

Міністерство освіти і науки України  
Центральноукраїнський національний технічний університет

**Минайленко Р.М., Коноплицька-Слободенюк О.К., Гермак В.С.**

# **Архітектура комп'ютера**

*Навчальний посібник  
(видання друге, доповнене)*

Кропивницький

2022

**УДК 004.75**

**ББК 32.973.2**

**М 61**

*Рекомендовано Вченою радою Центральноукраїнського національного технічного університету, протокол № 8 від 26.04.2022 р*

*Рецензенти:*

доктор технічних наук, професор Одарченко Р.С.  
доктор технічних наук, професор Євсєєв С.П.

**Минайленко Р.М., Коноплицька-Слободенюк О.К., Гермак В.С.**

**М 61** Архітектура комп'ютера: навч. посіб. — Кропивницький: Видавець Лисенко В. Ф., 2022. — 153 с.

В посібнику викладено основні поняття та концепції архітектури комп'ютерів.

Крім теоретичного матеріалу в посібнику наведено практичні завдання, вирішення яких відбувається з використанням електронних емуляторів пристроїв, що дозволяє візуально відстежувати процес їх програмування.

Навчальний посібник призначений для студентів, які навчаються за спеціальністю «Комп'ютерна інженерія», а також може бути корисним для спеціалістів в галузі інформатики, обчислювальної техніки та програмування.

**ББК 32.973.2**

© Минайленко Р.М., Коноплицька-Слободенюк О.К.,  
Гермак В.С., 2022

Видавець Лисенко В. Ф., 2022

## ЗМІСТ

Лекція 1.....	5
1.1 Функціональна схема ЕОМ.....	5
1.2 Інтерфейс МП з пам'яттю.....	6
1.3 Інтерфейс МП з ОЗП.....	8
1.4 Взаємодія мікропроцесора з периферійними пристроями при введенні та виведенні інформації.....	11
1.5 Стандартні прийоми програмування, що використовуються для розв'язання інженерних задач.....	14
Контрольні питання та завдання.....	17
Лекція 2.....	18
2.1 Програмований контролер переривань КР580ВН59.....	18
2.2 Основні функції контролера.....	18
2.3 Блок схема програмованого контролера переривань.....	20
2.4 Принцип роботи контролера.....	23
2.5 Програмування контролера.....	23
2.6 Робочі команди.....	25
Контрольні питання та завдання.....	31
Лекція 3.....	32
3.1 Програмований паралельний інтерфейс введення-виведення К580ВВ55.....	32
3.2 Принцип роботи ВІС ВВ55.....	33
3.3 Режими роботи К580ВВ55.....	35
3.4 Приклади програмування.....	44
Контрольні питання та завдання.....	45
Лекція 4.....	46
4.1 Контролер прямого доступу до пам'яті.....	46
4.2 Принципи роботи контролера ПДП.....	46
4.3 Блок схема КПДП.....	49
4.4 Опис внутрішніх регістрів КПДП.....	51
4.5 Програмне керування контролером ПДП.....	54
4.6 Програмна модель КПДП. Програмування КПДП.....	57
Контрольні питання та завдання.....	65
Лекція 5.....	66
5.1 Програмований інтервальний таймер КР580ВІ53. Принцип роботи таймера.....	66
5.2 Блок схема таймера.....	66
5.3 Режими роботи таймера.....	69
5.4 Програмування таймера.....	76
Контрольні питання та завдання.....	83
Лекція 6.....	84
Програмований адаптер послідовного інтерфейсу.....	84
6.1 Послідовні інтерфейси.....	84
6.2 Програмування послідовного адаптера.....	88
Контрольні питання та завдання.....	101

Лекція7.....	102
Контролер прямого доступу до пам'яті.....	
7.1 Принцип роботи контролера ПДП.....	102
7.2 Блок схема КПДП.....	102
7.3 Внутрішні реєстри КПДП.....	105
7.4 Програмне керування контролером ПДП.....	108
7.5 Програмна модель КПДП. Програмування КПДП.....	111
Контрольні питання та завдання.....	120
Література.....	121

## Лекція 1

### 1.1 Функціональна схема ЕОМ

Для системного програміста має значення сукупність програмно – доступних засобів, тобто програмна модель різних пристроїв ЕОМ. У цьому випадку здобуває важливе значення яку інформацію необхідно посилати у відповідні пристрої для вибору режиму, керування і контролю стану даних пристроїв. В даний час широке поширення одержали машини в основу яких покладений принцип відкритості, який полягає в тому, що на системній платі розташовується система дротів (шин), що об'єднані в групи: шина адреси (ША), шина даних (ШД), шина керування (ШК), на які через різні роз'єми (слоти) можливе підключення різних пристроїв. Таким чином, принцип відкритості дозволяє будувати системи зі змінною структурою, а також робити нарощування системи.

Історичний розвиток ЕОМ відбувався в цьому напрямку тому, що основним недоліком відкритої архітектури є наявність системної шини, яка дозволяє працювати процесору тільки з одним пристроєм. Інші пристрої в цей час не можуть користуватися системною шиною. Функціональну схему ЕОМ показано на рис.1:

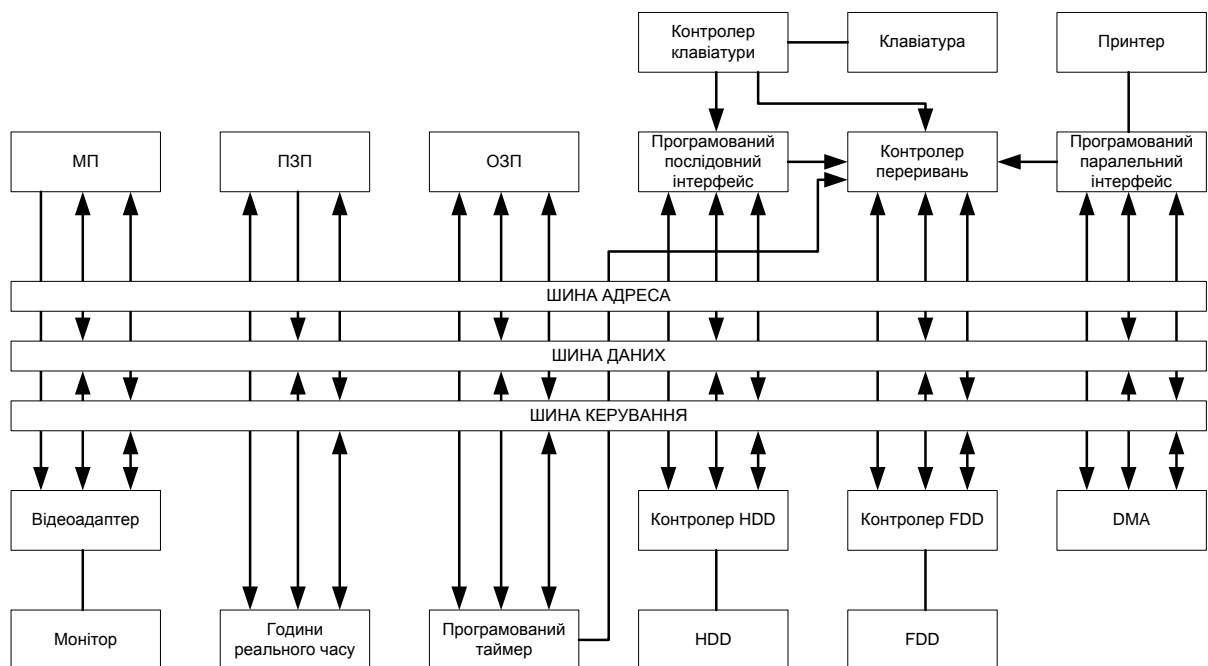


Рисунок 1 - Функціональна схема ЕОМ

### Програмна архітектура ЕОМ

Програмна архітектура ЕОМ дозволяє визначити порядок і правила за якими програміст може здійснювати вплив на різні пристрої, які входять в склад ПЕОМ, а також через комунаційні порти та периферійні пристрої.

Тому важливо знати, які реєстри входять в склад пристроїв ЕОМ і як програміст може вводити або виводити дані в реєстри, читати зміст реєстрів для визначення стану пристроїв, вибрати необхідні режими роботи і здійснювати керування їх роботою.

Головні реєстри які входять в склад пристроїв:

- реєстри стану;
- реєстри керування;
- реєстри режимів.

Чим більше реєстрів входить в склад пристрою тим легше програмісту здійснювати контроль та керування їх роботою.

## **Інтерфейс мікропроцесора**

Під інтерфейсом розуміється сукупність уніфікованих технічних та програмних засобів, необхідних для підключення пристроїв до системи або однієї системи до іншої.

Передача даних від і до МП здійснюється в наступних формах:

- 1) вилучення із пам'яті;
- 2) запис в пам'ять;
- 3) вилучення із ПВВ;
- 4) запис в ПВВ;
- 5) керування перериванням або скиданням

### **1.2 Інтерфейс МП з пам'яттю**

#### **Інтерфейс МП з ПЗП**

**ПЗП** – постійний запам'ятовуючий пристрій, що дозволяє тільки зчитування інформації. Інформація в ПЗП заноситься заздалегідь і зберігається там незалежно від стану комп'ютера (ввімкнений чи вимкнений).

Розглянемо задачу розробки інтерфейсу з ПЗП. На рисунку 2 представлена частина системи включення МП з ПЗП. З виходами ПЗП з'єднанні 8 ліній шини даних. Вхід керування зчитуванням RD надходить з МП на вхід активізації ПЗП /OE.

З постійним запам'ятовуючим пристроєм ємністю 4К з'єднані 12 ліній адресної шини молодших розрядів (A<sub>0</sub>-A<sub>11</sub>). Дешифратор, вбудований в ІС ПЗП, може отримати доступ до будь-якого з 4096 ( $2^{12}=4096$ ) 8 розрядних слів ПЗП. Адресні лінії чотирьох старших розрядів (A<sup>12</sup>-A<sup>15</sup>) потрапляють до дешифратора адреси, при цьому вибирається одна з 16 ( $2^4$ ) можливих мікросхем ПЗП.

Для доступу до ПЗП та зчитування даних МП повинен:

- активізувати лінії адреси A<sub>0</sub>–A<sub>11</sub>;
- встановити сигнал L на лінії керування зчитуванням /OE;
- встановити сигнал L на лінії дешифратора адреси та вибору мікросхеми (HIGH і LOW - рівні сигналу).

МП через системну шину читає інформацію з ПЗП в такий спосіб: якщо програміст вказує в програмі LDA adc1, то мікропроцесор виставляє на ША зазначену адресу і через якийсь час на шину керування видається сигнал RD (низького рівня) у зв'язку з тим, що МП I8080 має у своєму складі 12-ти розрядні регістри, а шина адрес мала 16 ліній ( $A_0 \dots A_{15}$ ), то зі старших ліній  $A_{12} \dots A_{15}$  за допомогою дешифратора адреси був сформований сигнал CS (чіпселект) для вибору одного з необхідних чіпів пам'яті ємністю 2 (4 КХ8) кожний. Це дало можливість за допомогою коротких адрес працювати з пам'яттю (64к).

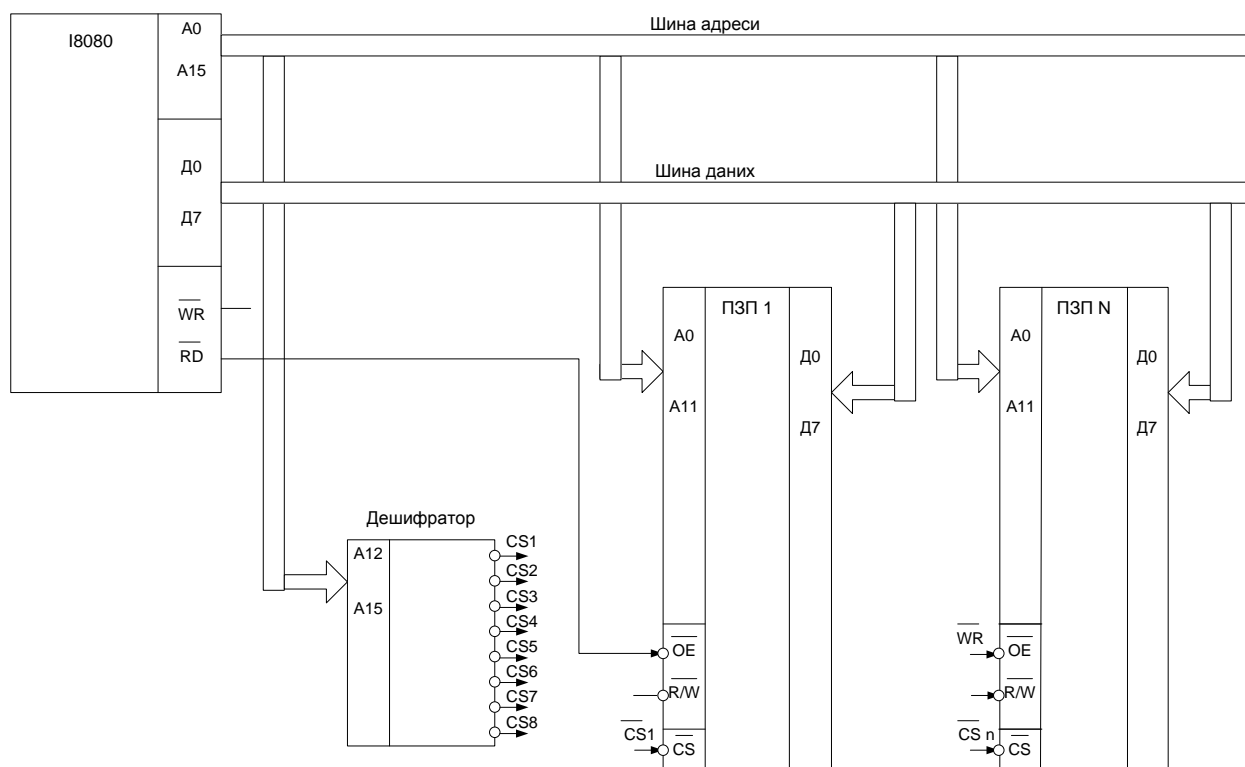


Рисунок 2 – Інтерфейс ПЗП з МП

В інтерфейсі з ПЗП важливе значення має синхронізація. На рисунку 3 приведена часова діаграма сигналів МП, керуючих зчитуванням 8-розрядного слова з ПЗП. Верхня лінія діаграми показує перехід адресних ліній  $A_0 \dots A_{15}$  на відповідний логічний рівень. Через деякий час вихід керування зчитуванням /RD МП активізує процес виводу даних з ПЗП. Розміщені тут дані розміщуються на шині даних та приймаються МП. Пунктир відповідає стану високого опору.

### Сигнали МП у ході операцій зчитування з ПЗП Часова діаграма

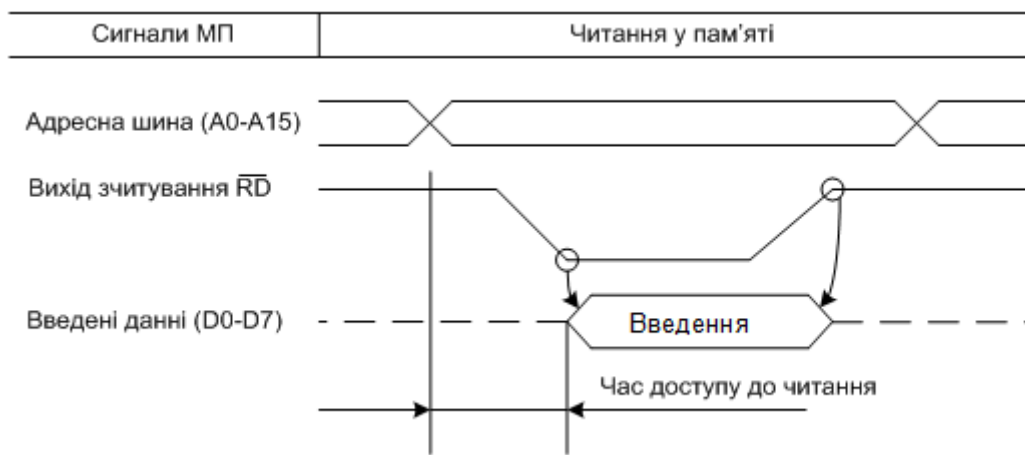


Рисунок 3 - Часова діаграма сигналів МП та ПЗП

### 1.3 Інтерфейс МП з ОЗП

**Інтерфейс МП з ОЗП** (оперативний запам'ятовуючий пристрій) - пристрій розміщення даних, що допускає їхній запис і зчитування. Інформація в ОЗП зберігається тільки при включеному комп'ютері, при вимиканні – інформація губиться. Існують два типи ОЗП: статичні та динамічні.

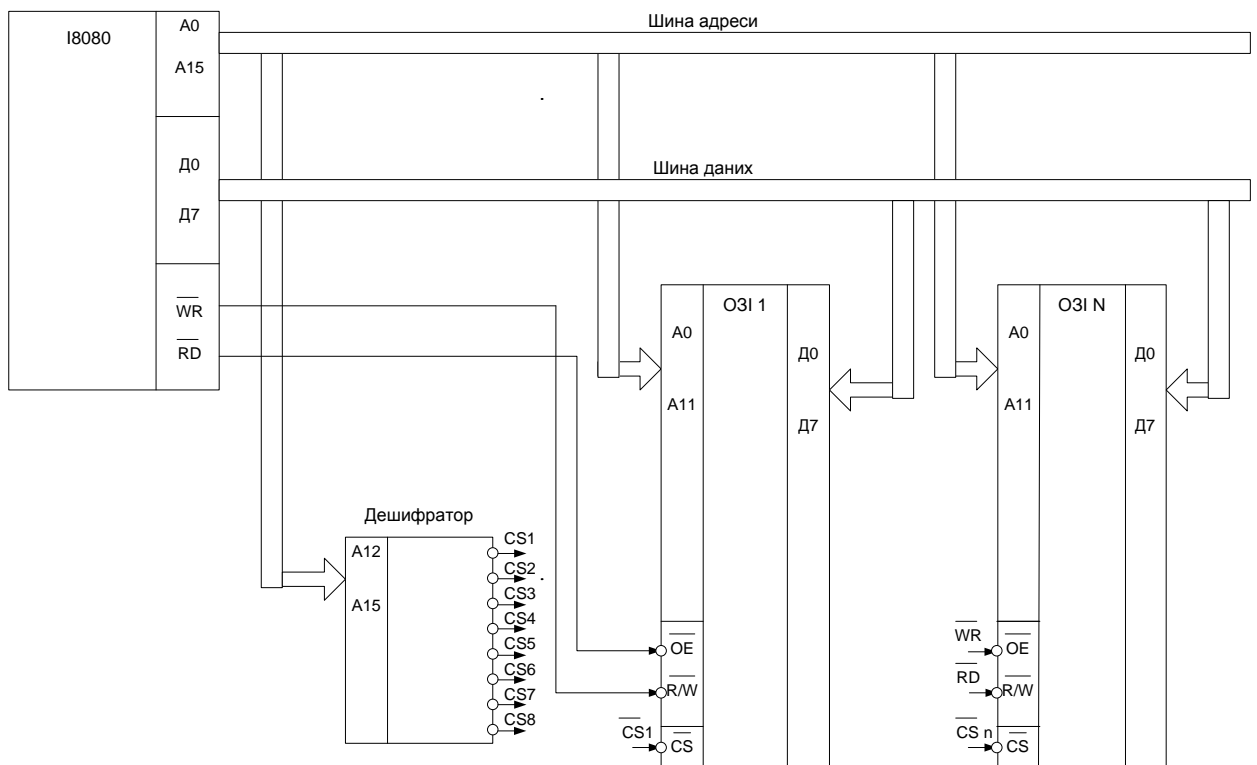


Рисунок 4 - Інтерфейс МП з ОЗП

Інтерфейс МП з ОЗП побудований аналогічно інтерфейсові МП із ПЗП. Єдиною відмінністю є те, що в цьому випадку відбувається як запис, так і

зчитування інформації з обраного модуля MS, що призводить до введення додаткового сигналу WR, який надходить на вхід модуля R/W, і в залежності від активізації сигналів RD або WR відбувається запис або зчитування інформації.

