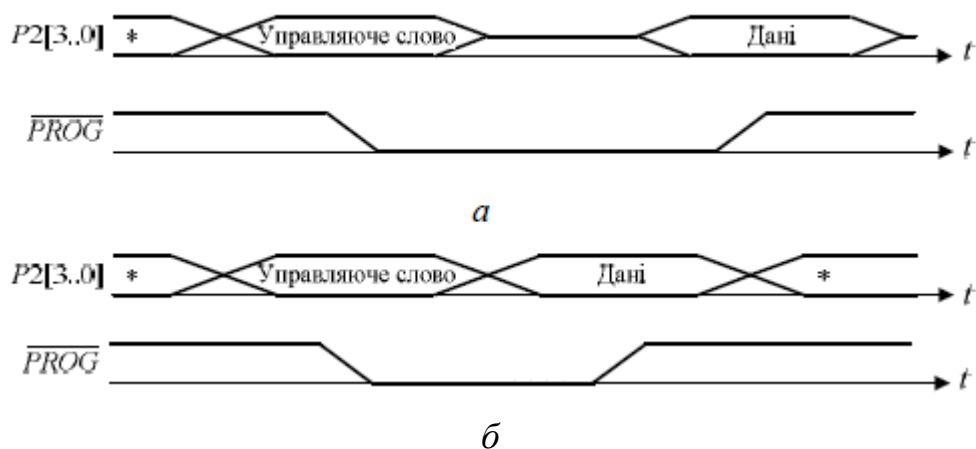


Рис. 2.10. Часова діаграма роботи з додатковими портами P4, P5, P6, P7: а- ввід даних, б- вивід даних

## 2.5. Таймер

### Опис таймеру

Таймер/лічильник TCNT являється восьмирозрядним лічильником, що підсумовує, який можна читати і завантажувати через акумулятор А, використовуючи відповідні команди MOV. Він може працювати в режимі таймера і в режимі лічильника. В режимі таймера на вхід TCNT через дільники частоти поступають сигнали з частотою  $F/480$ , де  $F$  — частота, що задається кварцовим резонатором або зовнішнім генератором. Наприклад, при  $F = 6$  МГц лічильник збільшує свій стан на 1 через кожні 80 мкс. Шляхом встановлення лічильника у певний вихідний стан і аналіз його переповнення можуть бути реалізовані різні часові затримки. Якщо 256 станів не забезпечують бажану затримку, то можна розрахувати декілька періодів роботи TCNT, накопичуючи в робочому регістрі необхідне число переповнень лічильника. Під час переходу TCNT із стану 255 в стан 0 ознака TF встановлюється в одиницю. Ця ознака може бути проаналізована командою JTF. Крім того, якщо переривання від TCNT дозволено командою EN TCNT, то при встановленні TF в 1 здійснюється перехід до підпрограми обслуговування переривання за вектором 0007. Переривання від TCNT може бути не дозволено командою DIS TCNTI. Після виконання команди JTF і при переході до підпрограми обслуговування переривання TF переходить в 0. В режимі лічильника події TCNT збільшує свій стан на 1 кожен раз, коли сигнал на вході T1 переходить із стану 1 в стан 0. В режимі таймера TCNT запускається командою STRT T, а в режимі лічильника — командою STRT CNT. Зупинка TCNT здійснюється командою STOP TCNT або системним скиданням.

### Команди управління таймером

В групу команд управління режимами роботи входять команди управління таймером/лічильником, перериваннями і ознаками переключення банків регістрів і банків пам'яті програм.

Таймер, залежно від застосованої команди, може бути використаний як лічильник тактів від внутрішнього джерела сигналів або як лічильник подій від зовнішнього джерела сигналів.

					ІАЛЦ 462619.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		15

Система команд МК48 має в своєму розпорядженні засоби дозволу або заборони переривання від таймера. Спеціальною командою ENTO на вивід TO дозволяється передача імпульсів з частотою тактового синхросигналу, діленою на три. Вивід цього сигналу може бути відключена тільки сигналом загального скидання. Син-

хросигнал на виході TO використовується для загальної синхронізації зовнішніх пристроїв, узгоджених з МК48 за частотою роботи.

Приклади команд:

MOV T,A ; завантаження таймеру.

STRT T ; запуск таймеру.

STRT CNT ; запуск лічильника.

EN TCNTI ; дозвіл переривання від таймеру.

DIS TCNTI ; заборона переривань від таймеру.

#### Приклад

MOV R5,#10 ; завантаження константи 10 в R5

MOV A,#FBH ; завантаження константи

; (- 5)ДК в таймер

MOV T,A

ORL P1,#2H ; встановлення сигналу Y5

STRT T ; запуск таймера

Label2: JTF Label1 ; відлік часового інтервалу

JMP Label2 ; з використанням таймеру

; (400мкс)

Label1: end

## 2.6. Контролер пріоритетних переривань

За технічним завданням система має децентралізований контролер переривань. Це означає, що в кожному зовнішньому пристрої є свій блок переривань, що формує відповідний сигнали. При складанні системи всі зовнішні пристрої об'єднуються в деїзі-ланцюжок у напрямку зниження пріоритетів. Такий спосіб забезпечує фіксовану систему пріоритетів (поміняти порядок з'єднання зовнішніх пристроїв на схемі вкрай складно), але в той же час є більш масштабованим (теоретично кількість зовнішніх пристроїв, що з'єднуються, не обмежена).

					ІАЛЦ 462619.004 ПЗ	Арк.
Зм.	Арк.	№ докum.	Підпис	Дат		16



Схема підключення зовнішніх пристроїв за таким принципом представлена на рис. 2.11.

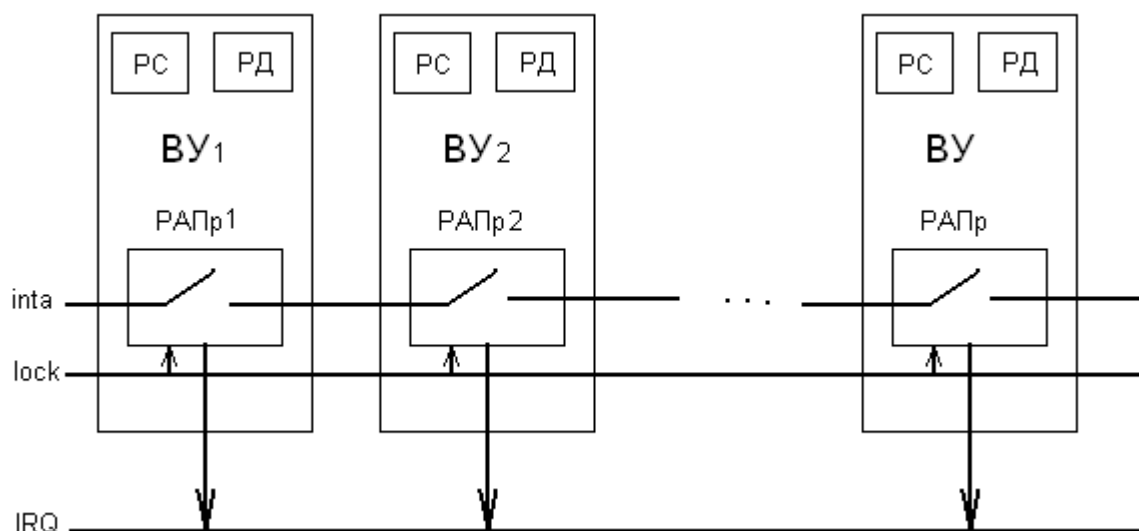


Рис. 2.11. Система децентралізованих переривань

Також можливо використовувати спеціальний регістр маски та за допомогою нього маскувати переривання від певних пристроїв.

## 2.7. Контролер прямого доступу до пам'яті

З метою звільнення центрального процесора від необхідності керувати пересиланням слова або масиву слів між оперативною пам'яттю і зовнішніми пристроями і надання процесору в цей час виконувати інші завдання в систему включений КППД – контролер прямого доступу до пам'яті.

Підвищення швидкості передачі даних відбувається за рахунок того, що КППД не звертався через системну магистраль за командами для роботи з пам'яттю, як це робив би процесор.

За таким же принципом, як і контролер пріоритетних переривань, контролер прямого доступу до пам'яті може бути централізованим (окрема мікросхема, що оброблює сигнали) і децентралізованим (організовувати пріоритетний деїзі-ланцюжок із зовнішніх пристроїв, в кожному з яких присутній розподілений арбітр прямого доступу до пам'яті). За технічним завданням в системі децентралізованих контролер прямого доступу до пам'яті. Схема типового децентралізованого контролера прямого доступу до пам'яті представлена на рис. 2.12.