### Сигнали МП і ОЗП в ході операції зчитування з пам'яті Часова діаграма

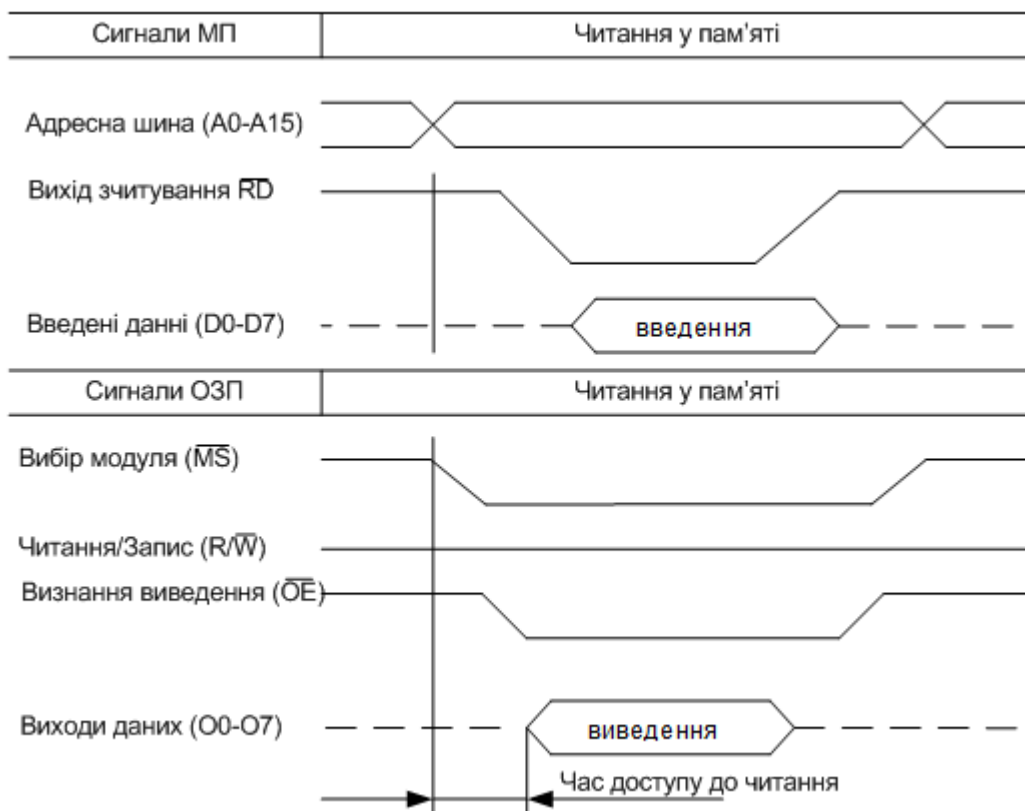


Рисунок 5 - Часова діаграма зчитування даних з ОЗП

Шина даних є двонаправленою 8-розрядною для зчитування або запису даних в ОЗП. Дванадцять молодших розрядів адреси надходять на адресні входи модулю ОЗП. Чотири адресні лінії старших розрядів надходять до дешифратора адреси на якому формується сигнал вибору одного з модулів ОЗП.

Часова діаграма ілюструє зміну сигналів МП та ОЗП під час зчитування з ОЗП. Адресні лінії мікропроцесора активовані і мають потрібну адресу. При переході сигналу /RD до L-рівня, тристабільна шина даних переходить в стан зчитування. Таким чином МП готовий прийняти дані з ОЗП. Сигнали /MS (/CS) вибору модулю ОЗП та /OE дозволу виходу переходять до L-рівня. Через деякий час активується шина даних і дані ОЗП можуть прийнятися МП. Як і ПЗП, час дозволу доступу до пам'яті є важливим показником ОЗП і залежить від типу (модифікації) мікросхеми ОЗП.

Інтерфейс ОЗП і МП при записі в пам'ять часові діаграми відрізняються тим, що замість сигналу Read активізується сигнал Write і в модулі ОЗП.

Послідовність операцій починається з встановлення адреси та сигналу вибору модуля ОЗП. Після деякого інтервалу часу, що називається часом адресації, сигнал /WR МП активує вхід /RD-/WR ОЗП встановлюючи його в стан запису. Сигнал запису повинен бути встановлений в найкоротший час, який називається часом імпульсу запису. Після встановлення необхідних сигналів дані МП розміщуються на шині даних і через деякий час записуються в комірку пам'яті ОЗП. Час циклу запису, час адресації і довжина імпульсу запису залежить від модифікації мікросхеми ОЗП.

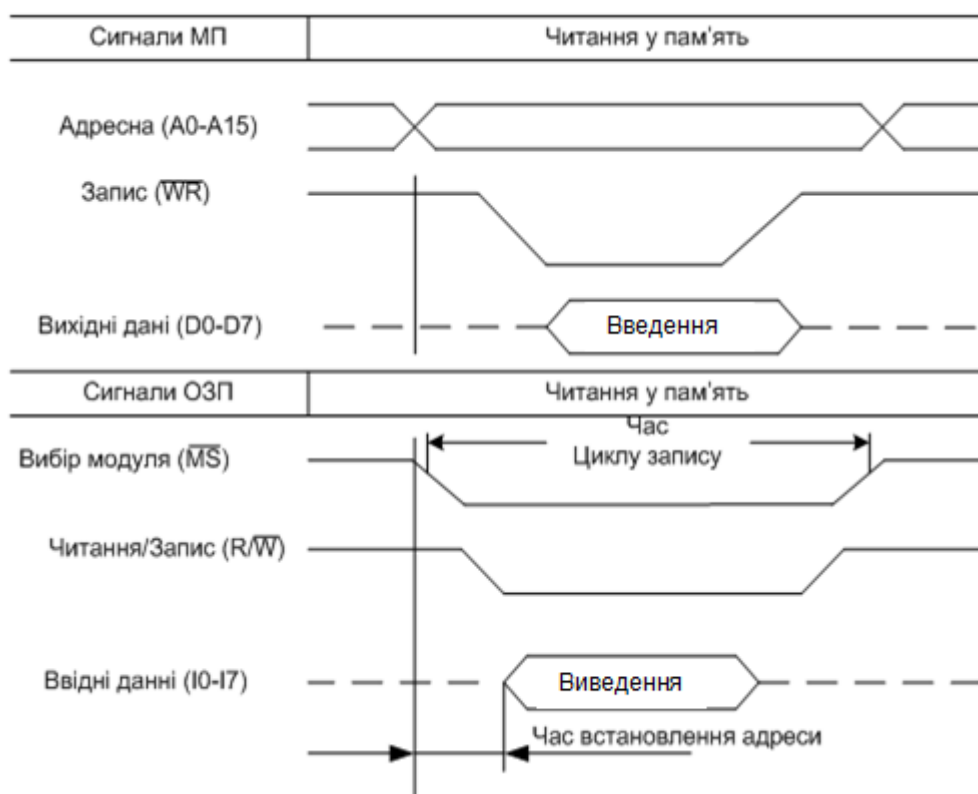


Рисунок 6 - Часова діаграма запису даних до ОЗП

Інтерфейс МП з ПЗП і ОЗП показано на рисунку 7:

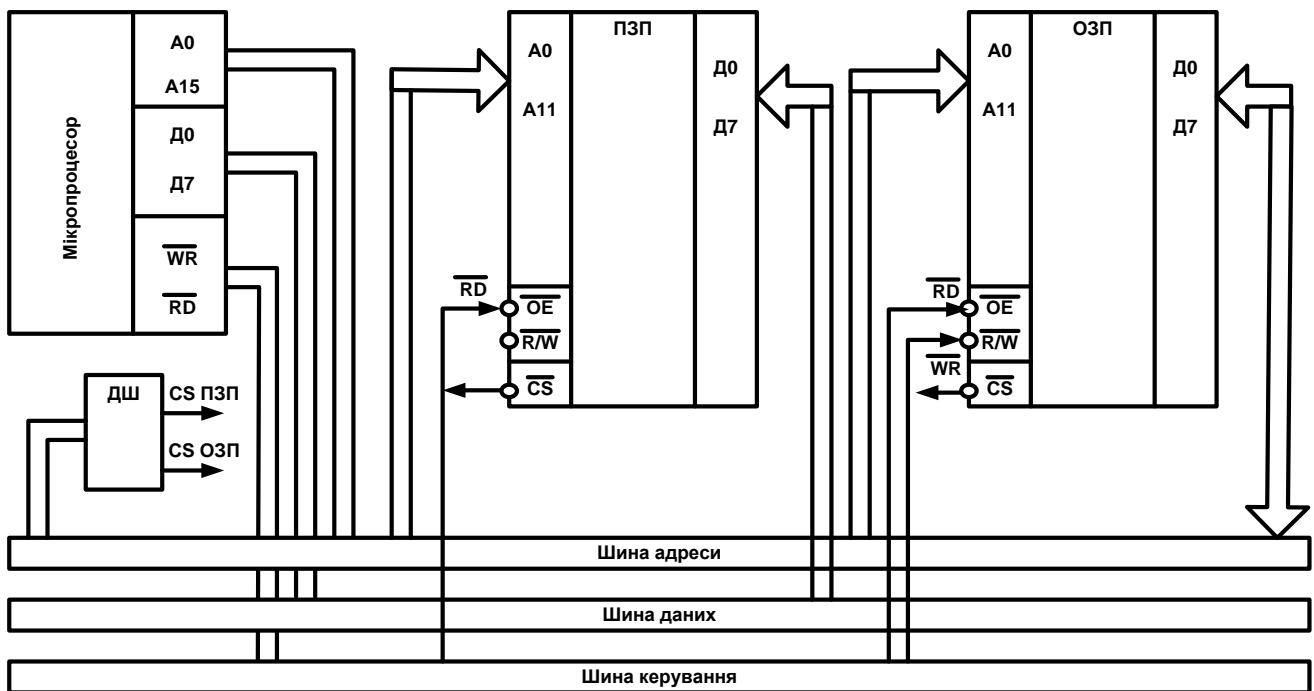


Рисунок 7 - Інтерфейс МП з ПЗП і ОЗП

#### 1.4 Взаємодія мікропроцесора з периферійними пристроями при введенні та виведенні інформації

Введення-виведення (ВВ) МП включає в себе потрібні периферійні пристрої. Існують 2 методи взаємодії з периферійними пристроями:

- засіб взаємодії МП з периферійними пристроями як з комірками пам'яті ЕОМ;
- засіб взаємодії МП з периферійними пристроями адреси яких ізольовані від пам'яті ЕОМ.

В ЕОМ застосовуються три режими ВВ:

- програмно-керований ВВ (називаний також програмним або нефорсованим ВВ),
- ВВ по перериваннях (форсований ВВ);
- прямий доступ до пам'яті.

Операція вводу або виводу включає в себе передачу даних із або в потрібну периферію. Пристрої, за допомогою яких МП спілкується з зовнішніми об'єктами називаються портами вводу/виводу. На рис. 8 відображено два способи ВВ даних в порти.

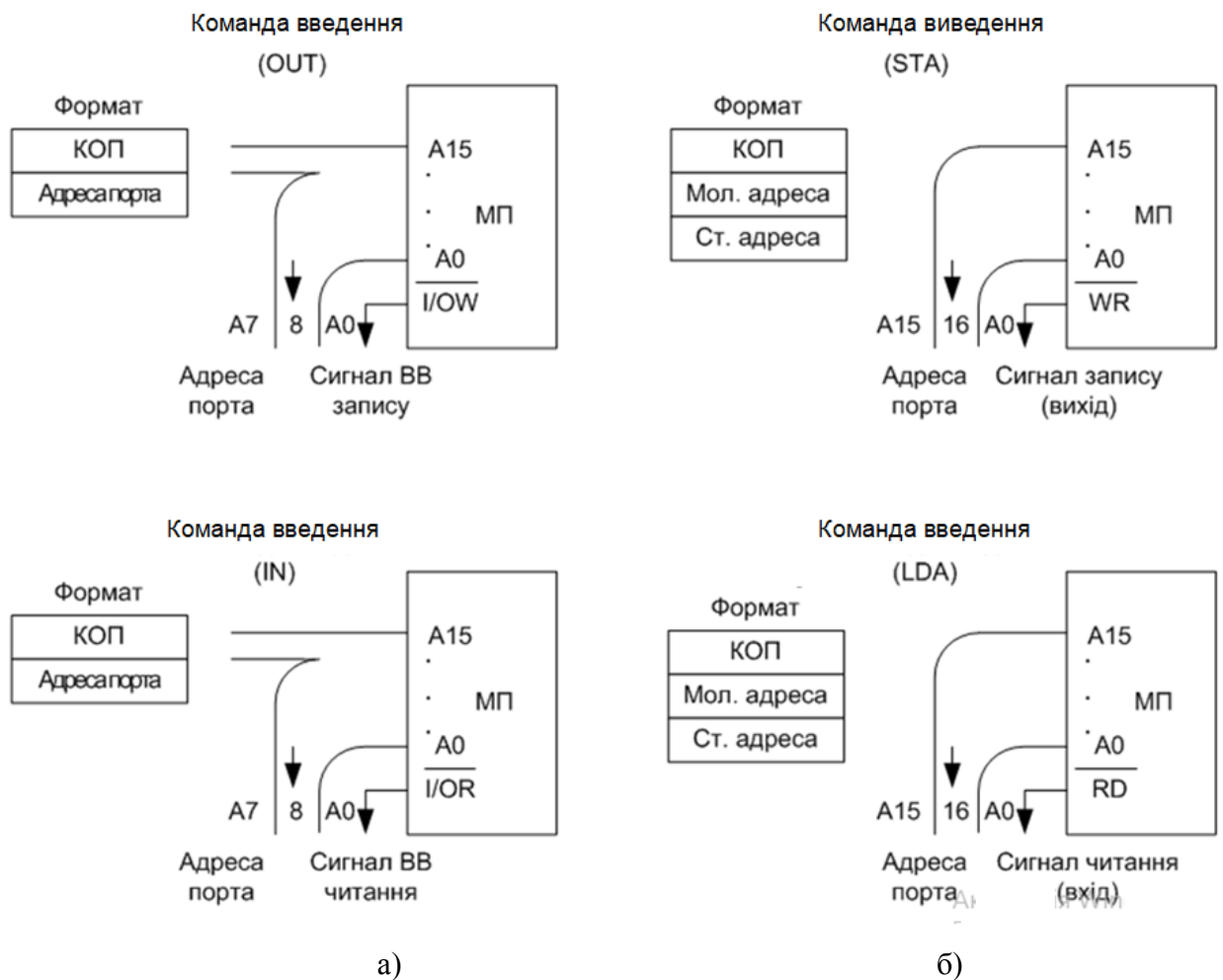


Рисунок 8 - ВВ даних та сигнали МП  
 а) ізольований ВВ; б) ВВ по принципу доступу до пам'яті

При реалізації ВВ за допомогою команд IN, OUT адреса порту передається за допомогою 8-ми адресних ліній молодших розрядів: 1байт=256 портів. При операції OUT використовується особливий сигнал виводу - запис (I/OW). В разі операції IN використовується сигнал вводу читання (I/OR). Це дає можливість використовувати загальні адреси при організації ВВ за допомогою команд IN, OUT. При реалізації ВВ по принципу безпосереднього доступу до пам'яті використовуються команди LDA і STA. По команді STA відбувається виведення змісту акумулятора А в порт виведення, який сприймається процесором як комірка пам'яті. При цьому застосовується керуючий сигнал МП /WR. По команді LDA відбувається введення змісту порту введення (комірки пам'яті в А), при цьому застосовується керуючий сигнал МП /RD.

### Переваги і недоліки ВВ який адресується як пам'ять:

1) Будь яка команда, яка працює з даними, які знаходяться в пам'яті, може працювати з даними, які знаходяться в пристрої вводу або виводу. Ніяких особливих команд вводу-виводу не потрібно і програмування багатьох задач спрощується.

2) Не потрібно окремої системи для декодування ВВ.

3) В систему легко включити інтерфейс ВІС і спеціальні контролери. Такі пристрої часто містять регістри, які настраюються процесором. Ці обставини є суттєвими, так як керування в таких умовах за допомогою портів ВВ неможливо здійснити без ускладнення апаратного та програмного забезпечення.

### Переваги ізолюваного ВВ:

1) Адреса порту вводу-виводу може бути короткою. В більшості систем для адресації порту ВВ достатньо 8 розрядів, що спрощує декодуючі системи і дає можливість використовувати короткі команди.

2) Можливо легко розробити додаткові сигнали передачі інформації при ВВ (стробування, запуск).

3) Програми стають більш наглядними, так як операції ВВ відбуваються за допомогою команд IN, OUT, які відрізняються від інших.

4) Розробка систем ВВ може відбуватись окремо від розробки моделей пам'яті.

5) Дає можливість використання при малому ступеню інтеграції ВІС.

### Інтерфейс МП при ВВ інформації

На рис. 9 відображений інтерфейс МП з периферією.

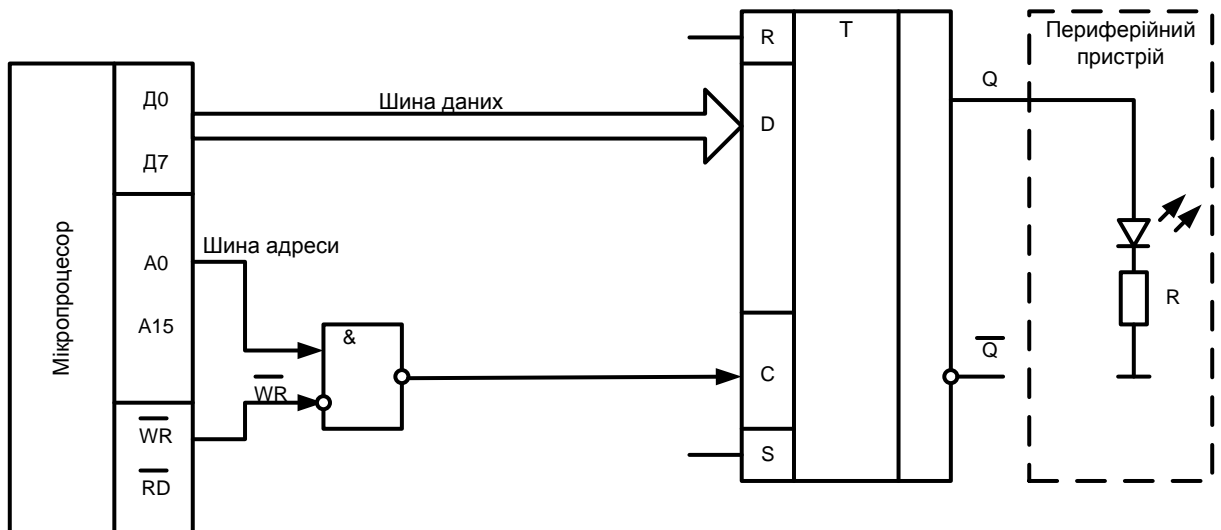


Рисунок 9 - Інтерфейс МП при виводі інформації.

Припустимо, що МП виводить дані з (A)=00000001. Якщо встановити адресу 8000H, то лінія A15 буде знаходитись в Н-активному стані і активізує нижній вхід елемента &. З появою сигналу МП /WR активізується елемент & і підтверджує 1 на D-вході тригера. На виході тригера з'являється Н-сигнал (Q=1) і запалює світлодіод.

Інтерфейс МП при введенні одиничної інформації показано на рисунку 10:

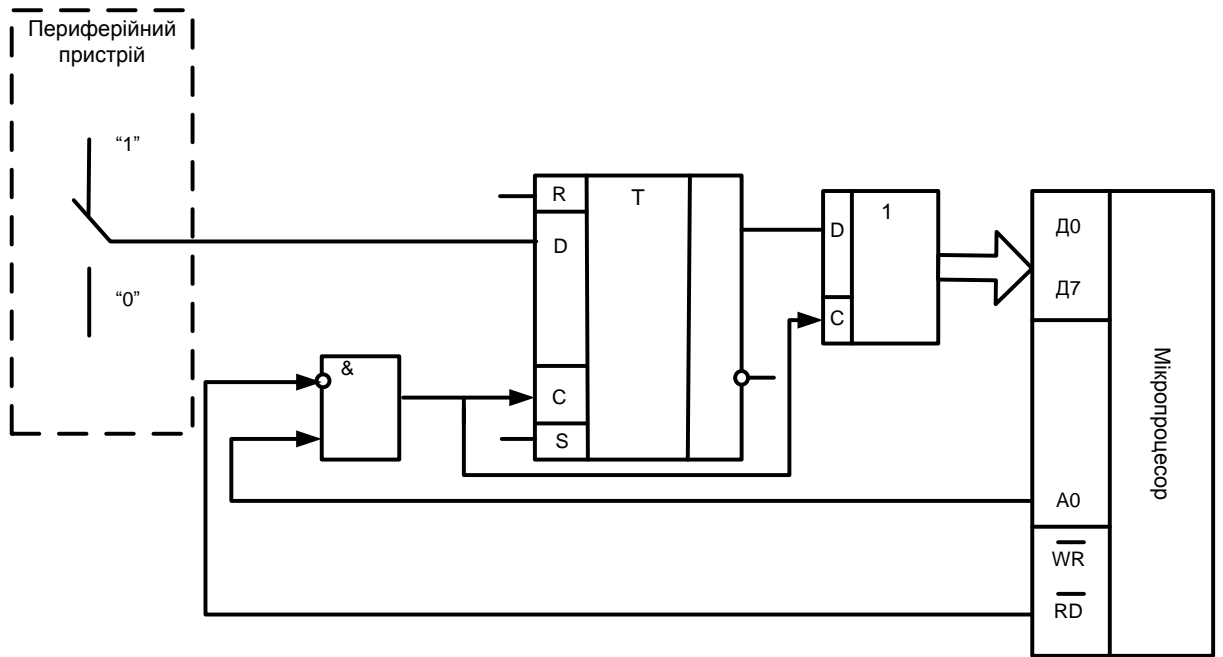


Рисунок 10 - Інтерфейс МП при введенні одичинної інформації.

### 1.5 Стандартні прийоми програмування, що використовуються для розв'язання інженерних задач

Однією з таких задач є визначення стану тумблерів системи при наступних умовах:

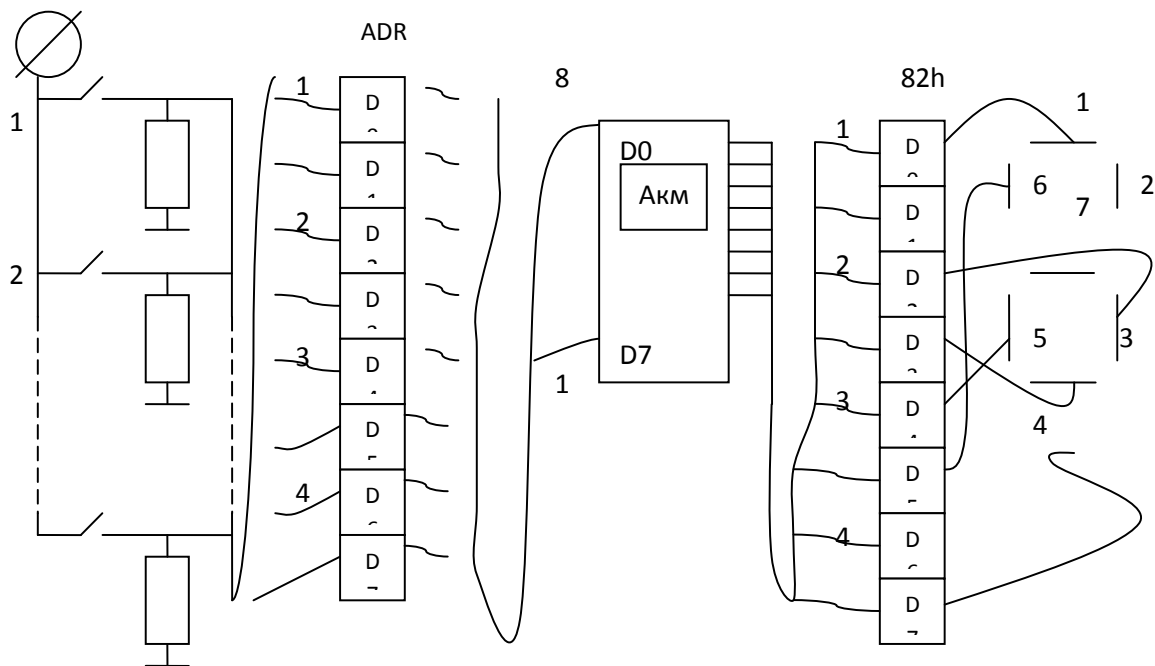


Рисунок 11 - Схема підключення тумблерів

- кількість тумблерів 8
- адреса регістру до якого підключені тумблери 900h
- адреса індикатора 901h

Необхідно визначити кількість ввімкнених тумблерів та їх положення. Спочатку складемо структурну схему та алгоритм програми визначення кількості ввімкнених тумблерів (рис.12):

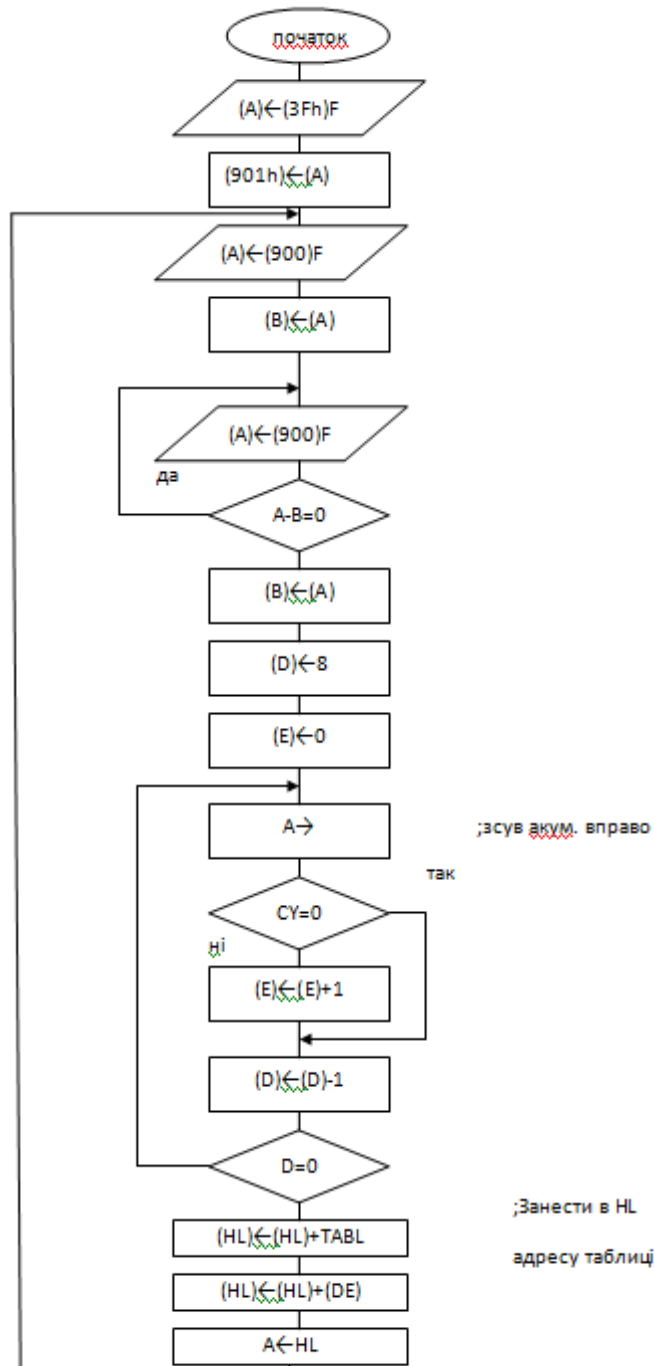


Рисунок 12 - Алгоритм програми визначення кількості ввімкнених тумблерів

Створимо таблицю кодів цифр, які треба виводити на індикатор.

Цифра	D7	D6	D5	D4	D3	D2	D1	D0	
1	0	0	0	0	0	1	1	0	06h
2	0	1	0	1	1	0	1	1	5bh
3	0	1	0	0	1	1	1	1	4fh
4	0	1	1	0	0	1	1	0	66h
5	0	1	1	0	1	1	0	1	6dh
6	0	1	1	1	1	1	0	1	7dh
7	0	0	0	0	0	1	1	1	07h
8	0	1	1	1	1	1	1	1	7fh
9	0	1	1	0	1	1	1	1	6fh
0	0	0	1	1	1	1	1	1	3fh
-	0	1	0	0	0	0	0	0	40h

-----  
; Програма визначення кількості включених об'єктів  
-----

--

```

    ORG 800h
    MVI A, 03Fh
    STA 901h
M1: LDA 900h
    MOV B, A
    LDA 900h
    CMP B
    JZ M1
    MOV B, A
    MVI D, 08h
    MVI E, 00h
M2: RAR
    JNC M3           ; якщо немає переносу
    INR E
M3: DCR 0
    JNZ M2
    LXI H, TABL
    DAD D           ; додаємо регістри
    MOV A, M
    STA 901h
    JMP M1
TABL: DB 06h, 01Dh, 04Fh, 066h, 06Dh,
        07Ch, 053h, 07Fh, 067h.
```

## **Контрольні питання та завдання**

1. Введення-виведення інформації. Інтерфейс портів вводу-виводу.
2. Інтерфейс МП з ПЗП та ОЗП. Функціональна схема системи МП та пам'яті. Сигнали та часові діаграми роботи.
3. На лабораторній роботі скласти програму по визначенню кількості та положення ввімкнутих тумблерів.

## Лекція 2

### 2.1 Програмований контролер переривань KP580BH59 Призначення і структура

В мікропроцесорній системі необхідно щоб пристрої ВВ такі як клавіатура, дисплей, датчики та інші компоненти мали ефективне обслуговування.

Найбільш поширений метод обслуговування таких пристроїв є опитування. ЦП послідовно перевіряє кожний пристрій, чи потребує він обслуговування. Таким чином більша частина основної програми буде складатися з безперервного циклу опитування, що негативно впливатиме на пропускну спроможність системи.

Більш корисним буде метод, який дозволяє мікропроцесору виконувати головну програму і зупинятися тільки на обслуговування периферійних пристроїв по запиті. Цей метод дозволяє проінформувати процесор про необхідність завершити операції які виконуються в цей час і отримати запит на обслуговування периферійного пристрою.

Після процедури обслуговування пристрою процесор повернеться до виконання програми в тому місці де вона була зупинена. Це називається перериванням. При цьому пропускну спроможність системи набагато збільшиться.

Програмований контролер переривань отримує запити від периферійних пристроїв, визначає які з вхідних запитів мають пріоритет вищого значення, ніж той що обслуговується в даний час .

### 2. 2 Основні функції контролера

Оскільки I8080 обслуговує тільки один запит на переривання, що надходить на вхід INT, то очевидно, що обслуговувати більшу кількість переривань без апаратної підтримки неможливо. Функції такої підтримки і покликаний виконувати контролер переривань. Контролер переривань обслуговує вісім запитів на переривання і має вбудовані апаратні засоби для збільшення кількості запитів до 64 ( $64=8*8$ ), шляхом каскадування, з іншими аналогічними контролерами. Порядок обслуговування переривань задається програмно. Кожен запит на переривання може бути заборонено/дозволено програмою маскування.

Програмований контролер переривань (ПКП, Programmable Interrupt Controller, PIC) реалізує векторну систему переривань.

Мікросхема I8259A фірми Intel, а так само її модифікації I8259A-2 і I8259A-8, підтримує 8 рівнів переривань від восьми різних пристроїв.

Блок схема контролера приведена на рис. 13.

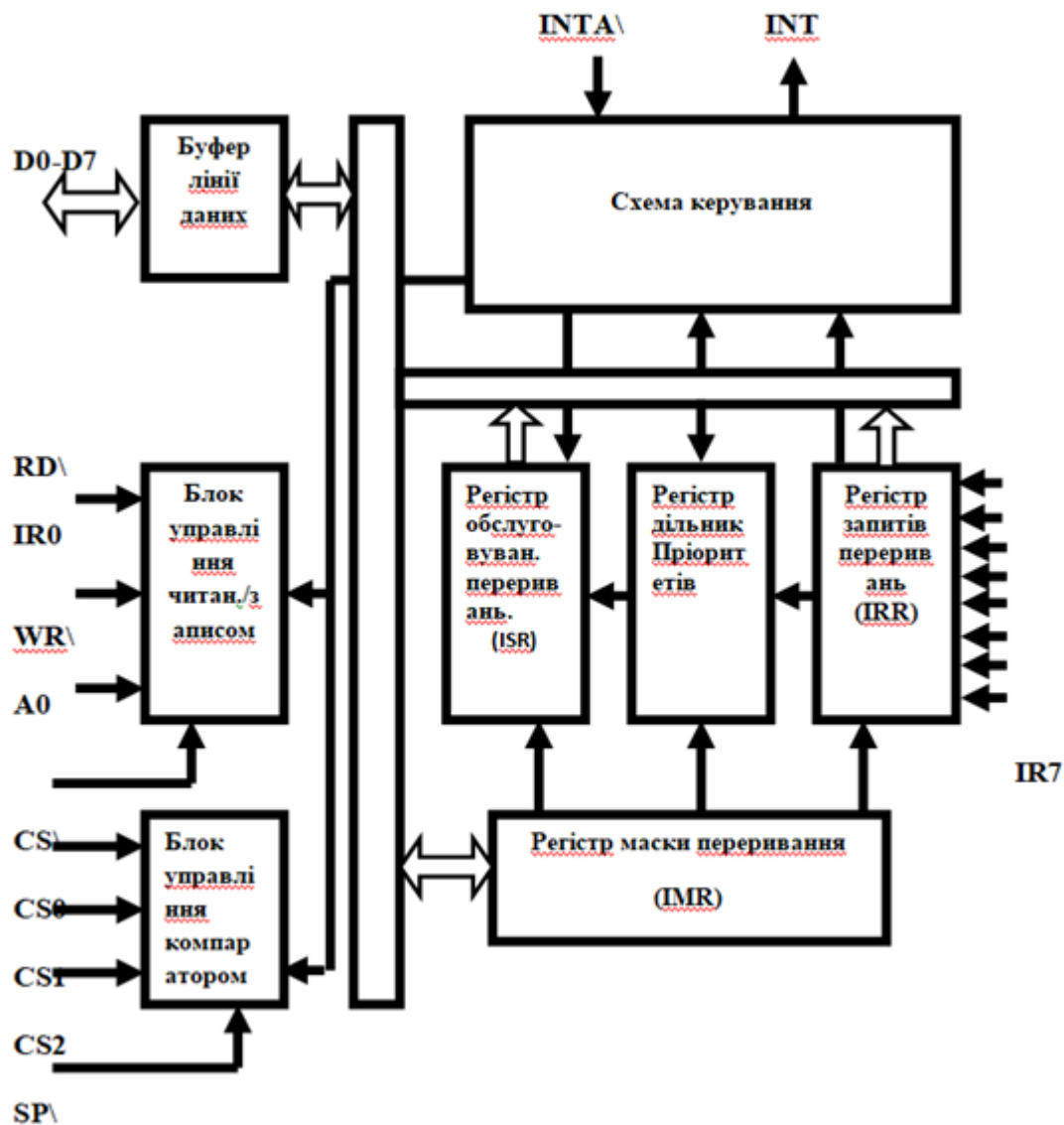


Рисунок 13 - Блок схема ПКП, де:

**D0.D7** - лінія даних для взаємодії з ЦП. В великих системах можливо будуть необхідні шини драйвери, а в малих системах достатньо прямих з'єднань.