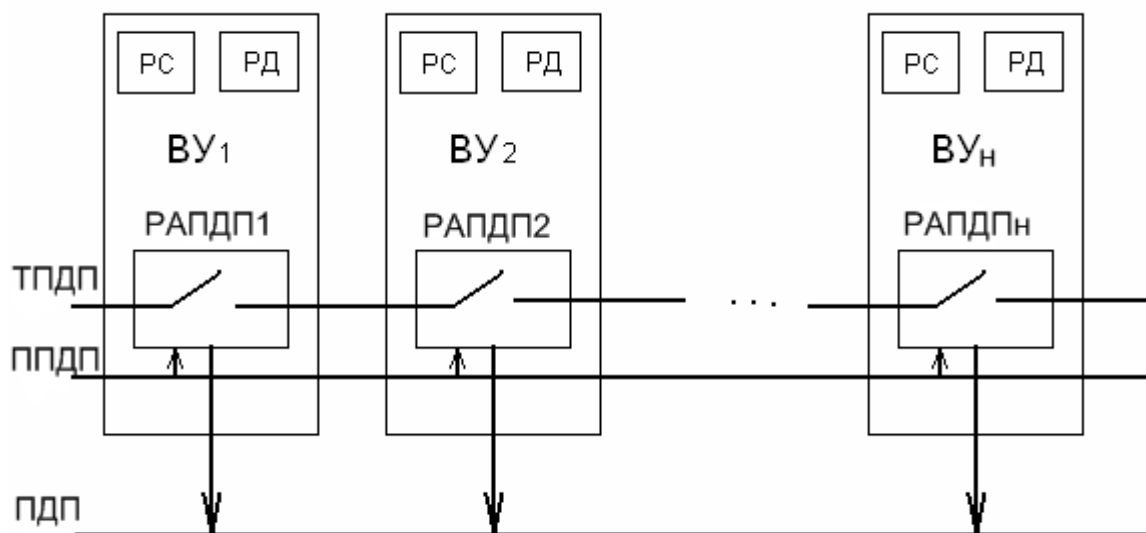


Рис. 2.12. Децентрализований контроллер прямого доступа к памяти

Для контроллера прямого доступа к памяти регистр маски, как правило, не применяется.

## 2.8. Порты вводу/выводу

Мікроконтролер вміщує три порти вводу/виводу: P1, P2 і BUS. Порти P1 і P2 називають «квазідвоспрямованими», структурна схема якого представлена на рис 2.13. Їх особливість полягає в тому, що при вводі даних, над ними та поточним станом порту (даними, які виводилися із порту останніми) виконується порозрядна логічна операція І. Вихідні дані в порту запам'ятовуються.

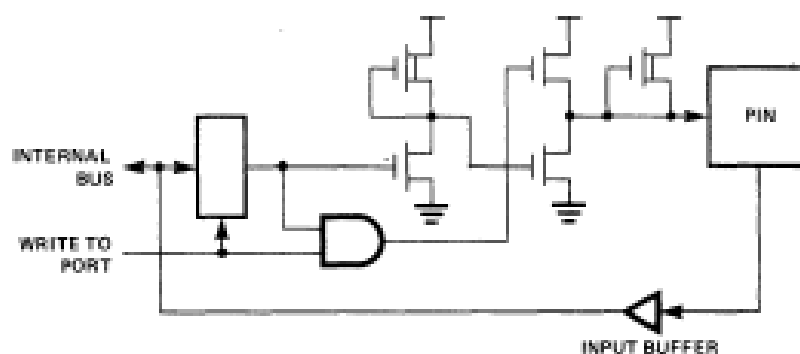


Рис 2.13. Структурна схема квазідвоспрямованого порту

При скиданні системи кожному розряду порту присвоюється значення 1. У системі команд МК48 є команди, які дозволяють виконувати запис нулів і одиниць в будь-якому розряді або групі розрядів порту, але оскільки в цих командах маска задається безпосереднім операндом, то необхідно знати розподіл ліній, що скидаються і встановлюваних, на етапі розробки прикладної програми.

В тому випадку, якщо маска обчислюється програмою і наперед не відома, в ОЗП необхідно мати копію стану порту виведення. Ця копія по командах логічних операцій об'єднується з обчислюваною маскою в акумуляторі і потім завантажуються в порт. Необхідність цієї процедури викликана тим, що в МК48 відсутня можливість виконати операцію читання значень портів P1 і P2 для визначення колишнього стану порту виведення.

Порт BUS має звичайні двоспрямовані виходи з трьома станами. Порт застосовується для побайтного вводу/виводу даних. За допомогою команд ORL і ANL можливо маскувати байти, що передаються через порт, з ціллю обробляти у байті окремі біти або групу бітів. Команди звернення до портів включають безпосередній номер порту. У мікропроцесорних системах простої конфігурації, коли порт BUS не використовується як порт-розширювач системи, обмін

виконується по командах INS, OUTU і MOVX. Можливе поперемінне використання команд OUTL і MOVX. Проте при цьому необхідно пам'ятати, що байт, який виводиться по команді OUTL фіксується в буферному регістрі порту BUS, а команда MOVX знищує вміст буферного регістра порту BUS. Команда INS не знищує вміст буферного регістра порту. В МПС що мають зовнішню пам'ять програм, порт BUS використовується для видачі адреси зовнішній пам'яті і для прийому команди із зовнішньої пам'яті програм. Отже, в таких системах використання команди OUTL позбавлене сенсу.

## 2.9. Спеціалізований обчислювач

### 2.9.1. Опис задачі

До складу мікропроцесорної системи входить спеціалізований обчислювач, який призначений для вирішення спеціалізованої задачі, а саме: зважена середньоквадратична апроксимація таблично заданої функції  $f(x)$ .

Програма призначена для апроксимації методом найменших квадратів функції  $y = f(x)$ , якщо задані її значення  $y_i = f(x_i)$  в вузлах  $x_i$  ( $i=1,..,n_1$ ) і задані відповідні ваги  $p_i$  ( $i=1,..,n_1$ )

Апроксимуючий поліном  $P_m(x) = \sum_{k=0}^m a_k x^k$  будується таким чином, щоб квадратичне відхилення

$$S = \sum_{i=1}^{n_i} [P_m(x_i) - y_i]^2$$

було мінімальним.

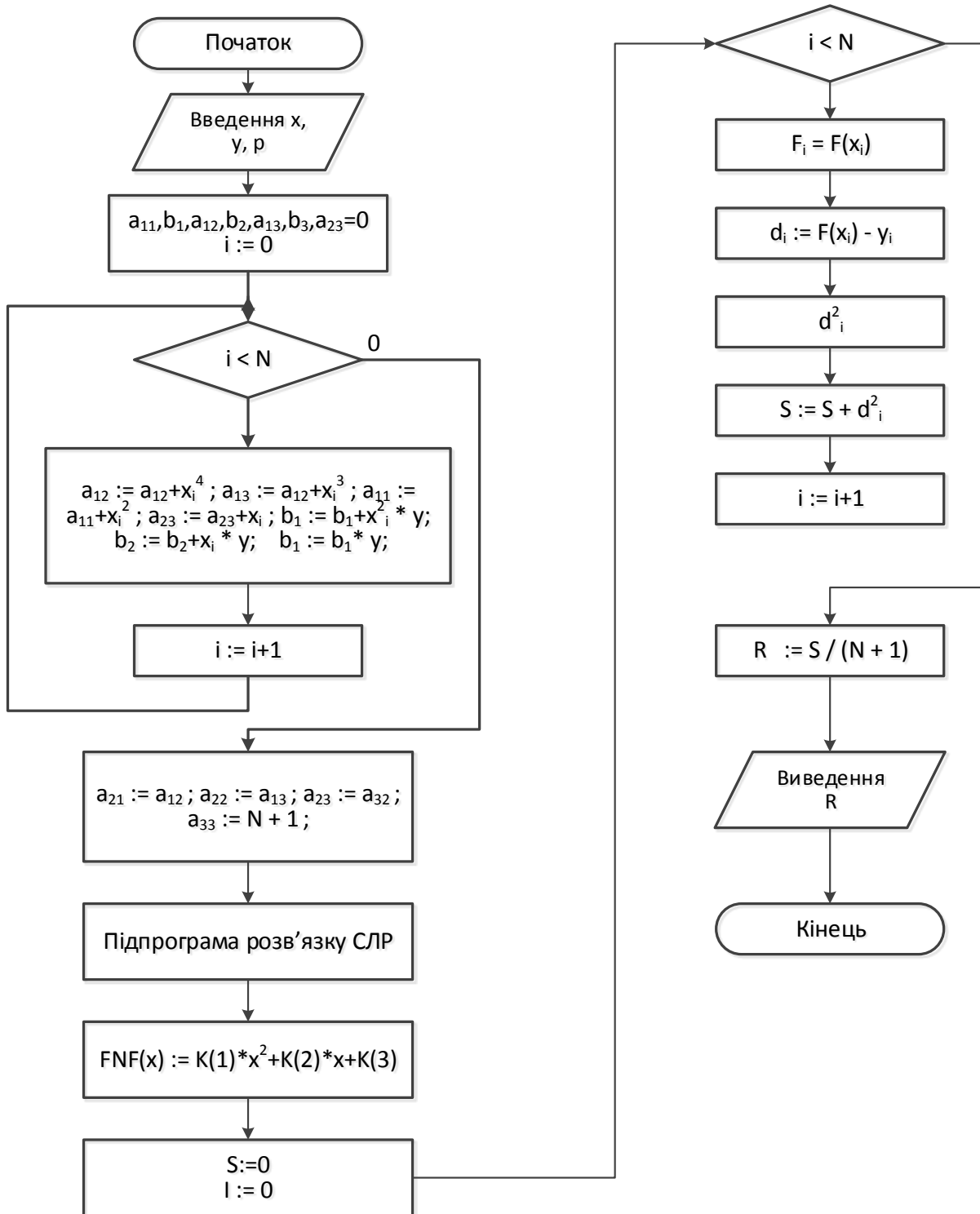
В такому випадку для знаходження коефіцієнтів  $a_0, a_1, \dots, a_m$  розв'язується система лінійних алгебраїчних рівнянь  $Ca = d$ , де  $C$  – матриця системи,  $a$  – вектор невідомих,  $d$  – вектор правих частин.