**RD/** - системний сигнал читання, дозволяє розмістити на шині даних зміст регістрів IMR, ISR або IMR або пріоритетний рівень, який видається на шині і залежить від стану контролера.

**WR/** - системний сигнал запису, який дозволяє прийняти дані з шини даних і використати їх для встановлення бітів в словах наказів.

**A0** - вхід вибору адреси; цей вхідний сигнал використовується сумісно з WR та RD для запису команд в різні командні регістри, а також для зчитування різних регістрів стану мікросхеми. Ця лінія може підключатися безпосередньо до однієї з адресних ліній.

**CS/** - чіпселект; вибір корпусу ВІС

**CAS0...CAS2** - лінії виходів каскадування, для ідентифікації конкретного підлеглого контролера.

**SP** - Виконує дві функції. Як вхід, визначає, яким є контролер: ведучим ( $SP\#/EN\# == 1$ ) або підлеглим ( $SP\#/EN\# == 0$ ). Як вихід, виконує заборону приймачів та передатчиків шини даних, коли дані передаються з контролера до ЦП.

**INT/** - вихід запиту переривання на процесор

**INTA/** - вхід підтвердження отримання переривання процесором

**IR0...IR7-** входи запиту

Основні функції контролера:

- 1) фіксація запитів на переривання від восьми зовнішніх джерел;
- 2) програмне маскуванню запитів, що поступають;
- 3) привласнення фіксованих або циклічно змінних пріоритетів входам контролера, на які надходять запити;
- 4) ініціація виклику процедури обробки апаратного переривання, що поступило.

Кількість обслуговуваних зовнішніх джерел переривань може бути збільшена шляхом каскадування декількох контролерів.

До складу контролера входять:

- схема управління читанням/записом;
- схема управління;
- схема каскадування;
- реєстр запитів на переривання;
- схема обробки пріоритетів;
- реєстр стану;
- реєстр маскуванню запитів на переривання.

ПКП може знаходитися в двох основних станах: налаштування і обслуговування запитів на переривання. В стані налаштування контролер приймає керуючі слова ініціалізації (Initialization Command Words, ICW), в стані обслуговування операційні керуючі слова (Operation Control Words, OCW).

### **2.3 Блок схема програмованого контролера переривань**

Контролер переривань обслуговує 8 запитів переривань і має вбудовані апаратні засоби для збільшення кількості обслуговуваних переривань до 64 шляхом підключення додаткових контролерів. Порядок обслуговуваних переривань задається програмно. Кожний запит переривання може бути заборонений (дозволений) програмним маскуванню. Порядок обслуговування переривань і маскуванню може динамічно змінюватися в довільний час.

### **Регістри INTERRUPT REQUEST REGISTER (IRR) та IN-SERVICE REGISTER (ISR)**

Запити переривань на входах IR керуються двома послідовно сполученими реєстрами обслуговуваних переривань (IRR і ISR). IRR призначений для зберігання усіх запитів переривань, які запитують обслуговування. ISR призначений для зберігання усіх запитів переривань, що обслуговуються. По фронту сигналу (запит переривань) на вході IR встановлюється розряд реєстру IRR рівня логічної «1» на лінії INT. В реєстрі IRR може бути встановлено і

декілька розрядів, якщо вони не замасковані. Гасіння IRR виконується послідовністю імпульсів /INTA від ЦП. Встановлення розрядів ISR імпульсом /INTA в той час, коли гаситься розряд в IRR. Цей розряд утримується на протязі усього часу роботи програми, до того часу поки контролер не отримає команду EOI ( кінець переривання).

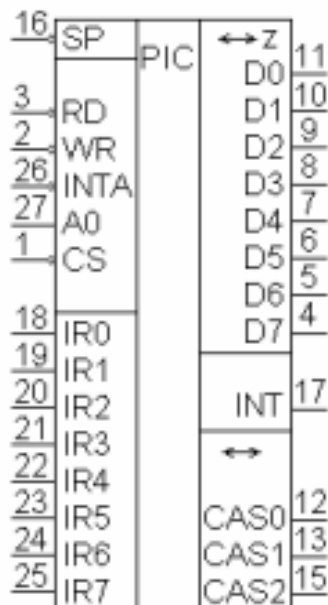


Рисунок 14 - Корпус мікросхеми I8259

### Розподілювач пріоритетів **PRIORITY RESOLVER**

Розподілювач пріоритетів визначає пріоритети розрядів, які встановлені в регістрі IRR. Вибирається розряд найвищого пріоритету і за допомогою імпульсу /INTA встановлюється відповідний розряд в регістрі ISR. Вивід INT (переривання) безпосередньо зв'язаний з відповідним входом ЦП. Вхід /INTA (підтвердження переривання) зв'язаний з регістром стану ЦП. Регістр стану виробляє три різних імпульси /INTA. За допомогою цих імпульсів контролер видає на шину даних 3-х байтну команду CALL. Ця команда забезпечує перехід ЦП до програми обслуговування переривань в відповідності з алгоритмом пріоритетів і в відповідності з умовами запитів переривань.

### Регістр маски **INTERRUPT MASK REGISTER (IMR)**

Регістр маски переривань (IMR) зберігає розряди запитів переривань, які повинні бути замасковані. Цей регістр працює на регістри IRR і ISR. Маскування розрядів з високим пріоритетом не впливає на запити переривань з низьким пріоритетом.

### Буфер даних **DATA BUS BUFFER**

Буфер шини даних, двонаправлений, 8-розрядний, з тристабільним станом використовується для зв'язку з контролером і передачі команд керування або зчитування слова стану.

### **Блок керування операціями читання та запису READ / WRITE CONTROL LOGIC**

Цей блок забезпечує прийом команд з ЦП. Він містить регістри команд ініціалізації (ICW) і робочих команд (OCW), які зберігають різні керуючі слова, що забезпечують роботу контролера. Крім того, цей блок дозволяє виводити на шину даних інформацію про стан контролера.

#### **Сигнал вибору мікросхеми CS (CHIP SELECT)**

Рівень логічного «0» на вході /CS дозволяє функціонування контролера.

#### **Сигнал – запис даних WR (WRITE)**

Рівень логічного «0» на вході WR/ дозволяє ЦП провести запис команд ICW і OCW в контролер.

#### **Сигнал - читання даних RD (READ)**

Рівень логічного «0» на вході RD/ дозволяє контролеру надсилати інформацію стану регістра запитів переривань (IRR), регістру обслуговуваних переривань (ISR), регістру маски переривання (IMR), або двійково-десятковий код номеру старшого пріоритету переривання на шині даних.

#### **Сигнал на вході A0**

Вхідний сигнал A0 використовується в сполученні з сигналами на входах WR/ і RD/ для запису команд в різні регістри команд, або читання інформації стану різних регістрів контролера. Вхід A0 може бути безпосередньо пов'язаний з однією із адресних ліній МП.

#### **Сигнал /SP**

Вхід SP/- сигнал підлеглого контролера, забезпечує збільшення кількості запитів переривання до 64, шляхом використання декількох контролерів переривань (каскадування). В цьому разі один контролер є головним (SP1), інші підлеглими (SP0).

#### **Блок буфера компаратора THE CASCADE BUFFER/COMPARATOR**

Блок буфера компаратора призначений для зберігання і порівняння ідентифікаторів усіх контролерів. Три контакти ВВ (CAS0.-CAS2) блоку є виходами, коли контролер є головним, або входами, коли контролер є підлеглим. Головний контролер посилає ідентифікатор CAS0-CAS2 для вибору підлеглого контролера. Вибраний контролер посилає адресу своєї підпрограми на шину даних протягом послідовних двох імпульсів /INTA.

## 2.4 Принцип роботи контролера

При здійсненні передачі даних з перериванням програми виконується наступна послідовність дій:

- встановлюється рівень логічної «1» на одній або декількох ліній запити переривання сигналізуючи контролеру, що зовнішній пристрій потребує обслуговування запити переривання;
  - контролер приймає ці запити, розподіляє пріоритети і надсилає сигнал INT до ЦП;
  - контролер підтверджує отримання сигналу INT посиленням сигналу /INTA;
  - при отриманні сигналів /INTA з регістру стану ЦП контролер виставляє на шину даних код команди CALL (11001101);
  - по цій команді виконується видача ще двох імпульсів сигналу /INTA;
  - ці імпульси дозволяють контролеру виставити на шину даних адресу підпрограми. По першому імпульсу виконується видача восьми молодших розрядів адреси, по другому – старших розрядів;
  - цим завершується 3-х байтна команда CALL, створена контролером.
- Регістр обслуговуваних переривань не гаситься до завершення програми, коли в контролер надходить команда EOI (команда кінця переривання).

## 2.5 Програмування контролера

Програмування VH59 здійснюється двома типами керуючих слів: ICW (Initialization Command Word) і OCW (Operation Command Word).

Три керуючих слова ICW1-ICW3 завантажуються перед початком роботи і встановлюють ВІС у режим фіксованих пріоритетів.

Для оперативного управління роботою контролера в будь-який час у нього можуть бути завантажені команди керування OCW1-OCW3.

Можливі наступні режими:

- режим повного вкладення;
- режим циклічного пріоритету;
- режим спеціального маскуванню;
- режим опитування.

### Команди ініціалізації

Контролер використовує три команди ініціалізації ICW1, ICW2 і ICW3.

Команда ініціалізації в якій біт керуючого слова D4=1 при A0=0 інтерпретується як команда ICW1. Після її отримання в контролері запускається послідовність ініціалізації, тобто автоматично виконуються наступні операції:

- гасяться всі тригери. Тому перехід в стан логічної «1» якого небудь входу (IR) після ініціалізації генерує запит переривання;
- гаситься регістр маски переривання;

- входу IR7 присвоюється пріоритет 7;
- гасяться тригери читання стану і реєстр спеціального маскуванню.

Два перших обов'язкових ICW1 і ICW2 визначають базову адресу таблиці входів. Адресний інтервал 4 або 8 задається бітом F в ICW1. Початкові адреси підпрограм обслуговування формуються за схемою:  $addr = base + k \times N$ , де  $k = 4$  або  $8$ ,  $N$  - номер запиту, прийнятого до обслуговування. У складі ICW1 біт S визначає відсутність або наявність каскадування ВН59. У каскадному варіанті завантажуються ще команда ICW3 в ведучий і ведений. ICW3 для ведучого містить 1 в розрядах, відповідних входів запитів, до яких підключені виходи INT ведених. ICW3 для веденого містить код номера входу ведучого, до якого він підключений.

Після ініціалізації ВН59 готовий до роботи в режимі фіксованих пріоритетів. Подальше управління роботою схеми здійснюється за допомогою команд OCW1 - OCW3, формати яких наведено на рисунку 15:

**ISW1:**  $A_0=0, D_4=1, CS=0$

7	6	5	4	3	2	1	0
$A_7$	$A_6$	$A_5$	1	0	F	S	0

S: S=1 – некаскадований контролер (один);

S=0 – декілька контролерів

F: інтервал адреси,

F=1 – 4 байти,

F=0 – 8 байт

$A_7$ - $A_5$ : старші розряди молодшого байта адреси

**ISW2:**  $A_0=1, CS=0$

7	6	5	4	3	2	1	0
$A_{15}$	$A_{14}$	$A_{13}$	$A_{12}$	$A_{11}$	$A_{10}$	$A_9$	$A_8$

$A_8$ - $A_{15}$ : старший байт початкової адреси обробки переривань

**ISW3:**  $A_0=1, CS=0$

7	6	5	4	3	2	1	0
$S_7$	$S_6$	$S_5$	$S_4$	$S_3$	$S_2$	$S_1$	$S_0$

для ведучого

$S_i=1$ , якщо до відповідного номеру входу запита ведучого підключений ведений

**ISW3:**  $A_0=1$

7	6	5	4	3	2	1	0
0	0	0	0	0			

для веденого

0,1,2-індексатор веденого

Рисунок 15 – Формати команд ICW1-ICW3

## 2.6 Робочі команди

### Робочі команди OCW

У довільний момент часу і незалежно від інших кожен запит командою OCW1 може бути замаскований. При встановленні 1 в розряді IMR забороняється приймання переривань з даного входу. Команда OCW2 служить для встановлення в 0 довільних розрядів ISR і циклічного зсуву пріоритетів з присвоєнням максимального значення будь-якому з восьми можливих рівнів. Для обслуговування запитів з рівними пріоритетами використовується циклічний зсув (команда EOI). При використанні команди зсуву одночасно зі скиданням ISR-біта, що має вищий пріоритет, реалізується циклічний зсув пріоритетів з присвоєнням нижчого щойно обслугованого рівня. Циклічний зсув не порушує послідовності вкладених один в одного переривань, що забезпечує правильне повернення з обслуговуючих їх програм. Пряма адресація рівня в слові OCW2 дозволяє скинути конкретний ISR- біт і таким чином завершити процедуру обслуговування цього запиту і циклічну зміну пріоритетів з явною вказівкою нижнього рівня. За допомогою OCW3 встановлюється режим спеціального маскуванню. У цьому режимі кожен біт в регістрі ISR забороняє тільки власний рівень, але дозволяє всі інші.

Командою OCW3 здійснюється управління режимом Поллінг (Polling) і вибір регістра IRR або ISR для читання його вмісту за допомогою програм. Режим ініціюється видачею в BH59 слова OCW3 з установкою біта P (Polling). Контролер визначає наступний цикл читання за  $A0 = 0$  як підтвердження переривання і видає на шину даних OCW3. За цим словом визначається запит з найвищим пріоритетом (рис. 16).



0	x	0	Немає операції
1	0	0	Немає операції
1	1	0	Спеціальний зсув
0	0	1	EOI
0	1	1	Спеціальний EOI
1	0	1	EOI з зсувом
1	1	1	Спеціальний EOI з зсувом



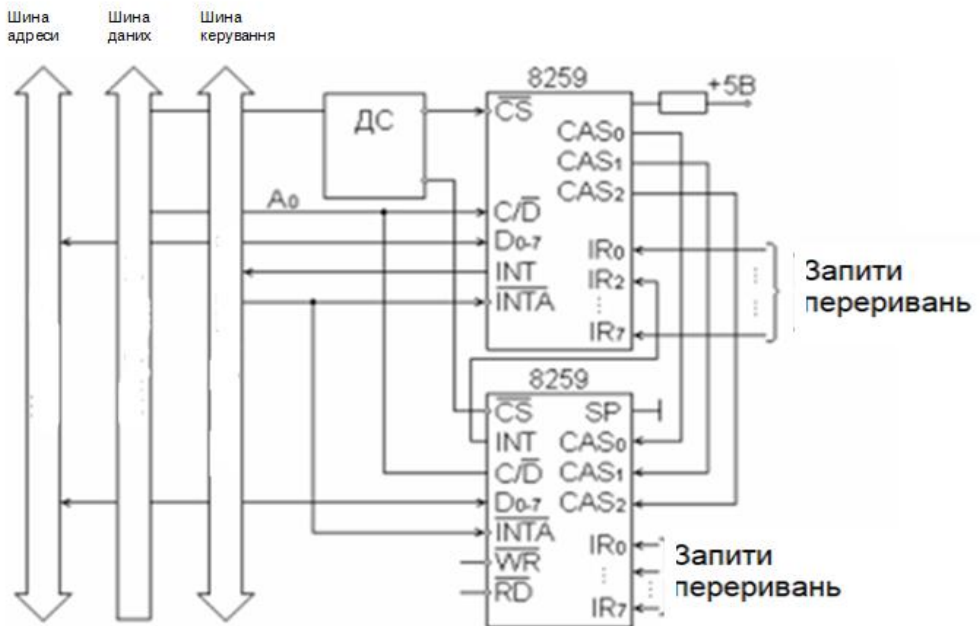


Рисунок 18 – Схема каскадування та підключення до шин

За допомогою емулятора програмованого контролера переривань I8259 прослідкуємо процес програмування пристрою. Загальний вигляд емулятора представлений на рисунку 19:

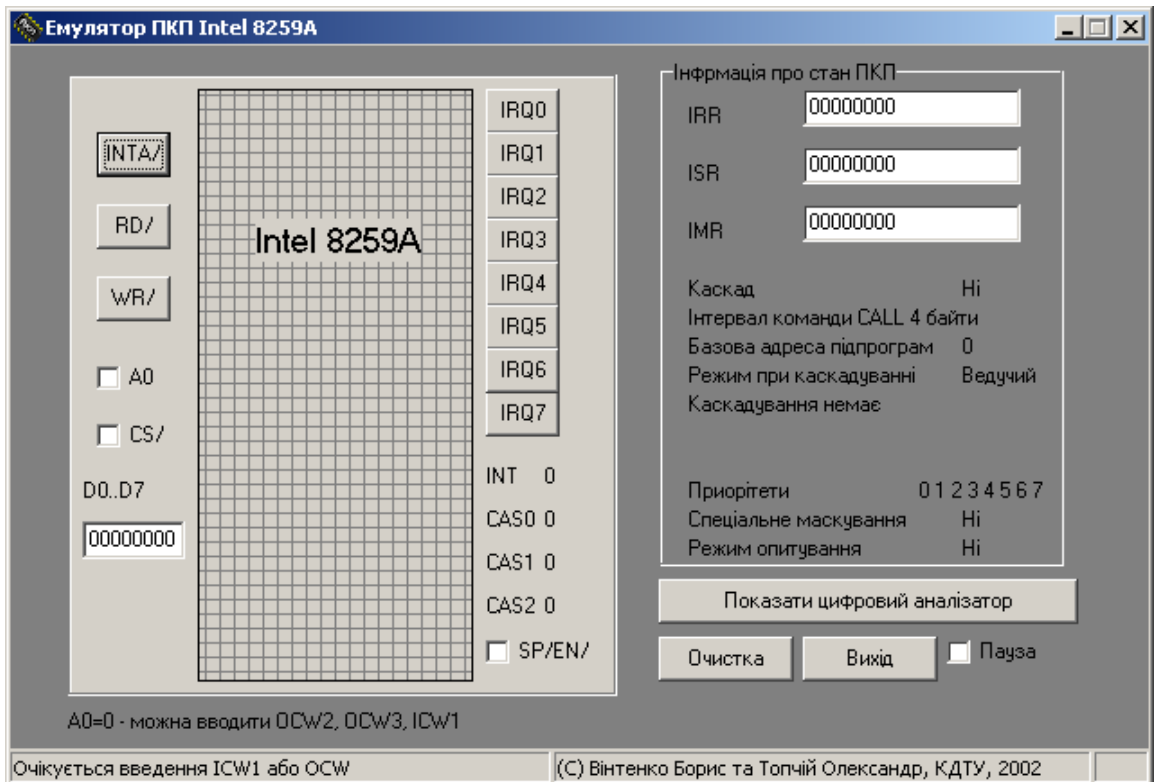


Рисунок 19 - Загальний вигляд емулятора контролера I8259

Порядок завантаження програми в емулятор:

1. Для початку роботи емулятора необхідно завантажити послідовно всі керуючі слова вибраного режиму роботи (команди ініціалізації та робочі команди), в віконце «D0..D7». У нижньому вікні є підказка яке керуюче слово можна ввести і яке очікує введення. Після введення кожного керуючого слова в віконце «D0...D7» необхідно для запису його у внутрішній регістр контролера натиснути кнопку «WR» і встановити необхідне значення A0. Після завантаження всіх команд в віконці «Інформація про стан ПКП» відобразиться зміст внутрішніх регістрів та інформація про запрограмований режим роботи.

2. Для початку обробки переривання необхідно натиснути кнопку вибраного запиту (кнопки IRQ0-IRQ7).

3. Спостерігати за роботою контролера можна на цифровому аналізаторі рис.21.

4. Перед перепрограмуванням контролера необхідно скинути всі попередні завдання (кнопка «Очистка»).

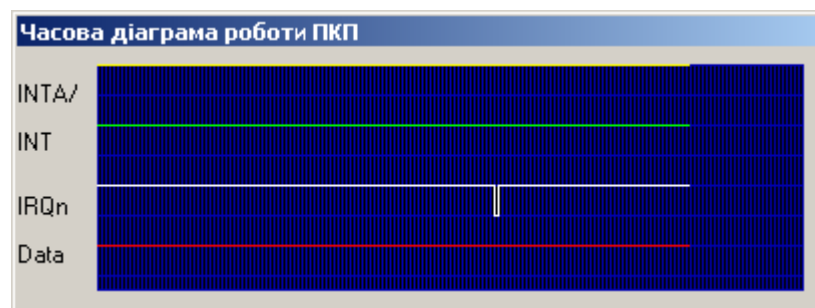


Рисунок 20- Цифровий аналізатор роботи ПКП

При запусканні програми можна в покроковому режимі відслідковувати завантаження всіх команд та роботу пристрою, який програмується.



Рисунок 21 - Загальний вигляд комплексу програмування контролера переривань

Відлагоджувач програм показано на рисунку 22:

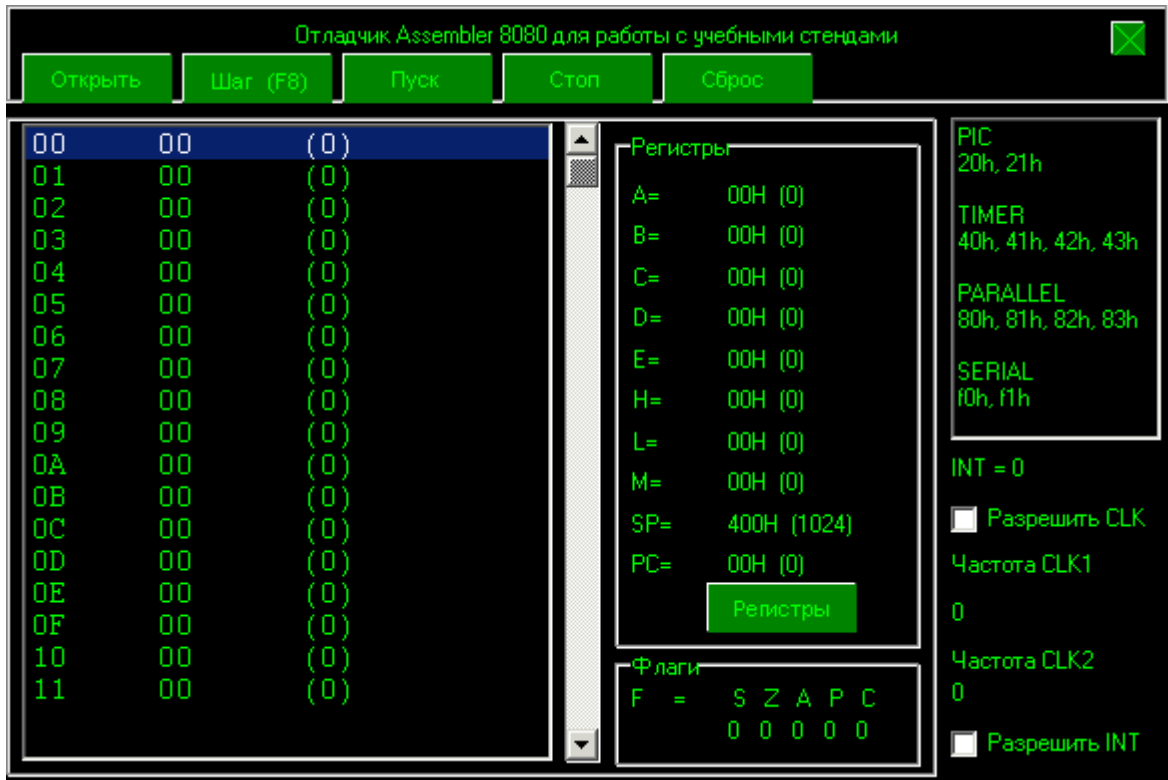


Рисунок 22 - Відлагоджувач програм

### Приклад ініціалізації на комплексі І8259, АSM-80

```

Pic a0          20h
Pic a1          21h

Програмування КП
mvi a,00010110b          ;ICW1 по 4
    out 20h
    mvi a,0h              ;ICW2
    out 21h
    mvi a,.....         ;OCW1 якщо потрібно
    out 21h
    mvi a,.....         ;OCW2 якщо потрібно
    out 21h

m1: головна програма
jmp m1
    hlt
Таблиця переходів
org 0000h
    call ir0          ; INT0
    ret              ; повернення з підпрограми
org 0004h
    call ir1          ; INT1
    ret              ; повернення з підпрограми
org 0008h
    call ir2

```

```

    ret
org 000bh
    call ir3
    ret
org 000fh
    call ir4
    ret
org 0010h
    call ir5
    ret
org 0014h
    call ir6
    ret
org 0018h
    call ir7
    ret

```

Таблиця векторів

```

org 0100h
ir0:  mvi b,0           ; номер вектора
      ret
ir1:  mvi b,1
      ; підпрограма
      mvi a,20h        ; OCW2 кінець переривання
      out 20h          ;
      ret
ir2:  mvi b,2
      ; підпрограма
      mvi a,20h        ; OCW2 кінець переривання
      out 20h          ;
      ret
ir3:  mvi b,3
      ret
ir4:  mvi b,4
      ret
ir5:  mvi b,5
      ret
ir6:  mvi b,6
      ret
ir7:  mvi b,7
      ret
      hlt
end

```

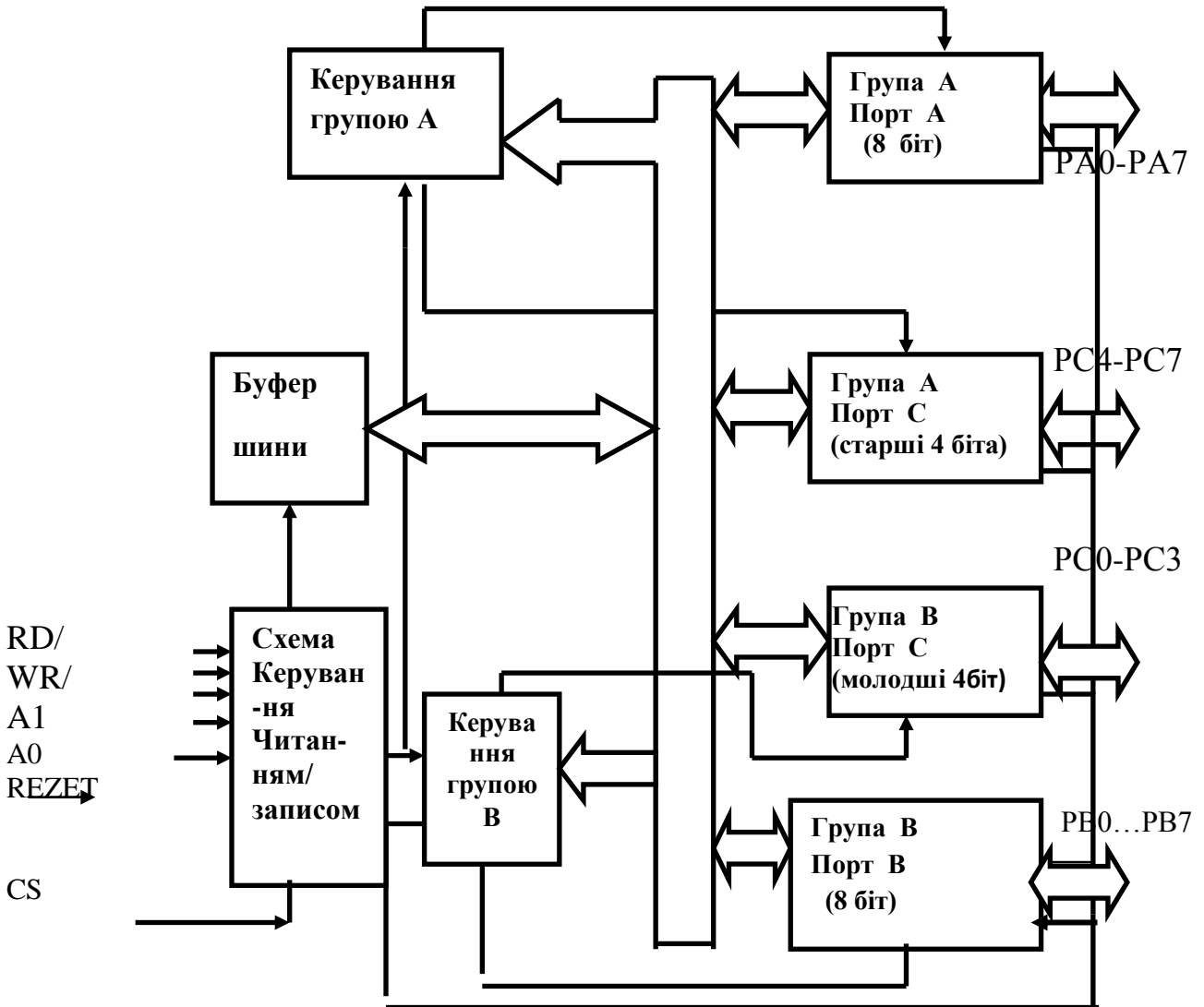
## Контрольні питання та завдання

1. Поясніть, які можливості мають МП для організації обміну за запитами переривання (на прикладі МП ВМ80, ВМ85, ВМ86).
2. Яка організація радіальної системи переривань?
3. Поясніть переваги і недоліки прийому запитів на переривання у вигляді рівнів напруг (високого / низького) і фронтами сигналу.
4. Що розуміється під однорівневими і багаторівневими перериваннями?
5. Які способи формування початкової адреси підпрограм для обслуговування переривань вам відомі?
6. Наведіть приклади векторної 8-рівневої системи переривання.
7. Назвіть функції програмованого контролера ВН59.
8. Назвіть керуючі слова ініціалізації ВН59 та їх призначення.
9. Які функції виконують операційні управляючі слова ОСW1-ОСW3?
10. Зобразіть по пам'яті структурну схему контролера ВН59 і поясніть призначення її компонентів.
11. Назвіть режими обробки запитів по пріоритетам контролера ВН59.
12. На лабораторній роботі написати програму ініціалізації контролера.

### Лекція 3

## 3.1 Програмований паралельний інтерфейс введення-виведення K580BB55

### Блок схема адаптера



D0...D7 -лінії даних  
REZET –апаратне скидання  
CS/ -вибір корпусу ВІС  
RD -вхід керування введенням  
WR/ -вихід керування висновком  
A0, A1 -адреса порту  
РА0...РА7 -порт А  
РВ0-РВ7 - порт В  
РС0-РС7-порт С

Рисунок 23 – Блок схема адаптера

### 3.2 Принцип роботи ВІС ВВ55

ВІС паралельного інтерфейсу I8255, блок-схема якої представлена на рис.24, містить три 8-ми розрядні порти А, В, С. Порти А, В, С керуються програмно і можуть бути включені в різних функціональних конфігураціях, при цьому кожний з портів має притаманні йому функціональні характеристики. Характеристики портів:

Порт А – один 8-ми розрядний буферний регістр виводу даних і один 8-ми розрядний регістр вводу даних;

Порт В - один 8-ми розрядний буферний регістр ВВ даних і один 8-ми розрядний буфер вводу даних;

Порт С - один 8-ми розрядний буферний регістр виводу даних і один 8-ми розрядний буфер вводу даних. Цей порт може бути поділений на 2 4-ох розрядних порти. Кожний 4-ох розрядний порт має 4-ох розрядний регістр і може використовуватись при передачі сигналів керування і приймання сигналів стану при роботі сумісно з портами А і В.

ЦП задає режими роботи портів за допомогою двох типів керуючих слів:

- 1) команди встановлення режимів роботи (розряд D7=1);
- 2) команди встановлення розряду порта С (розряд D7=0).

Формати слів керування і встановлення даних відповідно на рис. 24, 25.

Тип слова керування визначається станом розряду D7.

Формат команди встановлення/гашення розряду порта С

<b>D7</b> <b>0</b>	<b>D6</b>	<b>D5</b>	<b>D4</b>	<b>D3</b>	<b>D2</b>	<b>D1</b>	<b>D0</b>
-----------------------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

D7 – ознака слова

D6,D5, D4 – не використовуються

D3, D2, D1 – вибір біту порту С

D0 – 0 встановлення в 0; 1-в 1

Рисунок 24 – Формат команди встановлення гашення розряду порта С

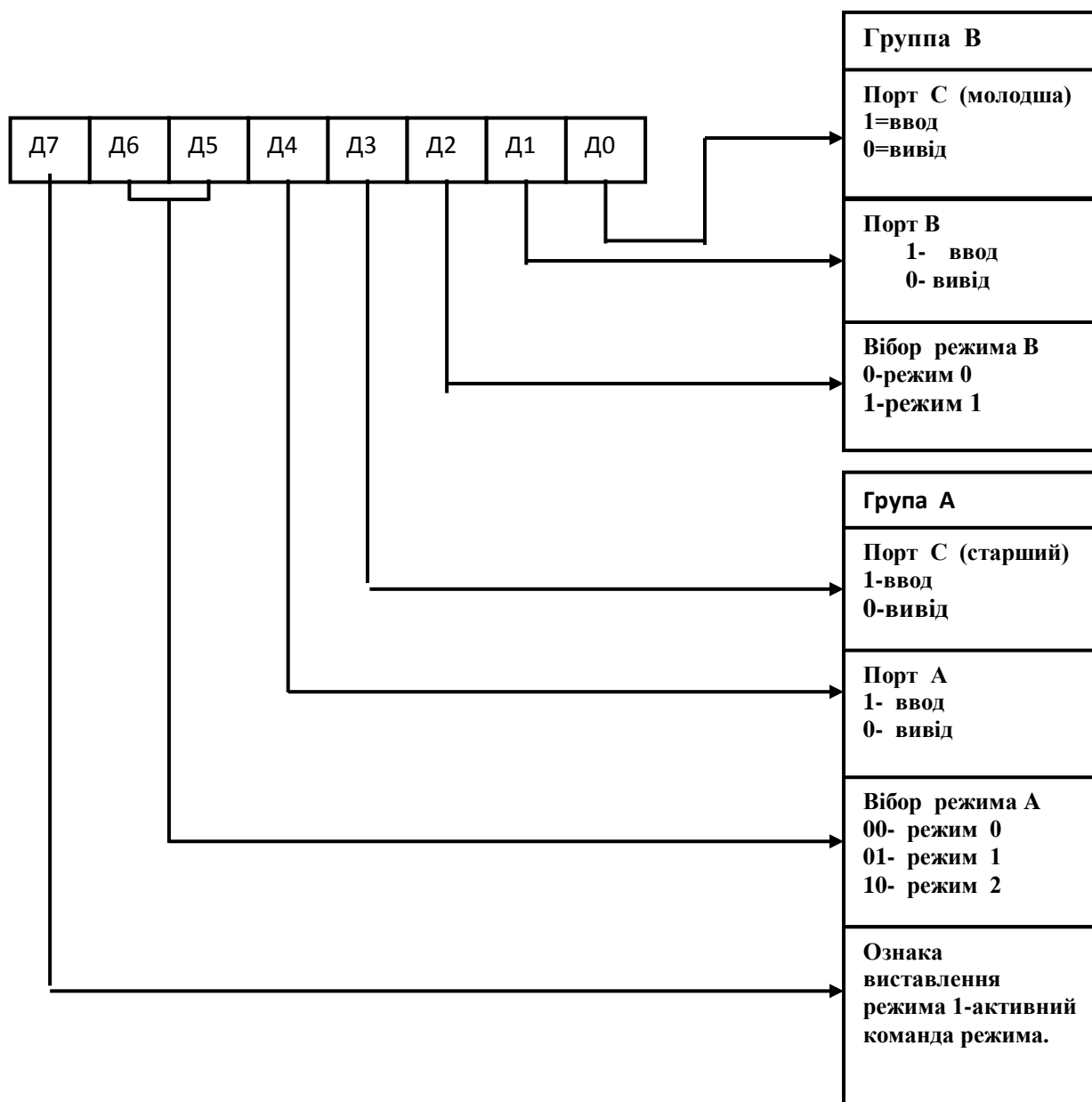


Рисунок 25 - Керуюче слово режиму

### Вибір режиму роботи

ВІС паралельного інтерфейсу може працювати в одному з трьох режимів, які задаються програмно:

Режим 0 – основний ВВ

Режим 1 - стробований ВВ

Режим 2 - двонаправлена передача даних.

При встановленні на вході R рівня "1" адаптер встановлюється у початковий стан, всі порти налаштовуються на введення в режимі 0 (тобто всі 24

вихідні лінії встановлюються в стан з високим опором). Після зняття сигналу встановлення у початковий стан порти адаптера залишаються в цьому режимі, додаткової команди режиму не потрібно. Для виконання програми вибору інших режимів роботи застосовується одна команда. Це дозволяє адаптеру обслуговувати велику кількість зовнішніх пристроїв засобами програмування.