Після визначення коефіцієнтів  $a_i$  в програмі обчислюється середньоквадратична похибка апроксимації  $T$ :

$$T = \sqrt{\frac{\sum_{i=1}^{n_i} p_i [P_m(x_i) - y_i]^2}{n_i}}$$

					ІАЛЦ 462619.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		20

## 2.9.2 Алгоритм розв'язання спеціалізованої задачі



### 2.9.3. Система команд

Для вирішення даної задачі використовується, набір команд, відображених у табл. 2.1

Таблиця 2.1 Система команд

Команда	Опис
MOV A,Rn	$(A) \leftarrow (Rn)$
MOV A,ad	$(A) \leftarrow (ad)$
MOV A,@Ri	$(A) \leftarrow ((Ri))$
MOV A,#d	$(A) \leftarrow \#d$
MOV @Ri,ad	$((Ri)) \leftarrow (ad)$
XCH A,Rn	$(A) \leftrightarrow (Rn)$
ADD A,Rn	$(A) \leftarrow (A) + (Rn)$
ADD A,ad	$(A) \leftarrow (A) + (ad)$
ADDC A,@Ri	$(A) \leftarrow (A) + ((Ri)) + (C)$
ADDC A,#d	$(A) \leftarrow (A) + \#d + (C)$
INC A	$(A) \leftarrow (A) + 1$
JZ rel	$(PC) = (PC) + 2,$ Якщо $(A) = 0,$ то $(PC) = (PC) + rel$
JNC rel	$(PC) = (PC) + 2,$ Якщо $(C) = 0,$ то $(PC) = (PC) + rel$
DJNZ ad, rel	$(PC) = (PC) + 2,$ $(ad) = (ad) - 1,$ Якщо $(ad) = 0,$ то $(PC) = (PC) + rel$
CALL ad11	$(PC) = (PC) + 2,$ $(SP) = (SP) + 1,$ $((SP)) = (PC0-7),$ $(SP) = (SP) + 1,$ $((SP)) = (PC8-15),$ $(PC0-10) = ad11$
RET	$(PC8-15) = ((SP)),$ $(SP) = (SP) - 1,$ $(PC0-7) = ((SP)),$ $(SP) = (SP) - 1$

## Розділ 3. Програмна частина

### 3.1. Система команд

В таблиці 3.1 представлені основні групи команд для мікроконтролера 1816BE48

*Табл. 3.1. Система команд МК 1816BE48*

Мнемоніка	Код команди	Коментарі
1	2	3
<b>Основна група команд та команди пересилки даних</b>		
<i>Команди звернення до акумулятора</i>		
CLR A	00100111	Встановлення вмісту акумулятора в нуль $A := 0$
CPL A	00110011	Інвертування вмісту A; $A := NOT A$
INC A	00010111	Інкремент вмісту A; $A := A + 1$
DEC A	00000111	Декремент вмісту A; $A := A - 1$
RR A	01110111	Циклічний зсув вмісту A вправо; $A[7] := A[0]$ ; $A[i] := A[i + 1]$ ; $i = \overline{6, 0}$
RL A	11100111	Циклічний зсув вмісту A вліво; $A[0] := A[7]$ ; $A[i] := A[i + 1] := A[i]$ ; $i = \overline{6, 0}$
RRC A	01100111	Циклічний зсув вмісту A з бітом переносу вправо; $A[7] := C$ ; $C := A[0]$ ; $A[i] := A[i + 1]$ ; $i = \overline{6, 0}$
RLC A	11110111	Циклічний зсув вмісту A з бітом переносу вліво; $A[0] := C$ ; $C := A[7]$ ; $A[i + 1] := A[i]$ ; $i = \overline{6, 0}$
SWAP A	01000111	Обмін тетрадами A; $A[7..4] \leftrightarrow A[3..0]$
DA A	01010111	Десяткова корекція вмісту A
MOV A, Rr ; $r = (7-0)$	11111rrr	Пересилка вмісту регістру в A; $A := Rr$
MOV Rr, A ; $r = (7-0)$	10101rrr	Пересилка вмісту A в регістр; $Rr := A$
XCH A, Rr ; $r = (7-0)$	00101rrr	Обмін вмісту A і регістру; $A \leftrightarrow Rr$
ANL A, Rr ; $r = (7-0)$	01011rrr	Логічне І вмісту A і регістру; $A := A AND Rr$
ORL A, Rr ; $r = (7-0)$	01001rrr	Логічне АБО вмісту A і регістру; $A := A OR Rr$
XRL A, Rr ; $r = (7-0)$	11011rrr	Виключне АБО вмісту A і регістру; $A := A XOR Rr$
ADD A, Rr ; $r = (7-0)$	01101rrr	Сума вмісту A і регістру; $A := A + Rr$
ADDC A, Rr ; $r = (7-0)$	01111rrr	Сума вмісту A, регістру і переносу C; $A := A + Rr + C$
DEC Rr ; $r = (7-0)$	11001rrr	Декремент вмісту регістру; $Rr := Rr - 1$
INC Rr ; $r = (7-0)$	00011rrr	Інкремент вмісту регістру; $Rr := Rr + 1$
<i>Команди звернення до внутрішньої пам'яті даних</i>		
MOV A, @Rr ; $r = 0, 1$	1111000r	Пересилка із внутрішньої пам'яті даних в A; $A := (Rr)$
MOV @Rr, A ; $r = 0, 1$	1010000r	Пересилка вмісту A до внутрішньої пам'яті даних; $(Rr) := A$
XCH A, @Rr ; $r = 0, 1$	0010000r	Обмін вмістом A і комірки внутрішньої пам'яті даних; $A \leftrightarrow Rr$
XCHD A, @Rr ; $r = 0, 1$	0011000r	Обмін молодшими тетрадами A і комірки внутрішньої пам'яті даних; $A[3..0] \leftrightarrow (Rr)[3..0]$
ANL A, @Rr ; $r = 0, 1$	0101000r	Логічне І вмісту A і комірки внутрішньої пам'яті даних; $A := A AND (Rr)$
ORL A, @Rr ; $r = 0, 1$	0100000r	Логічне АБО вмісту A і комірки резидентної пам'яті даних; $A := A OR (Rr)$
XRL A, @Rr ; $r = 0, 1$	1101000r	Виключення АБО вмісту A і комірки резидентної пам'яті даних; $A := A XOR (Rr)$

1	2	3
ADD A, @Rr ; r = 0, 1	0110000r	Сума вмісту A і комірки резидентної пам'яті даних; $A := A + (Rr)$
ADDC A, @Rr ; r = 0, 1	0111000r	Сума вмісту A, комірки резидентної пам'яті даних і переносу C; $A := A + (Rr) + C$
INC @Rr ; r = 0, 1	0001000r	Інкремент комірки резидентної пам'яті даних; $(Rr) := (Rr) + 1$
<i>Команди роботи з зовнішньою пам'яттю даних</i>		
MOVX A, @Rr ; r = 0, 1	1000000r	Пересилка із ЗПД в A; $A := (Rr)$
MOVX @Rr, A ; r = 0, 1	1001000r	Пересилка вмісту A до ЗПД; $(Rr) := A$
<i>Команди звернення до пам'яті програми</i>		
MOV Rr, #d ; r = (7-0)	10111rrr ddddddddd	Пересилка безпосереднього операнда до регістру; $(Rr) := d$
MOV A, #d	00100011 ddddddddd	Пересилка безпосередньої адреси до A $A := d$
MOV @Rr, #d ; r = 0, 1	1011000r ddddddddd	Пересилка безпосереднього операнда до внутрішньої пам'яті даних $(Rr) := d$
ANL A, #d	01010011 ddddddddd	Логічне І вмісту A з безпосереднім операндом; $A := A \text{ AND } d$
ORL A, #d	01000011 ddddddddd	Логічне АБО вмісту A з безпосереднім операндом; $A := A \text{ OR } d$
XRL A, #d	11010011 ddddddddd	Виключне АБО вмісту A з безпосереднім операндом; $A := A \text{ XOR } d$
ADD A, #d	00000011 ddddddddd	Сума вмісту A та безпосереднього операнду; $A := A + d$
ADDC A, #d	00010011 ddddddddd	Сума вмісту A, безпосереднього операнду та переносу C; $A := A + d + C$
MOVP A, @A	10100011	Пересилка даних із поточної сторінки пам'яті програм до A; $A := (PC[11..8].A)$
MOV P3 A, @A	11100011	Пересилка даних із сторінки 3 пам'яті програм до A; $A := (0011.A)$
<i>Команди звернення до регістру PSW</i>		
MOV PSW, A	11010111	Пересилка вмісту A до регістру PSW; $PSW := A$
MOV A, PSW	11000111	Пересилка вмісту регістру PSW до A; $A := PSW$
MOV A, T	01000010	Пересилка вмісту TCNT в A; $A := TCNT$
MOV T, A	01100010	Пересилка вмісту A в TCNT; $TCNT := A$
<i>Команди встановлення ознак</i>		
CLR C	10010111	Встановлення в нуль ознаки C; $C := 0$
CPL C	10100111	Інвертування ознаки C; $C := NOT C$
CLR F0	10000101	Встановлення в нуль ознаки F0; $F0 := 0$
CLR F1	10100101	Встановлення в нуль ознаки F1; $F1 := 0$
CPL F0	10010101	Інвертування ознаки F0; $F0 := NOT F0$
CPL F1	10110101	Інвертування ознаки F1; $F1 := NOT F1$
<i>Команди звернення до портів P1 і P2</i>		
ANL Pp, #d ; p = 1, 2	100110pp ddddddddd	Логічне І порту P1(P2) з безпосереднім операндом; $Pp := Pp \text{ AND } d$
ORL Pp, #d ; p = 1, 2	100010pp ddddddddd	Логічне АБО порту P1(P2) з безпосереднім операндом; $Pp := Pp \text{ OR } d$
IN A, Pp ; p = 1, 2	000010pp	Введення даних із порту P1(P2) в A; $A := Pp$
OUTL Pp, A ; p = 1, 2	001110pp	Виведення вмісту A в порт P1(P2) $Pp := A$