Режими роботи портів А і В можуть задаватися незалежно одне від одного, а порт С поділений на дві частини, режими роботи яких задаються в залежності від призначення режимів портів А і В. При зміні режимів роботи усі вихідні регістри адаптера, а також тригери стану, встановлюються в початковий стан, за виключенням тригера готовності (вивід готовий) у режимах 1 і 2. Вибираючи різні комбінації режимів роботи портів можна забезпечити необхідну функціональну організацію практично для будь-якої структури ВВ.

### **Керування портом С**

Кожний з 8 розрядів порта С може бути встановлений в «1» або «0» за допомогою однієї команди виведення. Ця можливість спрощує програмування ВІС паралельного інтерфейсу при використанні керуючих сигналів.

В випадку, коли порт С використовується для керування портом А або В, ці розряди можуть бути встановлені в «1» або «0» словом стану порта С. При цьому група порта С, в яку входить встановлюваний розряд, включається на виведення.

### **Керування сигналами переривання ВІС паралельного інтерфейсу**

При роботі в режимі 1 або 2 ВІС паралельного інтерфейсу формує сигнали, які можуть бути використані в якості запитів переривань до ЦП. Сигнали запитів переривань, які формуються в порту С, забороняються або дозволяються встановленням в «1» або гашенням тригера переривань за допомогою команди встановлення порту С. Це дозволяє керувати запитами переривань одних зовнішніх пристроїв до інших пристроїв в структурі переривання. При встановленні відповідного розряду порту С в «1» тригер переривань встановлюється в «1» і дозволяє переривання, а при встановленні цього розряду в «0» тригер переривань встановлюється в «0» і забороняє переривання.

При виборі режиму роботи портів і встановленні по входу R в початковий стан усі тригери дозволу гасяться.

## **3.3 Режими роботи K580BB55**

### **Режим 0 (головний ВВ)**

Функціональна конфігурація режиму 0 I8255 забезпечує простий ВВ даних через кожний з трьох портів. В цьому режимі немає сигналів керування асинхронною передачею даних, дані просто записуються в порт або зчитуються з

нього. Часові діаграми роботи ВІС паралельного інтерфейсу в режимі 0 представлена на рисунку 26:

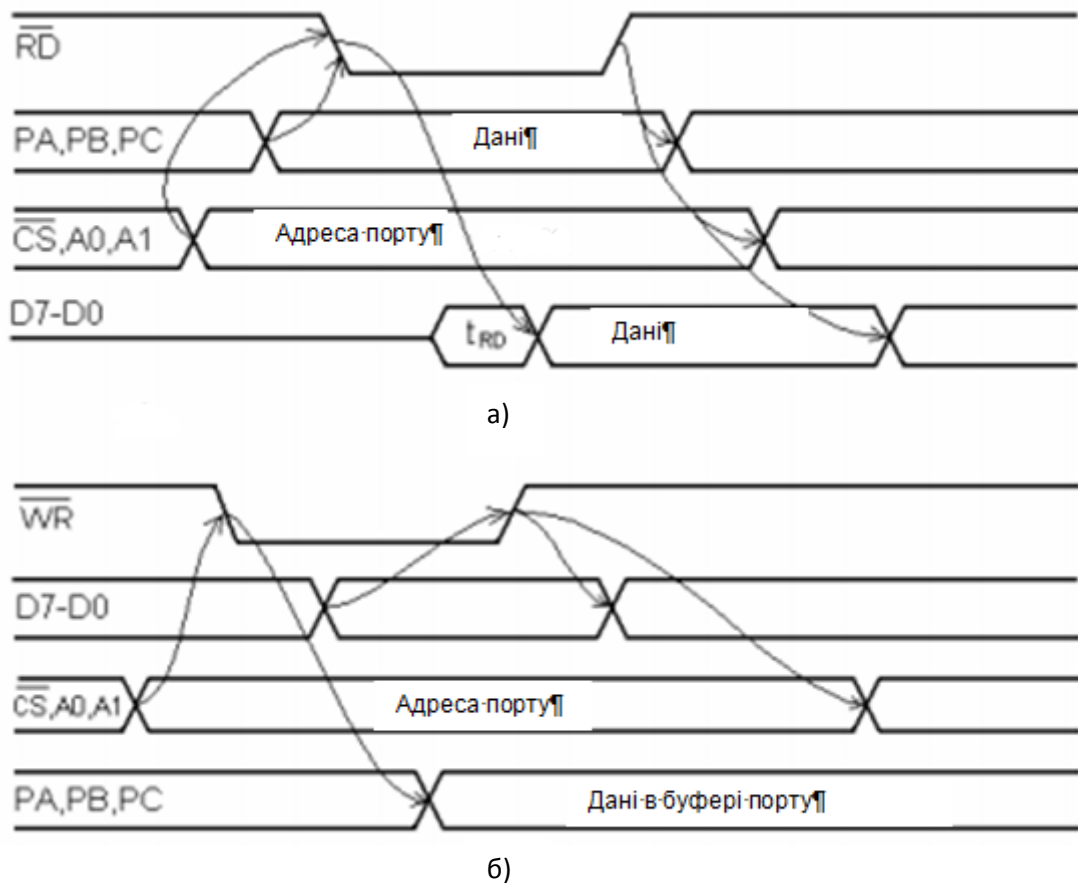


Рисунок 26 – Часові діаграми роботи режиму 0:  
а) режим введення; б) режим виведення

Режим 0 має наступні функціональні характеристики:

- два 8-розрядних порти і два 4-х розрядних порти;
- будь-який порт може бути включений на введення або виведення;
- виведення даних з проміжним збереженням;
- введення даних без збереження.

В цьому режимі можливі 16 конфігурацій ВВ; 4 можливих (з 16) конфігурацій представлено на рисунку 27:

MODE 0 PORT DEFINITION								
A		B		Group A			Group B	
D <sub>4</sub>	D <sub>3</sub>	D <sub>1</sub>	D <sub>0</sub>	Port A	Port C (Upper)	#	Port B	Port C (Lower)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

Рисунок 27 - Комбінація схем ВВ каналів I8255 в режимі 0

### Режим 1(стробований ВВ)

Функціональна конфігурація ВВ в цьому режимі забезпечує передачу даних в заданий порт або з нього сумісно з асинхронними сигналами керування передачею. В режимі 1 порт А і порт В використовують порт С для формування або прийому сигналів керування.

Режим 1 має наступні функціональні характеристики:

- 2 порти передачі даних ( А і В);
- кожний порт передачі містить восьмирозрядний порт даних і чотирьохрозрядний порт керування/даних (порт С);
- восьмирозрядний порт даних може бути включений на введення або виведення. ВВ даних здійснюється з проміжним збереженням.

Сигнали керування вводом в режимі 1:

- STB (Strobe-вхід стробу).

При рівні логічного «0» дані записуються у вхідний регістр;

- IBF ( Input Buffer Full-ознака заповнення регістру вводу).

Високий рівень логічної «1» на цьому вході вказує на те, що дані записані у вхідний регістр; сигнал IBF встановлюється в «1» по зрізу сигналу STB і гаситься по фронту сигналу RD ;

- INTR ( Interrupt - запит переривання).

Високий рівень на цьому вході може бути використаний для переривання ЦП при запиті обслуговування периферійним пристроєм. Сигнал INTR встановлюється в стан логічної «1» по фронту сигналу STB при IBF=1 і при внутрішньому сигналі дозволу переривання INTE=1. Гасіння запиту переривання

виконується зрізом сигналу RD. Такий засіб формування переривання дозволяє периферійному пристрою зробити запит обслуговування простим стробуванням даних, які передаються в порт.

- INTRa- внутрішній сигнал дозволу переривання, який встановлює розряд PC3 порту C.

-INTRb- внутрішній сигнал дозволу переривання, який встановлює розряд PC0 порту C.

Конфігурація і часова діаграма введення в режим 1 показані на рис.28 і 29 відповідно:

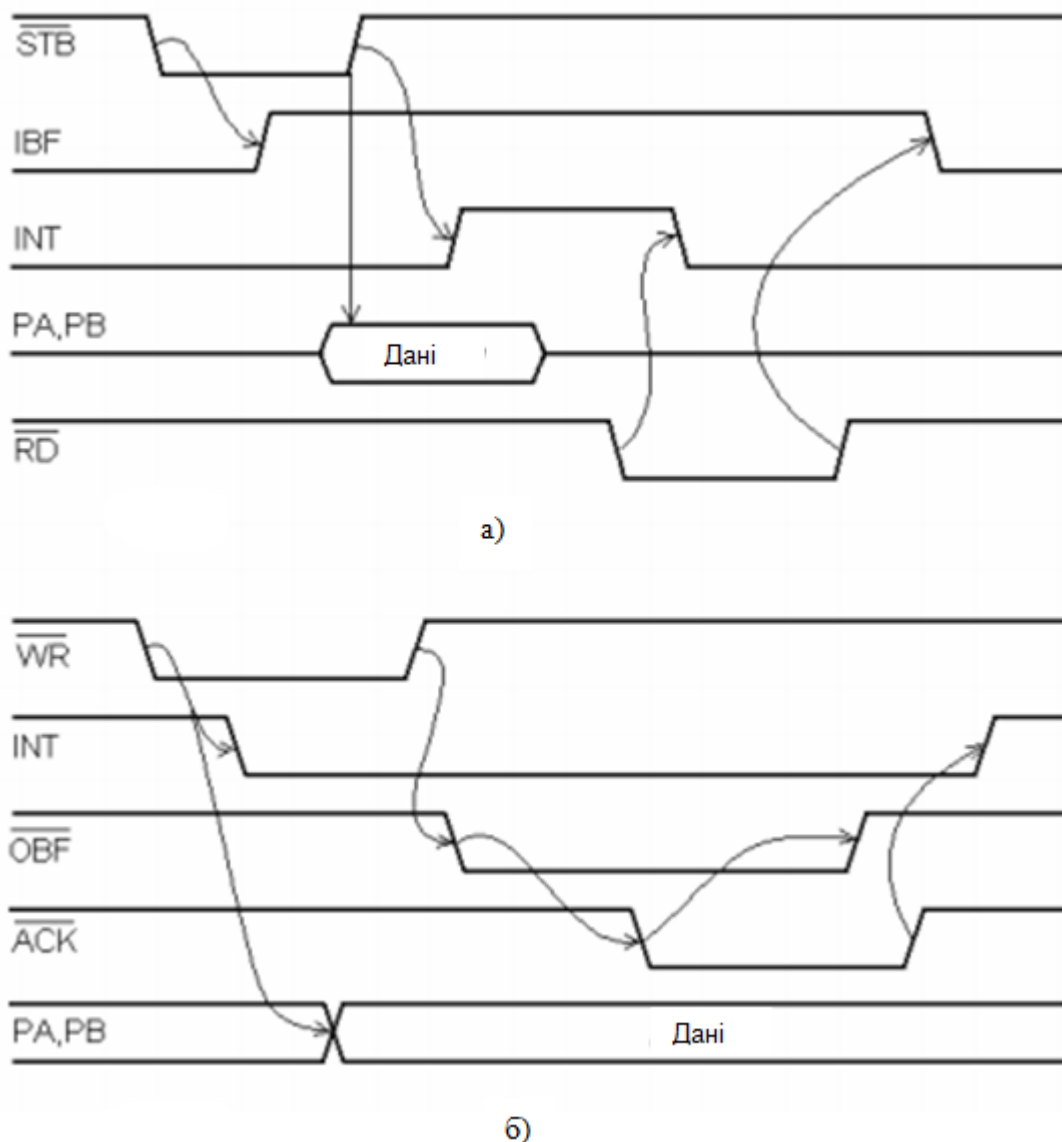


Рисунок 28 - Часові діаграми роботи режиму 1:  
а) режим введення; б) режим виведення

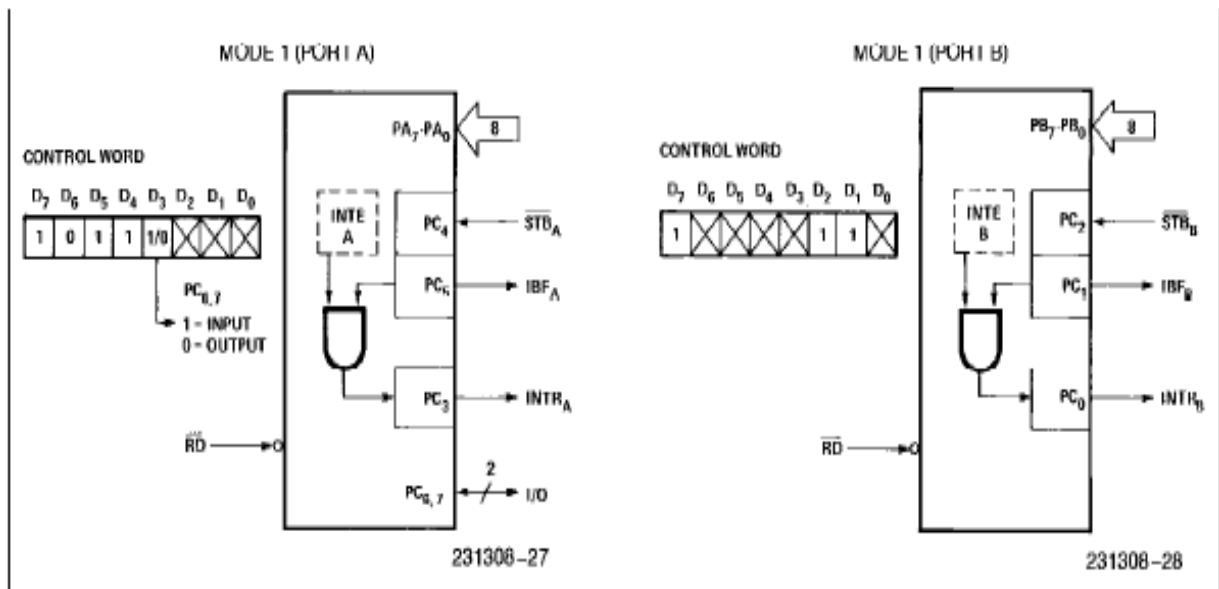


Рисунок 29 - Режим 1(введення)

Сигнали керування виведенням в режимі 1:

- /OBF (Output Buffer Full - ознака заповнення регістра виводу).

Переключення виводу до низького рівня логічної «0» вказує на те, що ЦП закінчив запис даних в заданий порт. Сигнал /OBF встановлюється по фронту сигналу WR і гаситься по зрізу сигналу /ACK.

- /ACK (ACKnowledge-підтвердження прийому).

Низький рівень логічного «0» сигналу повідомляє ВІС, що дані в порту А або В прийняті. Тобто це сигнал відповіді периферійного пристрою, який вказує на те, що прийняті дані, які передав ЦП.

- INTR (запит переривання).

Цей сигнал може бути використаний для переривання роботи ЦП для приймання даних периферійним пристроєм. Вихід INTR встановлюється в стан логічної «1» по фронту сигналу /ACK при /OBF =1 і INTE=1. Гашення сигналу INTR виконується по зрізу сигналу WR.

- INTRa.

Внутрішній сигнал дозволу переривання, який встановлює розряд PC3 порту С.

- INTRb.

Внутрішній сигнал дозволу переривання, який встановлює розряд PC0 порту С.

Конфігурація і часова діаграма виводу в режимі 1 ВІС показані на рис.30 і 31 відповідно:

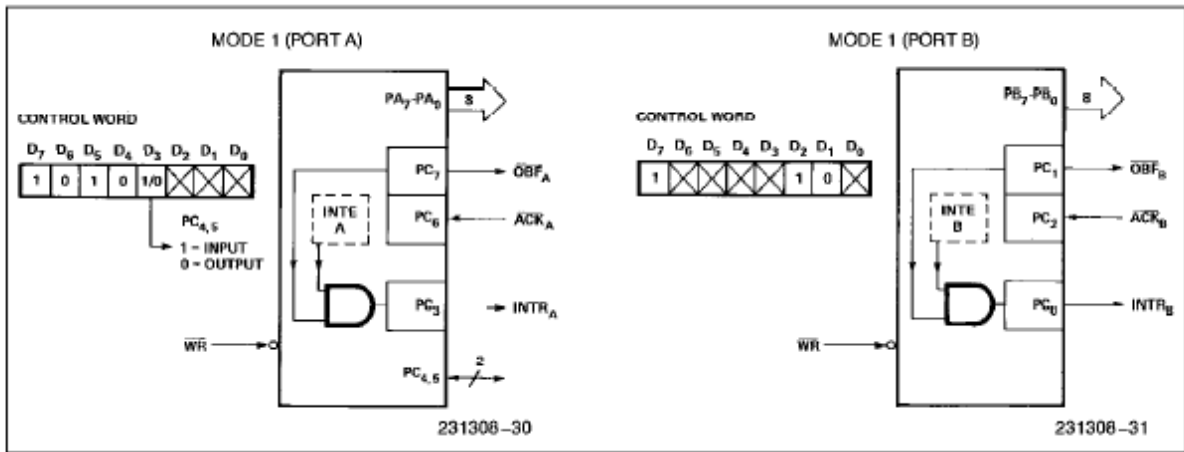


Рисунок 30 - Режим 1(виведення)

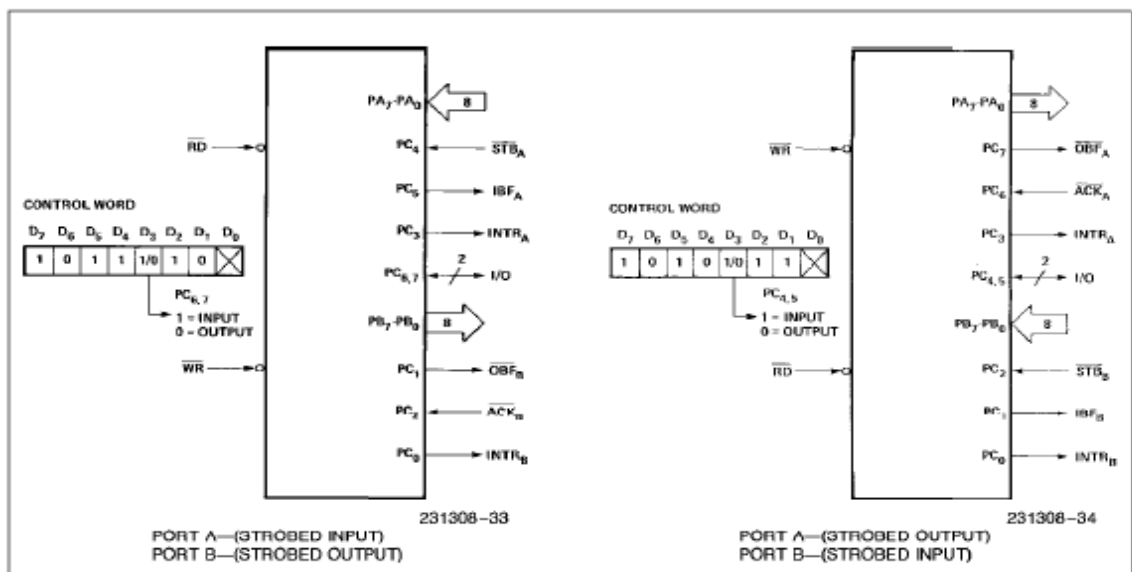


Рисунок 31 - Конфігурація режиму 1

## Режим 2

Функціональна конфігурація ВІС в режимі 2 забезпечує зв'язок з периферійними пристроями по одній 8-ми розрядній двонаправленій шині. Керування передачею даних здійснюється тими ж асинхронними сигналами, що і в режимі 1. Здійснюється також формування сигналів переривання і дозволу або заборони переривання.