1	2	3
<u>Команди звернення до портів P4, P5, P6, P7</u>		
ANLD Pr, A ; p = (7-4)	100111pp	Логічне І порту P4 (P5, P6, P7) з A; $P_r := P_r \text{ AND } A[3..0]$
ORLD Pr, A ; p = (7-4)	100011pp	Логічне АБО порту P4 (P5, P6, P7) з A; $P_r := P_r \text{ OR } A[3..0]$
MOVD A, Pr ; p = (7-4)	000011pp	Ввід із порту P4 (P5, P6, P7) в A; $A[7..4] := 0$ ; $A[3..0] := P_r$
MOVD Pr, A ; p = (7-4)	001111pp	Вивід молодшої тетради із A в порт P4 (P5, P6, P7); $P_r := A[3..0]$
<u>Команди звернення до порту BUS</u>		
ANL BUS, #d	10011000 ddddddddd	Логічне І порту BUS з безпосереднім операндом; $BUS := BUS \text{ AND } d$
ORL BUS, #d	10001000 ddddddddd	Логічне АБО порту BUS з безпосереднім операндом; $BUS := BUS \text{ OR } d$
INS A, BUS	00001000	Ввід даних із порту BUS в A; $A := BUS$
OUTL BUS, A	00000010	Вивід вмісту A в порт BUS; $BUS := A$
<u>Команди передачі управління</u>		
JMP a	aaa00100 aaaaaaaaa	Безумовний перехід $PC[10..0] := a[10..0]$ ; $PC[11] := MB$
JMPP @A	10110011	Безумовний перехід в межах поточної сторінки; $PC[7..0] := (A)$
JC a	11110110 aaaaaaaaa	Перехід, якщо C = 1, то $PC[7..0] := a$ інакше $PC := PC + 2$
JNC a	11100110 aaaaaaaaa	Перехід, якщо C = 0
DJNZ Rr, a	11101rrr aaaaaaaaa	Декремент вмісту регістру і перехід, якщо вміст регістру не дорівнює нулю
JZ a	11000110 aaaaaaaaa	Перехід, якщо вміст A дорівнює нулю
JNZ a	10010110 aaaaaaaaa	Перехід, якщо вміст A не дорівнює нулю
JF0 a	10110110 aaaaaaaaa	Перехід, якщо F0 = 1
JF1 a	01110110 aaaaaaaaa	Перехід, якщо F1 = 1
JT0 a	00110110 aaaaaaaaa	Перехід, якщо T0 = 1
JNT0 a	00100110 aaaaaaaaa	Перехід, якщо T0 = 0
JT1 a	01010110 aaaaaaaaa	Перехід, якщо T1 = 1
JNT1 a	01000110 aaaaaaaaa	Перехід, якщо T1 = 0
JTF a	00010110 aaaaaaaaa	Перехід, якщо TF = 1
JNI a	10000110 aaaaaaaaa	Перехід, якщо INT' = 0
JBb a	bbb10010 aaaaaaaaa	Перехід, якщо розряд Bb акумулятора встановлений в одиницю, де b = (7-0)
CALL a	aaa10100 aaaaaaaaa	Виклик підпрограми; $SP := SP + 1$ ; $(SP) := PSW[7..4]$ ; $PC[11] := MB$ ; $PC[10..0] := a[10..0]$
RET	10000011	Повернення із підпрограми; $SP := SP - 1$ ; $PC := (SP[11..0])$

1	2	3
RETR	10010011	Повернення із підпрограми з встановленням стану; $SP := SP - 1$ ; $PC := SP[11..0]$ ; $PSW[7..4] := (SP[15..12])$
<b>Команди управління режимами роботи</b>		
ENTO CLK	01110101	Дозвіл видачі імпульсів синхронізації на $T0$
SEL MBO	11100101	Вибір нульового банку пам'яті програм; $MB := 0$
SEL MB1	11110101	Вибір першого банку пам'яті програм; $MB := 1$
SEL RBO	11000101	Вибір нульового банку регістрів пам'яті даних; $RB := 0$
SEL RB1	11010101	Вибір першого банку регістрів пам'яті даних; $RB := 1$
NOP	00000000	Немає операції
EN I	00000101	Дозвіл зовнішніх переривань
DIS I	00010101	Заборона зовнішніх переривань
EN TCNTI	00100101	Дозвіл переривань від таймера/лічильника
DIS TCNTI	00110101	Заборона переривань від таймера/лічильника
STRT T	01010101	Запускання таймера/лічильника в режимі таймера
STRT CNT	01000101	Запускання таймера/лічильника в режимі лічильника
STOP TCNT	01100101	Зупинка таймера/лічильника

### 3.2. Робота з плаваючою комою

Згідно з завданням, необхідно виконати операції добування квадратного кореню.

Для цього – необхідно використати трибайтову мантису у формі подання – доповняльний код, а форма подання порядку – симетричний. Операційна схема представлена на рис. 3.1, а алгоритм цієї операції представлено на рис. 3.2.