Режим 2 має такі функціональні характеристики:

- в цьому режимі працює тільки порт А;
- один 8-ми розрядний порт двонаправленої передачі даних (порт А) і 8-ми розрядний порт керування (порт С);
- ВВ даних із збереженням;
- 5-розрядний порт керування (порт С) використовується для прийому і передачі сигналів керування портом двонаправленої передачі (портом А).

Сигнали керування двонаправленою передачею в режимі 2:

- INTR (запит переривання). Високий рівень на цьому виході (логічна «1») може бути використаний для переривання ЦП при операції ВВ.

Сигнали керування виводом в режимі 2:

- /OBF (ознака заповнення регістра виведення).

Вихід /OBF переключенням на низький рівень логічного «0» вказує на те, що ЦП записав дані в порт A.

- /ACK (підтвердження). Низький рівень на цьому вході з тристабільним станом вихідного буфера порта дозволяє передачу даних. В іншому випадку, вихідний буфер буде в стані високого імпедансу.

- INTE (INTE Flip-Flop, зв'язаний з /OBF). Контролює встановлення або скидання біту PC6.

Сигнали керування введенням в режимі 2:

- /STB (Strobe Input). Низький рівень на цьому вході дозволяє завантаження даних у вхідний буфер.

- IBF (Input Buffer Full F/F). Високий рівень на цьому виході вказує, що дані можуть бути зчитані з виходу зовнішнього пристрою.

- INTE (INTE Flip-Flop, зв'язаний з IBF). Контролює встановлення або скидання біту PC4.

Сигнали IBF або OBF інформують зовнішній пристрій про готовність прийняти або передати дані. В відповідності зі станом IBF або OBF зовнішній пристрій або генерує чергові дані, супроводжуючи їх стробом STB, або формує сигнал підтвердження прийому ACK, готуючись до прийому даних. Низький рівень сигналу ACK відкриває вихідні буфери порту A, дозволяючи видачу даних на шину. В інших випадках шина порту A знаходиться в Z-стані.

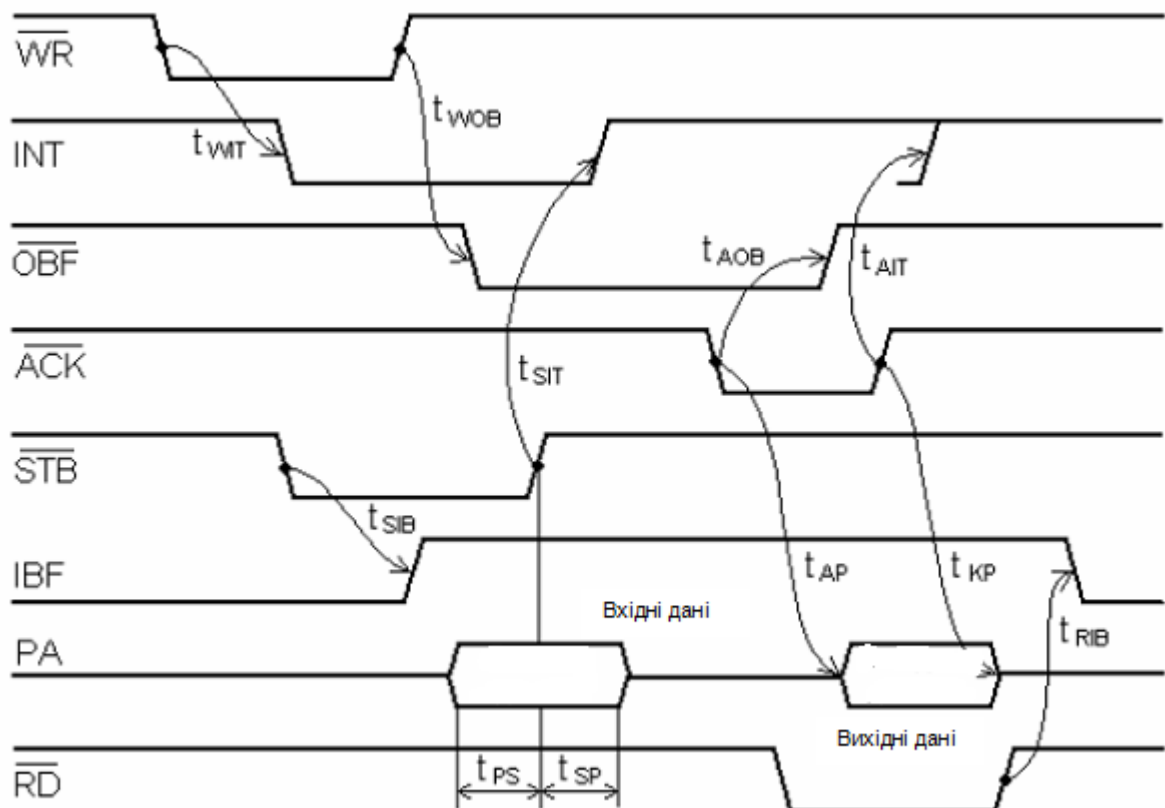


Рисунок 32 - Часова діаграма роботи ВІС в режимі 2

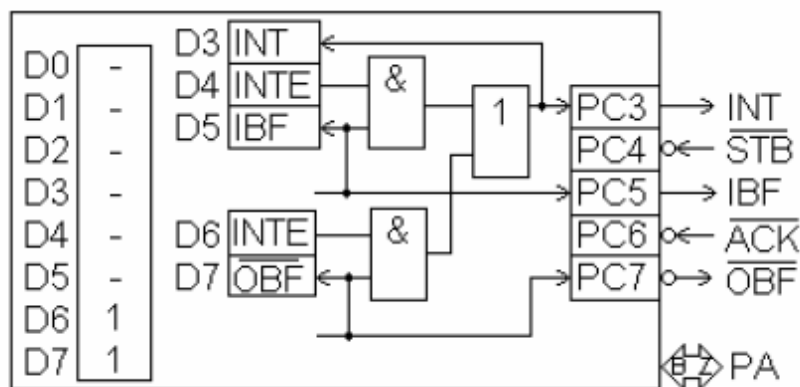


Рисунок 33 - Організація двонаправленого ВВ

Порти А, В, С для роботи в вище згаданих режимах програмуються незалежно один від одного.

За допомогою емулятора паралельного інтерфейсу I8255 прослідкуємо процес програмування пристрою. Загальний вигляд емулятора представлений на рис.34 (vv55-2.exe).

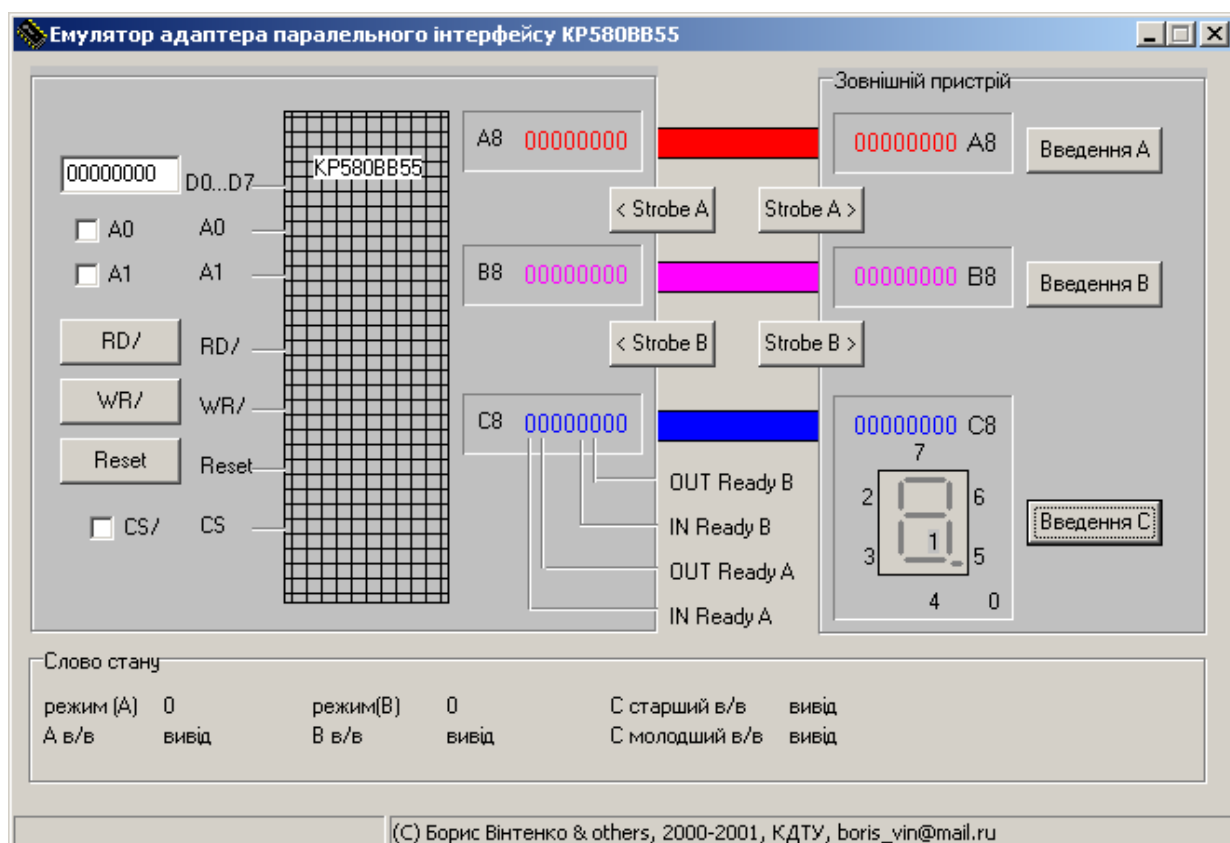


Рисунок 34 - Загальний вигляд емулятора адаптера I8255

### Завантаження програми до емулятора

1. Для початку роботи емулятора необхідно завантажити керуюче слово вибраного режиму роботи (режим 1, 2, 3) в вікнці «D0 D7». Так як адреса

регістра керуючого слова RUS 43h, встановимо A0 і A1 в 1 (мітку у віконці A0 і A1), потім наберемо керуюче слово і натиснемо кнопку «WR\». В віконці «Слово стану» відобразиться заданий режим.

2. Послідовно до кожного порту введемо дані при відповідному значенні A0 ті A1 (порт А 80h A0=0,A1=0; порт В 81h A0=1,A1=0; порт С 82h A0=0,A1=1). Після кожного введення необхідно натиснути кнопку «WR\», запис до відповідного порту.

В режимі 0 дані будуть передані до зовнішнього пристрою і відобразяться у відповідному віконці. Якщо дані будуть передаватися з зовнішнього пристрою до адаптера, то необхідно виконати введення (кнопки відповідних портів «Введення А», «Введення В», «Введення С»).

3. В режимі 1 та 2 після програмування адаптера (пункт 1-3) для виконання передачі або прийому по сигналу «Strob» необхідно після введення даних натиснути кнопку «Strob» відповідного каналу на прийом або на передачу в залежності від заданого режиму роботи.

4. Перед перепрограмуванням адаптера виконаємо його скидання натисканням кнопки "Reset".

5. До порту С підключений семисегментний індикатор (розряди порту до яких підключенні сегменти вказані на рисунку).

Використавши комплекс для програмування адаптера, загальний вигляд якого представлений на рисунку 35, можна набрати та відлагодити програму роботи пристрою. При запуску програми є можливість в покроковому режимі відслідкувати завантаження всіх команд та роботу пристрою, який програмується на індикаторах формату команд, даних та вхідних і вихідних сигналів ВІС.



Рисунок 35 - Загальний вигляд комплексу програмування адаптеру

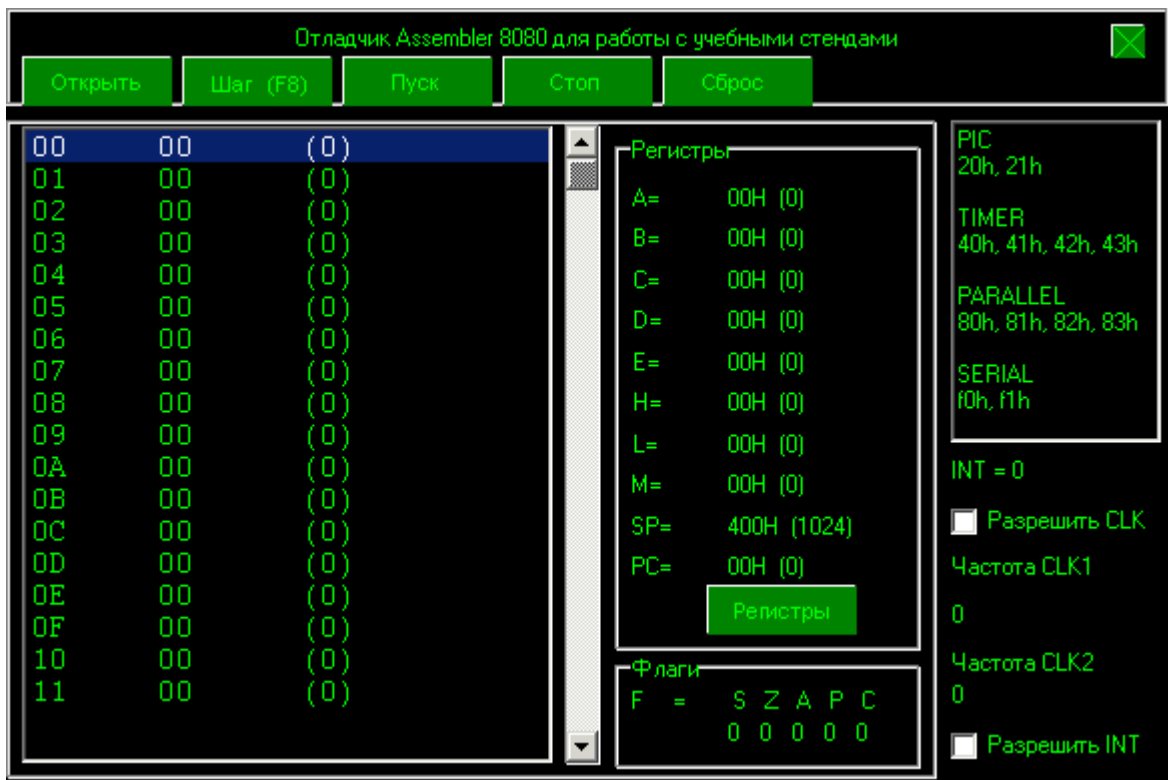


Рисунок 36 - Загальний вигляд вікна компілятора

### 3.4 Приклади програмування

#### Прийом / передача даних в режимі 0

Виведення даних в порт A, B; введення з C в режимі 0

Port A            80h

Port B            81h

Port C            82h

RUS              83h

Org 2000h

Керуюче слово режиму 10001001-89h

    Mvi a,89h            ; RUS

    Out 83h

    Mvi a,xxh            ; виведення даних

    Out 80h`            ; в A

    Mvi a,xxh

    Out 81h            ; в B

    In 82h            ; з C

    Mov m,a

    Hlt

    end

## Контрольні питання та завдання

1. Архітектура ВІС паралельного інтерфейсу І8255. Призначення, структурна схема, призначення основних блоків, вхідних/вихідних сигналів.
2. Система мікрокоманд, формати команд, формат слова-стану. Режими роботи ВІС КР580ВВ55 (паралельний інтерфейс). Підключення до шин МП.
3. Розробіть підпрограму ВВ байта через порт В в режимі 0.
4. Розробіть схему підключення друкуючого пристрою і програму управління обміном, використовуючи режими 1 портів А і В.
5. Визначте переваги режиму 1 у порівнянні з режимом 0 на основі аналізу розроблених вами програм управління.
6. Складіть часові діаграми обміну даними з квітуванням.
7. Зобразіть схему підключення І8255 до шини МПС.
8. Поясніть використання SW при організації ВВ даних.

## Лекція 4

### 4.1 Контролер прямого доступу до пам'яті

Контролер прямого доступу до пам'яті (ПДП, DMA - Direct Memory Access) забезпечує високошвидкісний обмін даними між пристроями ВВ й ОЗП без використання центрального процесора, що дозволяє звільнити процесор для виконання обчислень паралельно з обміном і незалежно від нього. Найбільш часто можливості ПДП використовуються при роботі з дисковими накопичувачами. Відчутні переваги дає використання ПДП у процесі обміну з пристроями, що приймають або передають дані досить великими порціями з високою швидкістю.

### 4.2 Принципи роботи контролера ПДП

Загальні принципи організації ПДП. Режим ПДП є самим швидкісним способом обміну, який реалізується за допомогою спеціальних апаратних засобів - контролерів ПДП без використання програмного забезпечення. Для здійснення режиму ПДП контролер повинен виконати ряд послідовних операцій:

- 1) прийняти запит DREQ на ПДП від ВП;
- 2) сформувані запит HRQ на захоплення шин для ЦП;
- 3) прийняти сигнал HLDA, що підтверджує цей факт після того, як ЦП увійде у стан захоплення (ШД, ША, ШК в z-стан);
- 4) сформований сигнал DACK, повідомляє ВП про початок виконання циклів ПДП;
- 5) сформувані на ША адресу комірки пам'яті, призначений для обміну;
- 6) виробити сигнали, IOR, MW і IOW, MR, що забезпечують управління обміном;
- 7) після закінчення циклу ПДП або повторити цикл ПДП, змінивши адресу, або припинити ПДП, зняттям запиту на ПДП.