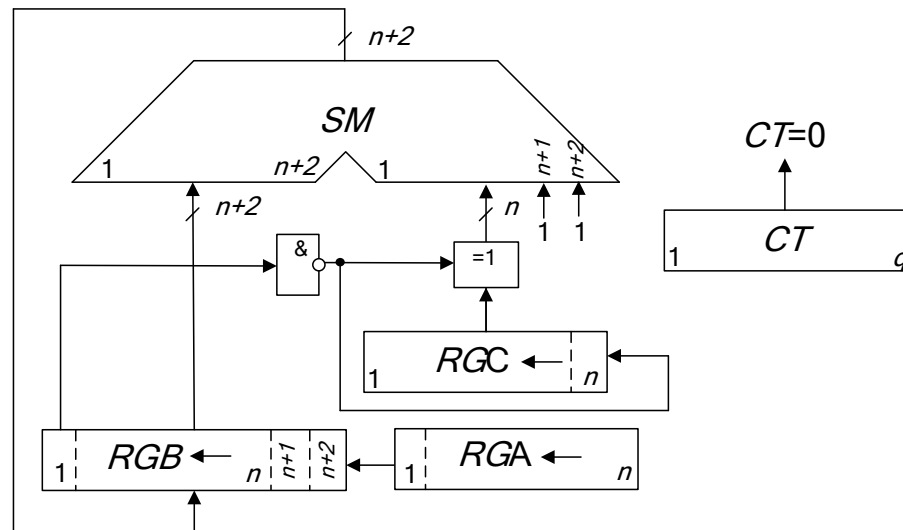


Рис 3.1. Операційна схема добування квадратного кореня

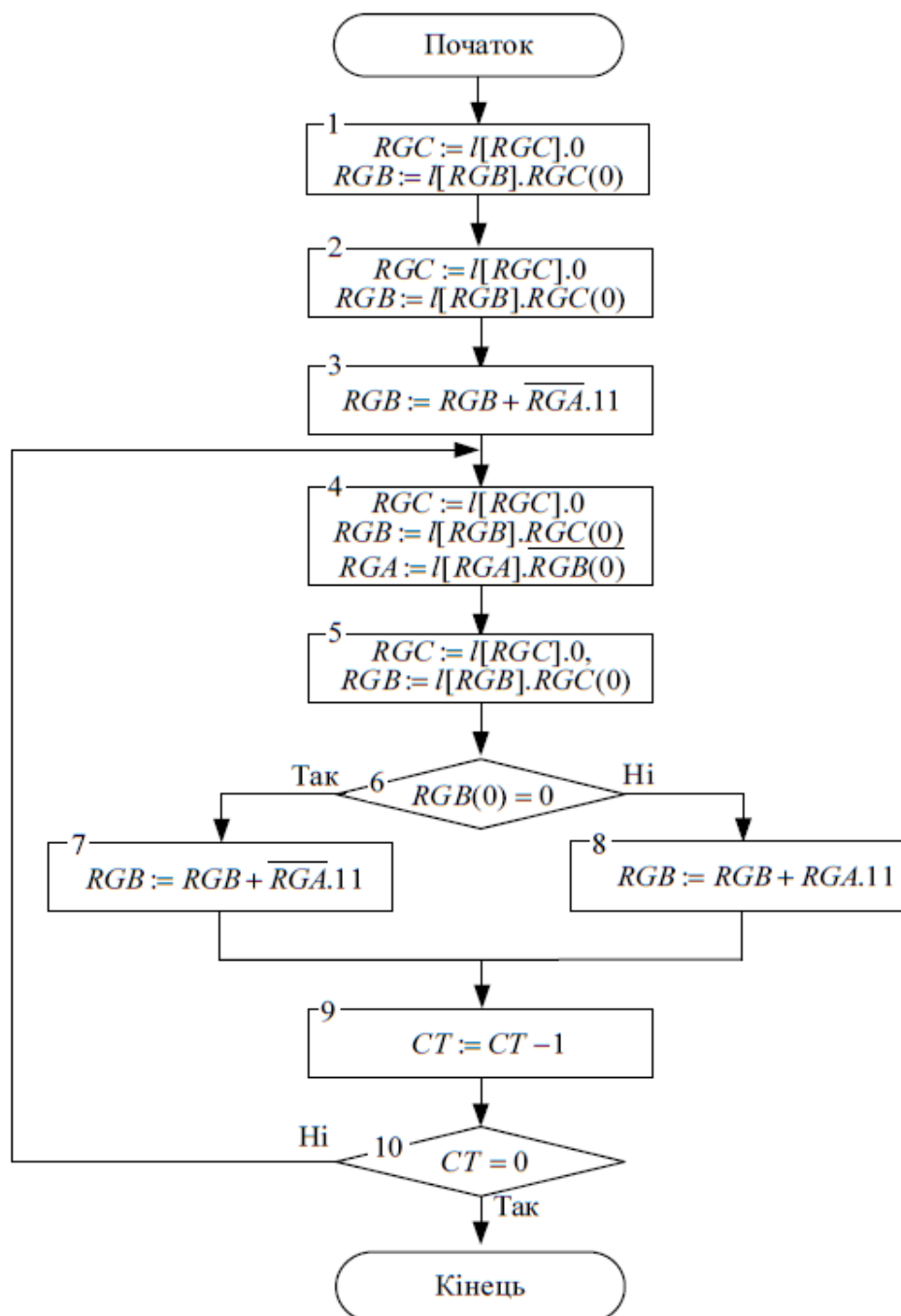


Рис. 3.2. Алгоритм операції здобування квадратного кореню

#### Лістинг програми

INS A,BUS ;	; завдання кількості по-
MOV R2,A ; A - завантаження	вторень циклу CT:=4
X	MOV R3,#4h
MOV R2,#10010001 ;	LL3: MOV A,R0
MOV R1,#0h ; B - для під-	MOV R4,A ; R4:=A(R4:=A.11)
сумовування	MOV R5,#0h ; B[ZN] :=0
MOV R0,#0h ; C - для	; два зсуви вліво A, B
; зберігання результату	CLR C
	MOV A,R2

```

RLC A
MOV R2,A
MOV A,R1
RLC A
MOV R1,A
CLR C
MOV A,R2
RLC A
MOV R2,A
MOV A,R1
RLC A
MOV R1,A
JB7 LL1 ; аналіз B[ZN]
MOV A,R4
CPL A
MOV R4,A ; R4:=!A
LL1: MOV A,R4
RLC A

```

```

RLC A
ORL A,#3h
MOV R4,A ; R4:=A.11
ADD A,R1
MOV R1,A
JB7 LL2
MOV R5,#1h ; !B[ZN]:=1
LL2: CLR C
MOV A,R0
RLC A
ADD A,R5 ; A[N]:=!B[ZN]
MOV R0,A
DJNZ R3,LL3 ; перевірка циклу
MOV A,R0
OUT BUS,A
END

```

## Розділ 4. Розробка функціональної й принципової схеми заданого в курсовому проекті функціонального блоку

### 4.1. Опис структурно-функціональної схеми мікропроцесорної системи

Для побудови структурно-функціональної схеми *ІАЛЦ 462619.005 Е1*, було обрано мікроконтролер 1816ВЕ48 до нього, для створення більш складних програм, під'єднано 3 пристрої зовнішньої пам'яті програм (ПП1 – ПП3) по 1К кожний. З'єднання відбувається через дешифратор ДС(1).

Для розширення оперативного запам'ятовуючого пристрою було підключено 4 пристрої зовнішньої пам'яті даних (ПД1 – ПД4) по 256х8байти кожний.

Для збільшення кількості ліній зв'язку МК48 з об'єктом управління підключають додаткові чотирирозрядні порти  $P_4, P_5, P_6, P_7$ . Найбільш просто це здійснюється за використання спеціальної ІСКР580 ВР43. Спосіб підключення зображено на структурно-функціональній схемі забезпечує виконання всіх чотирьох команд роботи з додатковими портами –  $MOV D A, P_r; MOV D P_r, A; ANLD P_r, A$  та  $ORLD P_r, A$  (де  $p = 4, 7$ ), причому кожний вихід порту може бути налаштований як на введення такі на виведення інформації.

Для розробки програмованих пристроїв для вводу/виводу в МПС підключено програмований периферійний адаптер 580ВВ55. Програмований периферійний адаптер 580ВВ55 виготовляється за *NMOS* технології. Може бути застосований у МПС з мікропроцесорами МК51, МК48, МП8086.

До складу ППА входять наступні функціональні блоки.

- буфер шини даних (ШД)  $D_1 - D_0$ ;
- схема управління читанням/записом даних в регістри ППА;
- група А, порт  $PA$  – порт вводу/виводу  $PA$  групи А;
- група В, порт  $PB$  – порт вводу/виводу  $PB$  групи В;
- група С, порт  $PC$  – порт вводу/виводу  $PC_h$  групи А;
- група В, порт  $PC$  – порт вводу/виводу  $PC_l$  групи В;
- схема управління портами групи А: порти  $PA$  та  $PC_h$ ;
- схема управління портами групи А: порти  $PB$  та  $PC_l$ .

Схеми управління портами групи А та В містять регістр управління, що задає режими роботи портів. Всі порти оснащені буферними регістрами, че-

рез які здійснюється зв'язок між ППА і зовнішніми шинами. Програмування режимів роботи і управління ІС здійснюється мікропроцесором за допомогою сигналів  $\overline{M}$  –  $\overline{D}$ ,  $A_1$ ,  $A_0$ ,  $\overline{CS}$ ,  $\overline{RD}$ ,  $\overline{WR}$ , RESET.

Адреси портів ППА входять до загального адресного простору зовнішньої пам'яті даних. Доступ до портів під час запису та читання здійснюється за застосування команд MOVX A,@R<sub>r</sub>; MOVX @R<sub>r</sub>,A (де,  $r = 1, 0$ ).

Відмітимо, що по шині даних відбувається не тільки обмін даними, але і пересилання з МК48 в ППА управляючих слів, генерованих програмним забезпеченням процесора, а також передача в МК48 інформації про стан периферійного обладнання. Низький рівень сигналу на вході вибору кристалу  $\overline{CS}$  дозволяє інформаційний зв'язок між ППА і МП48.

Також, для розширення функцій МПС підключено 86 зовнішніх пристроїв.

#### 4.2. Побудова принципової схеми

На принциповій схемі зображено децентралізовані КПП і КПДП, а також – інтерфейси ЗП.

##### Централізований КПП

До переваг централізованих контролерів слід віднести наступне:

- можливість динамічно змінювати стратегію обслуговування заявок;
- швидке вибіркове маскування запитів на переривання.

Централізований КПП потенційно дозволяє забезпечувати різні дисципліни обслуговування заявок, оскільки всі заявки поступають в один пристрій. Для забезпечення пріоритетного обслуговування застосовується блок вибору пріоритету. У простому випадку в якості БВП використовується пріоритетний шифратор(ПШ), який формує код старшого рівня пріоритету. При цьому всі запити мають фіксовані пріоритети, рівень яких визначається номером входу КПП.

Оскільки регістр маски включений в адресний простір процесора, то за умови, що число запитів не перевищує розрядності шини даних(що зазвичай виконується), процесор може за одне звернення до цього регістра записати в нього будь-яку маску.

До недоліків централізованих КПП слід віднести:

- велику кількість ліній запитів в шині управління (що дорівнює кількості ЗП);

					ІАЛЦ 462619.004 ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дат		

- обмеження на максимальне число джерел переривань;
- можлива неоднорідність процесорних модулів.

### Централізований КПДП

З метою звільнення центрального процесора від необхідності керувати пересилкою слова або масиву слів між оперативною пам'яттю і зовнішніми пристроями та надання процесору в цей час виконувати інші завдання, в систему включений КПДП – контролер прямого доступу до пам'яті. Підвищення швидкості передачі даних відбувається за рахунок того, що КПДП не звертається через системну магістраль за командами для роботи з пам'яттю, як це робив би процесор. Точно за таким же принципом, як і контролер пріоритетних переривань, контролер прямого доступу до пам'яті може бути централізованим (окрема мікросхема, обробляє сигнали від зовнішніх пристроїв і передає результати центрального процесора) і децентралізованим (організовувати пріоритетний Дейзі-ланцюжок із зовнішніх пристроїв, в кожному з яких присутній розподілений арбітр прямого доступу до пам'яті). За технічним завданням в системі централізованих контролерів прямого доступу до пам'яті.

					ІАЛЦ 462619.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		31

## Розділ 5. Розробка заданого вузла ЕОМ на ПЛІС

### 5.1. Опис математичної складової АЛП

АЛП забезпечує обчислення функції  $Z = x^2 + 4Y$ . Возведення в 2 ступінь буде проводитися множенням  $X$  самого на себе, виходячи з ТЗ, множення виконується четвертим способом: здійснюється зі старших розрядів множника, множене зсувається вправо, а сума часткових добутків залишається нерухомою. Множення на 4 здійснюється за допомогою чотирьох послідовних зсувів вліво регістру із значенням  $Y$ .

### 5.2. Функціональна схема АЛП

Функціональна схема арифметико-логічного пристрою зображена на рисунку 5.1. Видно, що схема базується на операційній схемі множення двох чисел четвертим способом, але відрізняється наявністю додаткового регістру із початковим значенням  $Y$  та мультимплексором для того, щоб додати до результату множення значення  $4Y$ . Призначення регістрів:

RG1 — результат множення, кінцевий результат;

RG2 — множник ( $X$ ); RG3 — множене ( $X$ );

RG4 — значення  $Y$  та значення  $4Y$ .

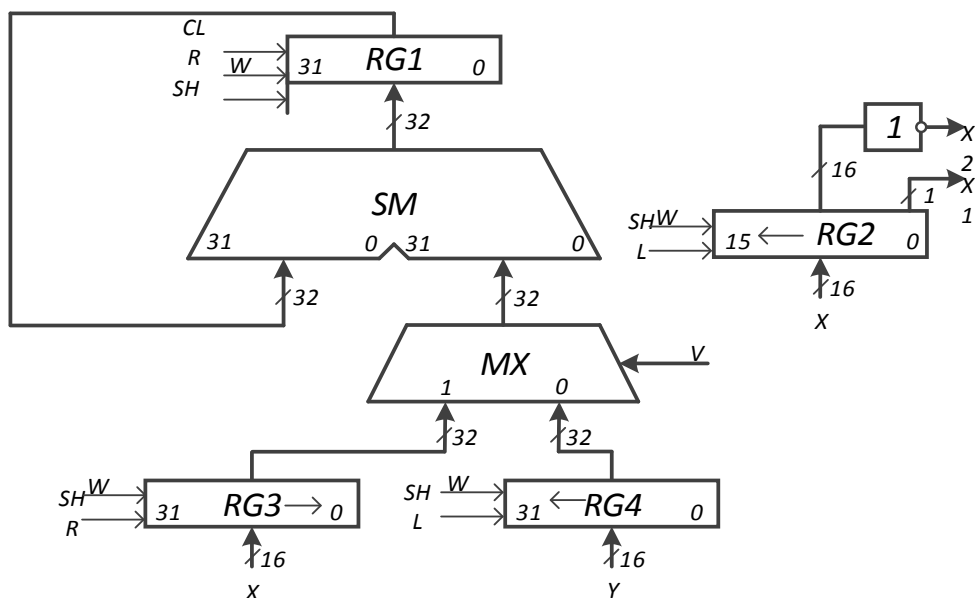


Рис.5.1. Функціональна схема АЛП



### 5.3. Змістовний та закодований мікроалгоритми

На рисунку 5.2 зображені змістовний, а на рис. 5.3. – закодований мікроалгоритми. У цілому, мікроалгоритм складається з алгоритму множення двох чисел четвертим способом (верхня частина) та операції завантаження, зсуву та додавання до  $x^2$  значення  $4Y$ .

Закодовані мікрооперації та сигнали приведені, відповідно у таблицях 5.1 та 5.2

					ІАЛЦ 462619.004 ПЗ	Арк.
						33
Зм.	Арк.	№ докум.	Підпис	Дат		

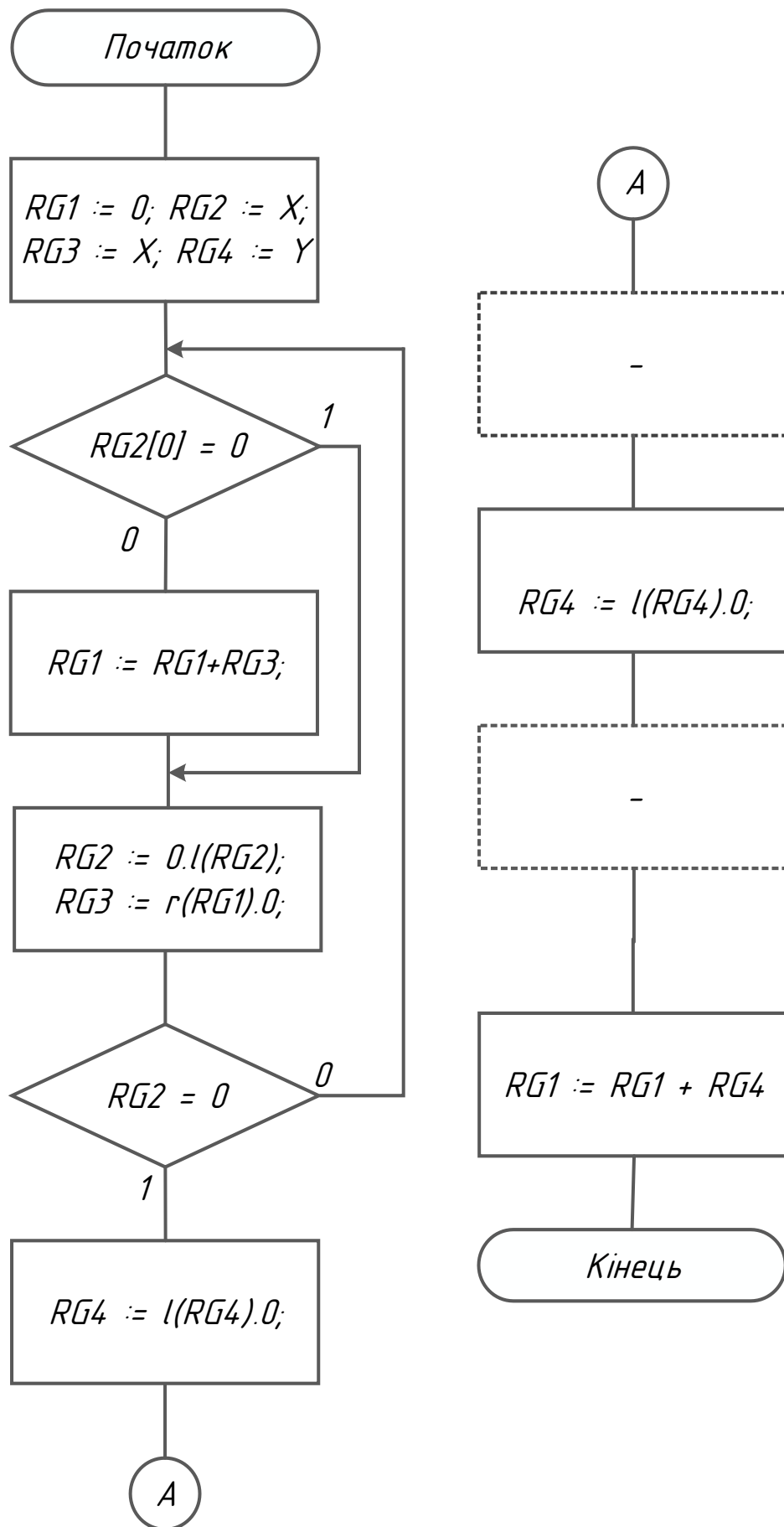


Рис. 5.2. Змістовний мікроалгоритм  
арифметико-логічного пристрою для обчислення функції

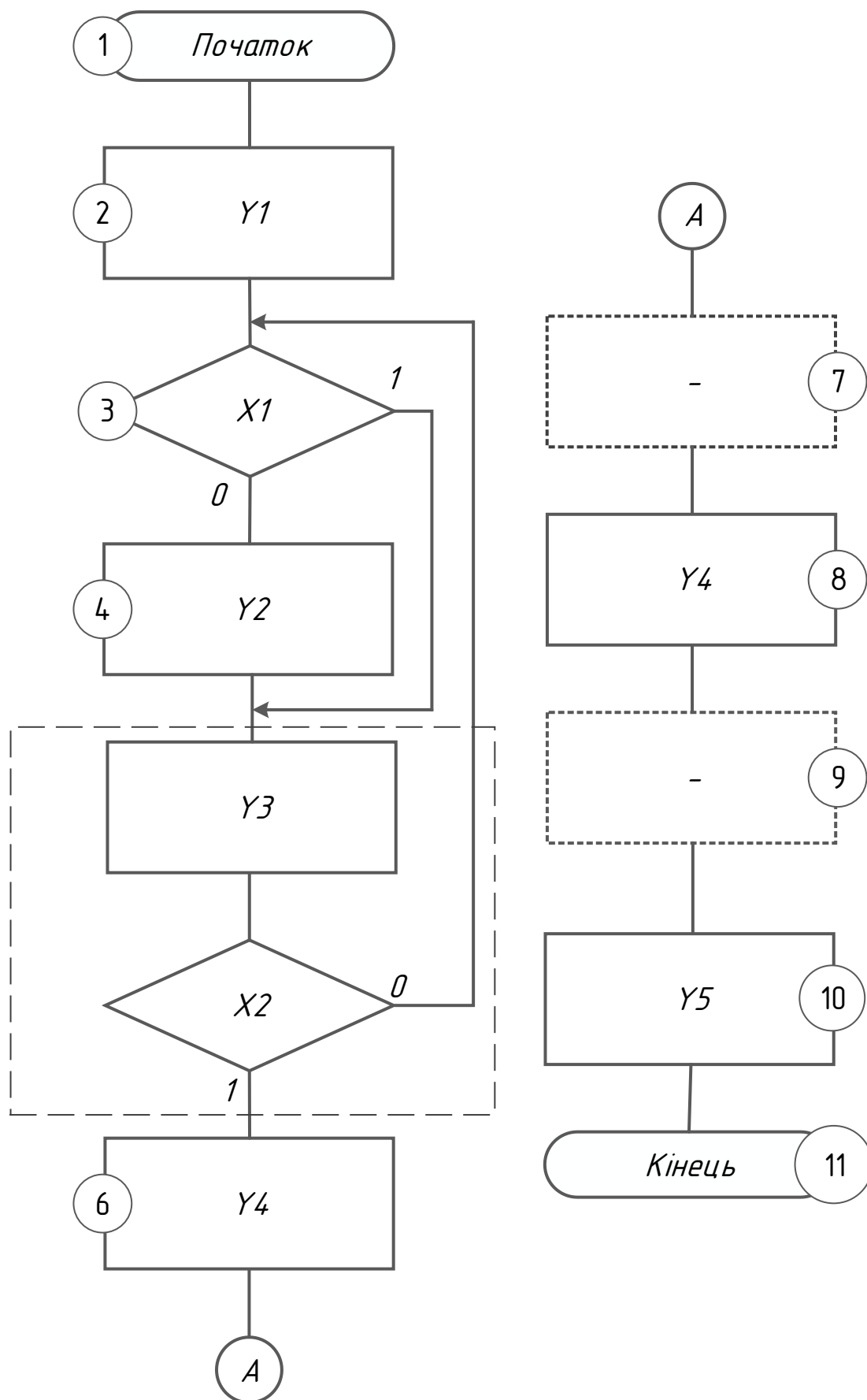


Рис. 5.3. Закодований мікроалгоритм  
арифметико-логічного пристрою для обчислення функції

Таблиця 5.1. Таблиця кодування мікрооперацій

Мікрооперації	Управляючі сигнали
$W_2, W_3, W_4$	Y1
$W_1$	Y2
$SHL_2, SHR_3$	Y3
$SHL_4$	Y4
$SUM_1$	Y5

Таблиця 5.2. Таблиця кодування сигналів

Логічні умови	Позначення
$RG2[0]$	X1
$RG2$	X2

#### 5.4. Розрахунок параметрів БМУ

Складемо таблицю кодування розрядів поля управління мультиплексором (таблиця 5.3). Розміщення команд в ПМК приведено в таблиці 5.4

Таблиця 5.3. Кодування розрядів поля управління мультиплексором.

$m_2m_1$	УС
00	0
01	X1
10	X2
11	1

Формат зони  $\beta 1$  (зони формування адреси наступної МК):

оскільки ємність слів 64, то регістр адреси має бути 6-розрядним:

$$n_a = \lceil \log_2 64 \rceil = 6$$

розмір адресної частини K має бути 2-розрядним:

$$n_k = 6 / 2 - 1 = 2$$

розмір частини управління мультіплексором М, враховуючи кількість умов в алгоритмі, має бути 2-розрядним:

$$n_m = \lceil \log_2(2+2) \rceil = 2$$

Таблиця 5.4. Розміщення команд в ПМК

	000	001	010	011	100	101	110	111
000							2	П(1)
001					5	4	3	6
010								7
011					К(11)	10	9	8

Отримаємо:

$$n_{\beta 1} = 5$$

Формат зони  $\beta 2$  (зона управляючих сигналів):

при горизонтальному (мінімальному) кодуванні управляючих сигналів, довжина зони  $\beta 2$  дорівнює кількості управляючих сигналів:

$$n_{\beta 2} = 6$$

Формат зони  $\beta 3$  (зона визначення тривалості управляючих сигналів):

$$n_{\beta 3} = \lceil \log_2 11 \rceil + 1 = 5$$

Формат зони  $\beta 4$  (зона службових розрядів):

Для контролю використовують операцію згортки (суму за модулем 2). У цьому випадку зона  $\beta 4$  має довжину 1 розряд, вміст цього розряду доповнює кількість 1 у слові мікрокоманди до парної.

$$n_{\beta 4} = 1$$

Враховуючи попередні обчислення отримаємо довжину команди:

$$n_{mk} = 5 + 6 + 5 + 1 = 17$$

Карта програмування БМУ зображена у таблиці 5.5., структурна схема БМУ зображена на рисунку 5.4.

Таблиця 5.5. Карта програмування БМУ

МК	Адреса МК		$\beta 1$			$\beta 2$					$\beta 3$	$\beta 4$
	Ряд.	Ст.	V	K	M	Y1	Y2	Y3	Y4	Y5		
П(1)	000	111	0	11	00	0	0	0	0	0	0000	0
2	000	110	1	00	11	1	0	0	0	0	0000	0
3	001	110	0	10	01	0	0	0	0	0	0000	0
4	001	101	0	10	00	0	1	0	0	0	1011	1
5	001	100	0	11	10	0	0	1	0	0	0000	0
6	001	111	1	01	00	0	0	0	1	0	0000	0
7	010	111	1	01	11	0	0	0	0	0	0000	0
8	011	111	0	11	00	0	0	0	1	0	0000	0
9	011	110	0	10	11	0	0	0	0	0	0000	1
10	011	100	1	10	00	0	0	0	0	1	1011	0
11	100	100	0	10	00	0	0	0	0	0	0000	1

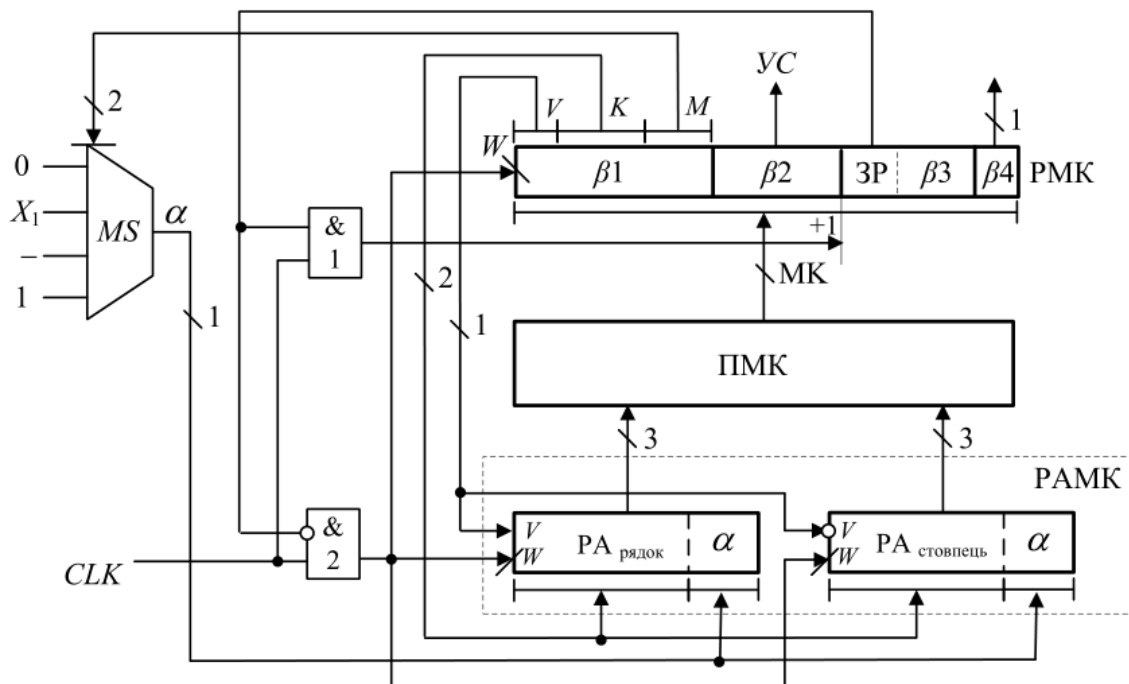


Рис.5.4. Структурна схема БМУ з матричною ПМК

## 5.5. Моделювання з використанням САПР Quartus II

На рис. 5.5. представлено схему БМЧ для керування АЛП, а схему АЛП – на рис.5.6. Загальна схема вузла ЕОМ для обчислення функції  $Z = x^2 + 4Y$  представлена на рис. 5.7.

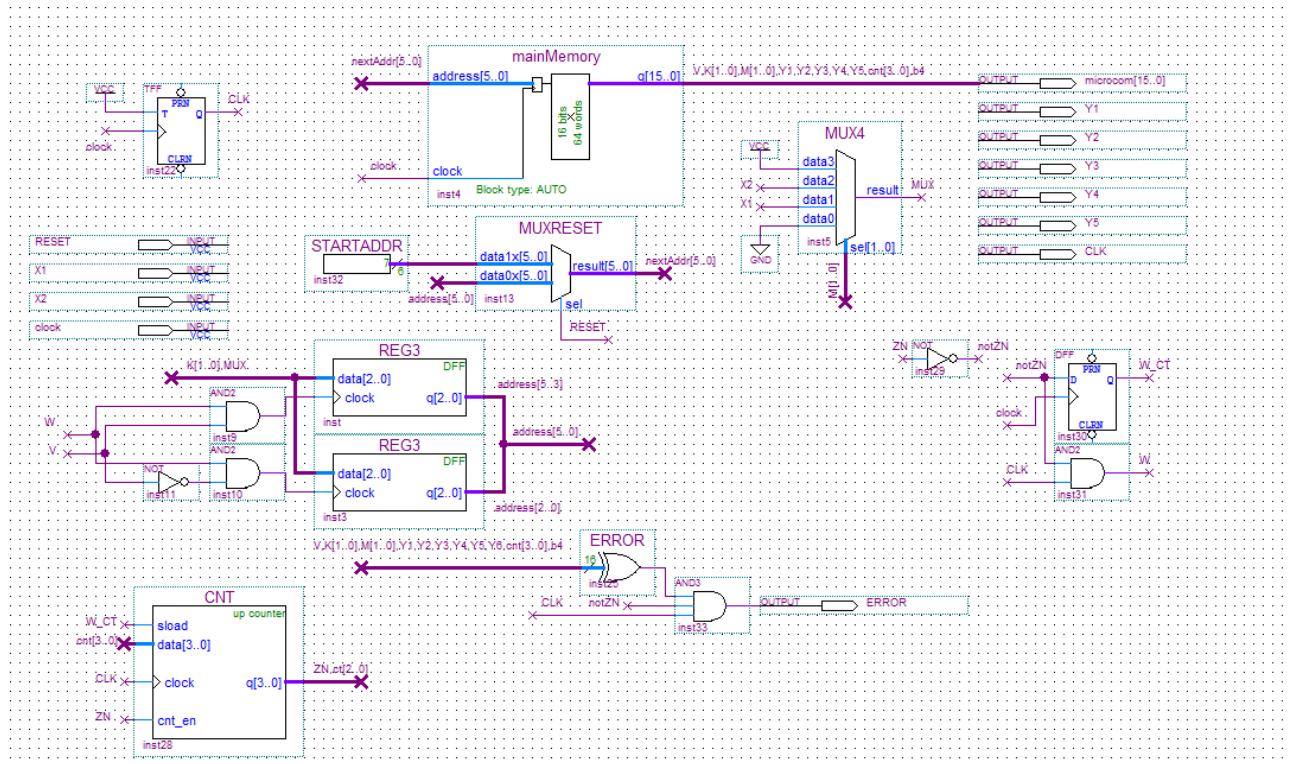
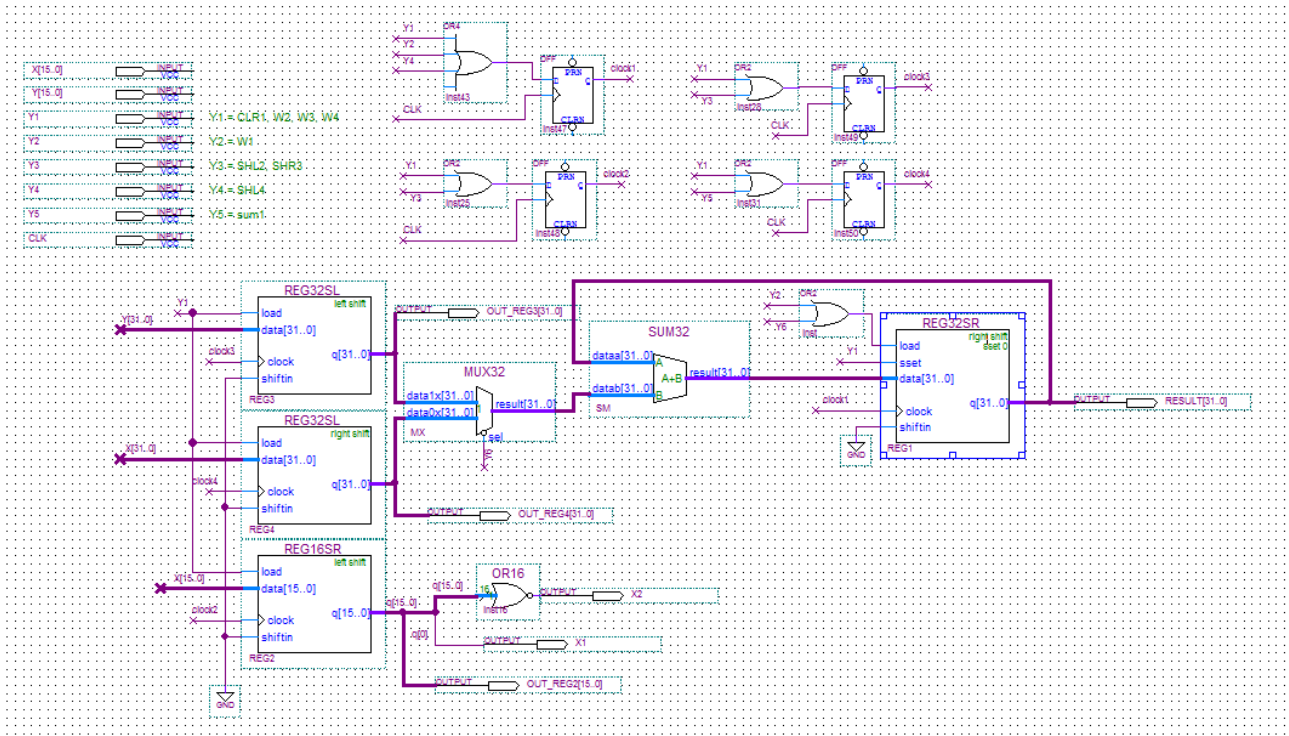


Рис.5.5. Схема БМЧ



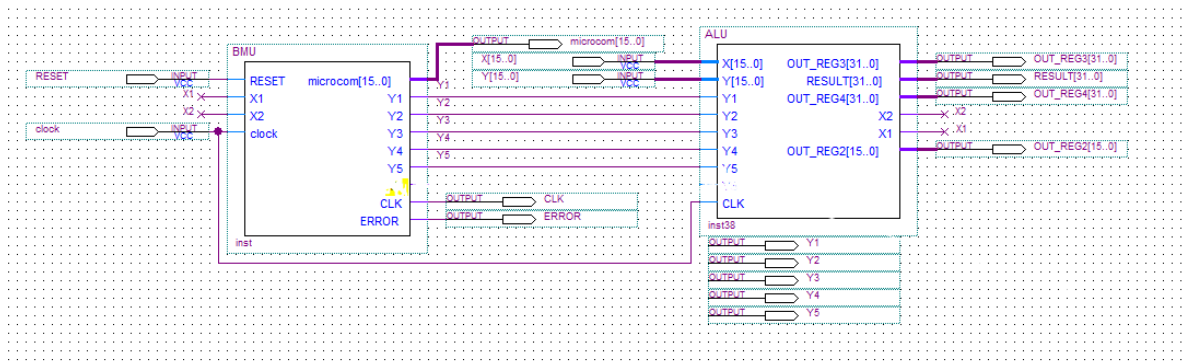


Рис.5.7. Схема ЕОМ

Зведений звіт про результати компіляції представлено на рис.5.7

Flow Status	Successful - Mon Jun 13 13:30:20 2016
Quartus II Version	9.1 Build 222 10/21/2009 SJ Web Edition
Revision Name	coursework2
Top-level Entity Name	MAIN
Family	Cyclone II
Device	EP2C35F672C6
Timing Models	Final
Met timing requirements	No
Total logic elements	206 / 33,216 ( < 1 % )
Total combinational functions	205 / 33,216 ( < 1 % )
Dedicated logic registers	128 / 33,216 ( < 1 % )
Total registers	128
Total pins	170 / 475 ( 36 % )
Total virtual pins	0
Total memory bits	1,024 / 483,840 ( < 1 % )
Embedded Multiplier 9-bit elements	0 / 70 ( 0 % )
Total PLLs	0 / 4 ( 0 % )

Рис.5.8. Compilation report – Flow summary



## ОСНОВНІ РЕЗУЛЬТАТИ І ВИСНОВКИ ПО РОБОТІ

У даній курсовій роботі приведений весь процес розробки мікропроцесорної системи – від складання структурної схеми за технічним завданням до розробки принципової схеми конкретних вузлів і написання програми для обчислення необхідної функції.

На базі мікроконтролера МК48 була розроблена мікропроцесорна система. Вона містить зовнішню пам'ять програм, пам'ять даних, зовнішні пристрої, про-  
грамований периферійний адаптер, КПП, додаткові порти вводу-виводу та спеціалізований обчислювач. Для організації переривань і прямого доступу до пам'яті створені централізовані арбітри пріоритетних переривань і прямого доступу до пам'яті. Перевага даного підходу в його гнучкості, тобто можливості додавання нових пристроїв. Недолік – фіксовані значення пріоритетів пристроїв, неможливість маскування переривань від окремих пристроїв.

Також в ході роботи був розроблений спеціалізований обчислювач на базі програмованої логічної інтегральної схеми, який виконує різні арифметичні й логічні операції. Його використання дозволяє підвищити ефективність роботи розробленої системи за рахунок винесення громіздких обчислень (наприклад ділення) в окремий пристрій.

Схема обчислювача розроблялася в системі автоматизованого проектування Quartus II, яка дозволяє створювати різні пристрої на базі програмованих логічних інтегральних схем. Перевагою даної методики є те, що, по-перше, їх використання дозволяє скоротити час розробки і, по-друге, виробництво малих партій є економічно виправдано, на відміну від виробництва замовлених інтегральних схем, що не можуть бути перепрограмовані.

					ІАЛЦ 462619.004 ПЗ	Арк.
						41
Зм.	Арк.	№ докум.	Підпис	Дат		

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Жабін В.І., Жуков І.А., Ткаченко В.В., Клименко І.А. — Мікропроцесорні системи. — К.: “СПД Гуральник О.Ю.”, 2009. — 492 с.
2. Жабін В.І., Жуков І.А., Клименко І.А., Спіренко С.Г. — Арифметичні та управляючі пристрої цифрових ЕОМ: Навчальний посібник. — К.: БЕК+, 2008. — 176 с.
3. Жабін В.І., Жуков І.А., Клименко І.А., Ткаченко В.В. — Прикладна теорія цифрових автоматів: Навч. посібник. — К.: Книжкове вид-во НАУ, 2007. — 364 с.
4. Intel Corporation — MCS-48™ FAMILY OF SINGLE CHIP MICROCOMPUTERS: USER'S MANUAL - Santa Clara, CA 95051, 1980 — 487 с.

					<i>ІАЛЦ 462619.004 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		42