Цикли ПДП виконуються з послідовно розташованими осередками пам'яті, тому контролер ПДП повинен мати лічильник адреси ОЗП. Число циклів ПДП визначається спеціальним лічильником. Управління обміном здійснюється спеціальної логічною схемою, формує в залежності від типу обміну пари керуючих сигналів: IOW, MR (цикли читання), IOR, MW (цикли запису). З викладеного випливає, що контролер ПДП на його вимогу повинен взяти на себе керування системними шинами і виконувати суміщені цикли читання/виведення або запису/введення до тих пір, поки вміст лічильника циклів ПДП не дорівнюватиме нулю. На рис. 37 показана структурна схема МПС з контролером ПДП.

КПДП дозволяє реалізувати передачу пам'ять-пам'ять, маючи широкі можливості програмного керування і каскадування. Кожен канал може виконати до 64К циклів ПДП і має можливість автоматичної ініціалізації, тобто повторення циклів ПДП з тими ж параметрами.

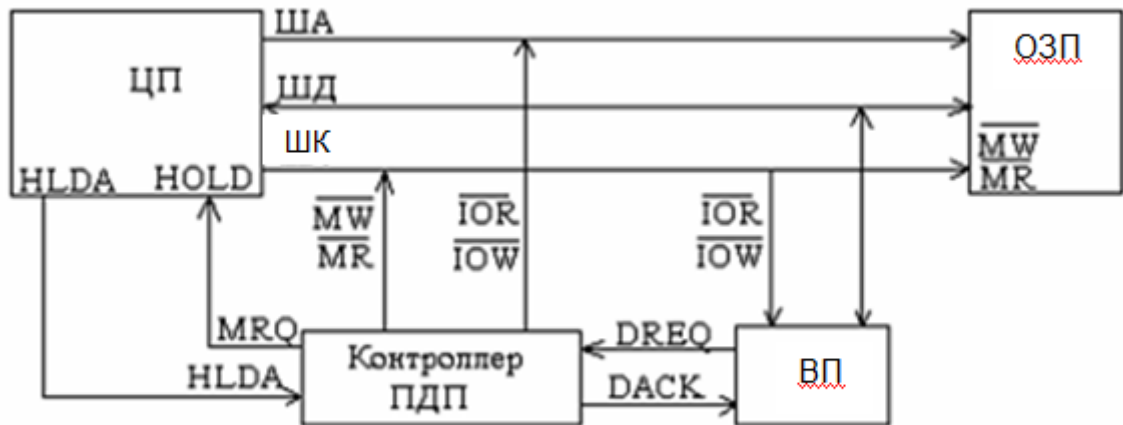


Рисунок 37 – Структурна схема МПС з контролером

У роботі КПДП розрізняються 2 головних цикли: цикл очікування (Idle cycle) і активний цикл (Active cycle). З циклу очікування контролер може бути переведений у стан програмування (Program Condition) шляхом подачі на вхід RESET сигналу високого рівня, тривалістю не менш 300 нс і наступної за ним подачі сигналу низького рівня (рівня 0) на вхід CS (Chip Select). У стані програмування контролер буде знаходитися доти, поки на вході CS зберігатиметься сигнал низького рівня. У процесі програмування контролеру задаються:

- початкова адреса пам'яті для обміну;
- зменшене на одиницю число переданих байтів;
- напрямок обміну,
- а також встановлюються необхідні режими роботи (дозволити або заборонити циклічну зміну пріоритетів, автоініціалізацію, напрямок зміни адреси при обміні і т.д.).

Завантаження 16-розрядних регістрів контролера здійснюється через 8-ми розрядні порти ВВ. Перед завантаженням першого (молодшого) байта повинен бути скинутий (очищений) тригер-засувка (тригер перший/останній, First/Last flip-flop), що змінює свій стан після надходження в порт першого байта це дає можливість наступною командою виводу в той же порт завантажити старший байт відповідного регістра.

Запрограмований канал повинний бути демаскований (біт маски каналу встановлюється при цьому в 0), після чого він може приймати сигнали «Запит на ПДП», згенерований тим зовнішнім пристроєм, який обслуговується через цей канал. Сигнал «Запит на ПДП» може бути також ініційований встановленням в «1» біта запиту даного каналу в регістрі запитів контролера. Після появи сигналу запиту контролер входить в активний цикл, у якому виконується обмін даними. Обмін може здійснюватися в одному з чотирьох режимів:

### 1. Режим одиничної передачі (Signle Transfer Mode).

Після кожного циклу передачі контролер звільняє шину процесорові, але відразу ж починає перевірку сигналів запиту і як тільки виявляє активний сигнал запиту, ініціює наступний цикл передачі.

## **2. Режим блокової передачі (Block Transfer Mode).**

У цьому режимі наявність сигналу запиту потрібна тільки до моменту видачі контролером сигналу «Підтвердження запиту на ПДП» (DACK), після чого шина не звільняється аж до завершення передачі всього блоку.

## **3. Режим передачі за вимогою (Demand Transfer Mode).**

Даний режим є проміжним між двома першими: передача йде безупинно доти, поки активний сигнал запиту, стан якого перевіряється після кожного циклу передачі. Як тільки пристрій не може продовжити передачу, сигнал запиту скидається ним і контролер припиняє роботу. Цей режим застосовується для обміну з повільними пристроями, що не дозволяють по своїх часових характеристиках працювати з ПДП у режимі блокової передачі.

## **4. Каскадний режим (Cascade Mode).**

Режим дозволяє включити в підсистему ПДП більше одного контролера в тих випадках, коли недостатньо чотирьох каналів ПДП. У цьому режимі один з каналів ведучого контролера використовується для каскадування з контролером другого рівня. Для роботи в каскаді сигнал HRQ («Запит на захоплення») веденого сигналу DACK («Підтвердження запиту») ведучого подається на вхід HDLA («Підтвердження захоплення») веденого.

Така схема підключення аналогічна підключенню ведучих (першого) контролера до мікропроцесора, з яким він проводить обмін сигналами HRQ і HDLA.

### **Типи передач**

#### **1. Передача пам'ять-пам'ять (Memory-to-memory DMA).**

Використовується для передачі блоку даних з одного місця пам'яті в інше. Вхідна адреса визначається в регістрах нульового каналу, вихідна - у регістрах першого каналу. Число циклів обміну (число байт мінус 1) задається в регістрі числа циклів каналу 1. Передача відбувається з використанням робочого регістра контролера (проміжна ланка для збереження інформації). При передачі пам'ять-пам'ять може бути заданий спеціальний режим фіксації адреси (Address hold), при якому значення поточної адреси в регістрі нульового каналу не змінюється, при цьому весь вихідний блок пам'яті заповнюється тим самим елементом даних, що знаходиться по заданій адресі.

#### **2. Автоініціалізація (Автозавантаження, Autoinitialization).**

Після завершення звичайної передачі використаний канал ПДП маскується і повинен бути перепрограмований для подальшої роботи з ним. При автоініціалізації маскування каналу після закінчення передачі не відбувається, а регістри поточної адреси і лічильник циклів автоматично завантажуються з відповідних регістрів з початковими значеннями. У такий спосіб для продовження (повторення) обміну досить виставити сигнал запиту на ПДП по даному каналі.

#### **3. Режим фіксованих пріоритетів.**

У цьому режимі канал 0 завжди має максимальний пріоритет, а канал 3 - мінімальний. Це означає, що будь-яка передача по каналу з більш високим пріоритетом буде виконуватися раніш, ніж по каналу з більш низьким пріоритетом.

#### **4. Циклічне зміщення пріоритетів.**

Дозволяє уникнути «забивання» шини одним каналом при одночасній передачі по декількох каналах. Кожному каналові, по якому пройшла передача, автоматично привласнюється нижчий пріоритет, після чого право на передачу одержує канал з найвищим пріоритетом, для якого передача в даний момент можлива. Таким чином, якщо на початку роботи розподіл пріоритетів був звичайним (канал 0 - найвищий), і надійшли сигнали запиту на ПДП по 1-му і 2-му каналах, то спочатку буде виконуватися передача по першому каналу, потім він одержить нижчий пріоритет (а канал 2, відповідно, вищий, тому що зміщення пріоритетів циклічне) і передача виконається по 2-му каналі, що потім одержить нижчий пріоритет, а вищий пріоритет одержить, відповідно, канал 3, що і буде мати переважаюче право на передачу.

#### **5. Стиснення часу передачі (Compressed transfer timing).**

У випадку, якщо часові характеристики швидкодії пристроїв, що обмінюються, збігаються, ПДП може скоротити час виконання кожного такту передачі на 2 цикли годин за рахунок тактів очікування, що входять у кожен цикл передачі.

### **4.3 Блок схема КПДП**

Спрощена структурна схема КПДП приведена на рисунку 38. До складу ВІС входять: двонаправлений двостабільний буфер даних (BD), призначений для обміну інформацією між МП і КПДП; схема керування читанням/записом (RWCU), що адресує внутрішні регістри КПДП і керує обміном по шині D(7-0); блок керування (CU), що містить регістри режиму і стану КПДП та забезпечуючий послідовність операцій, необхідну для організації режиму прямого доступу до пам'яті; блок керування пріоритетами (PCU), що забезпечує визначений порядок обслуговування запитів зовнішніх пристроїв; чотири канали прямого доступу (CH0-CH3), кожен з яких містить регістр адреси комірки пам'яті, з яким відбувається обмін, і лічильник циклів обміну, два старших розряди якого відведені для задання операції обміну.

Призначення вхідних, вихідних і керуючих сигналів КПДП:

CLK - вхід для підключення тактового генератора  $F_{CLK} = 3 \text{ МГц}$ .

CS - вибір кристала. 0 CS = дозволяє роботу КПДП.

RESET - скидання. Сигнал високого рівня переводить КПДП в початковий стан, встановлюючи в нуль регістри команд, умов, тимчасового зберігання, а також встановлюючи в одиницю всі розряди маски.

READY - готовність. Вхідний сигнал, що використовується для синхронізації роботи КПДП з менш швидкодіючими пристроями.

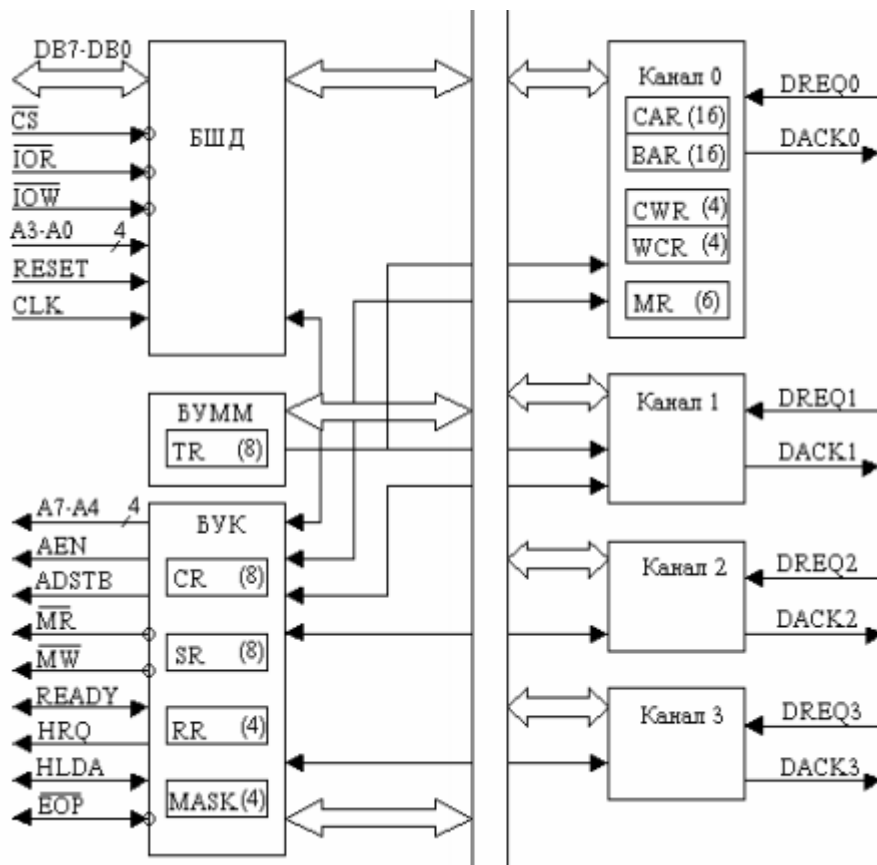


Рисунок 38 – Структурна схема КПДП

HLDA - підтвердження захоплення. Вхідний сигнал, що використовується ЦП для повідомлення КПДП про можливість виконання циклів ПДП.

DREQ3 - DREQ0 - входи запитів на ПДП від зовнішніх пристроїв. Полярність запитів задається програмно. Сигнали на цих входах повинні утримуватися до приходу сигналу DACK. У початковому стані пріоритет запитів природній, DREQ0 має найвищий пріоритет.

DB7 - DB0 - двонаправлена шина даних з буфером, що мають z-стан. У циклах ПДП на ці лінії видається вісім старших розрядів адресного коду, які необхідно «заклацнути» на зовнішньому регістрі сигналом ADSTB. У режимі роботи з ЦП по цих лініях здійснюється приймання/передача даних.

IOR - читання; як вхід використовується ЦП для читання вмісту внутрішніх регістрів КПДП; як вихід у режимі ПДП дозволяє видачу даних із зовнішніх пристроїв.

IOW - запис; як вхід використовується ЦП для завантаження даних в регістри КПДП; як вихід у режимі ПДП дозволяє запис даних в регістри зовнішніх пристроїв.

EOP - закінчення процесу. Вхід/вихід, який використовується для вказівки закінчення процесу передачі даних у режимі ПДП. Подаючи на цей вхід сигнал низького рівня, можна припинити передачу даних. Після завершення передачі даних по одному з каналів на виході встановлюється сигнал 0 EOP =. За цим сигналом (зовнішнім або внутрішнім) знімається запит, і обслуговування припиняється. Якщо встановлено режим автоініціалізації, то відбувається

завантаження робочих регістрів даного каналу вмістом базових регістрів, а розряди регістра маски не змінюються. У режимах без автоініціалізації розряди маски і розряд ТЗ у слові-стану встановлюються відповідно до стану обслуговуваного каналу. При передачі пам'ять - пам'ять вивід ЕОР орієнтований на вихід, і після закінчення рахунку на цьому виході формується сигнал. Якщо вивід ЕОР не використовується, то він повинен бути підключений через резистор до шини живлення (+5 В) для запобігання формування хибних сигналів закінчення процесу.

A3 - A0 - адресні входи/виходи. Використовуються як вхідні в режимі роботи з ЦП і для адресації до каналів і регістрів каналів КПДП. У режимі ПДП є виходами, по яких передаються чотири молодших розряду адреси ОЗП.

A7 - A4 - адресні виходи, на які в режимі ПДП передаються відповідні розряди адреси ОЗУ. У режимі роботи з ЦП переходять у z-стан.

HRQ - вихід запиту захоплення шин. Запит до ЦП для переходу в режим ПДП.

DACK3 - DACK0 - підтвердження ПДП. Вихідні лінії, на які видано або видаються повідомлення для ВК про можливість виконання циклів ПДП. Полярність сигналу задається програмно. Після сигналу RESET на виходах DACK встановлюється нуль.

AEN - дозвіл адреси. AEN = 1 встановлюється на час видачі восьми старших розрядів адреси ОЗП на лінії DB7 - DB0.

ADSTB - строб адреси. Вихід, на якому формується імпульс (строб), який здійснює запис старших розрядів (A15 - A8) адреси ОЗП з шин DB7 - DB0 у зовнішній буферний регістр.

MEMR - читання з пам'яті. Вихід, який використовується в режимі ПДП для управління операцією читання з пам'яті.

MEMW - запис в пам'ять. Вихід, який використовується в режимі ПДП для управління операцією запису в пам'ять.

Ucc - шина живлення (+5 В).

GND - загальний.

#### **4.4 Опис внутрішніх регістрів КПДП**

Контролер має 344 біта внутрішньої пам'яті, організованої у виді регістрів. Опис внутрішніх регістрів ПДП приведено в таблиці нижче.

##### **Регістр початкової адреси (Base Address Register).**

У цьому регістрі задається стартова адреса ОЗП, з якої починається передача. Регістр містить 16 розрядів і визначає адресу всередині заданої сторінки пам'яті. Задання номера сторінки пам'яті здійснюється через спеціальні сторінкові регістри (Page Registers), підтримувані зовнішньою логікою. Кожен канал ПДП має свій регістр початкової адреси і сторінковий регістр. Такий розподіл пам'яті сторінки не дозволяє здійснити обмін із блоком пам'яті, що знаходиться на перетинанні двох сторінок. Кожна сторінка починається із сегментної адреси, кратної 1000h (0, 1000h, 2000h, ..., 9000h).

Таблиця — Регістри контролера ПДП

Найменування регістра	Розрядність (біт)	Число регістрів
Регістр початкової адреси (Base Address Register)	16	4
Регістр початкового лічильника циклів (Base Word Count Register)	16	4
Регістр поточної адреси (Current Address Register)	16	4
Регістр поточного лічильника циклів (Current Word Count Register)	16	4
Робочий регістр адреси (Temporary Address Register)	16	1
Робочий регістр лічильника циклів (Temporary Word Count Register)	16	1
Регістр стану (Status Register)	8	1
Регістр команд (Command Register)	8	1
Регістр режиму (Mode Register)	6	4
Робочий регістр (Temporary Register)	8	1
Регістр масок (Mask Register)	4	1
Регістр запитів (Request Register)	4	1

#### **Регістр початкового лічильника циклів (Base Word Count Register).**

У цьому регістрі задається початкове число циклів передачі для програмованого каналу. Фактичне число переданих під час роботи ПДП елементів даних на одиницю перевищує задане число циклів, тобто якщо задається 100 циклів передачі, а розмір елемента буде дорівнювати 1-му байту, то за сеанс обміну буде переданий 101 байт інформації.

#### **Регістр поточної адреси (Current Address Register).**

Початкове значення заноситься в цей регістр одночасно з регістром початкової адреси. Надалі під час передачі значення поточної адреси автоматично збільшується або зменшується (конкретний напрямок зміни задається при програмуванні в регістрі режиму). Якщо дозволено автоініціалізацію, то після закінчення передачі в регістр автоматично встановлюється значення з регістра початкової адреси.

#### **Регістр поточного лічильника циклів (Current Word Count Register).**

Регістр містить поточне значення лічильника циклів (число циклів передачі, що залишилися). Відображуване в ньому число циклів завжди на одиницю менше числа ще не переданих елементів даних, тому що зміна значення в цьому регістрі відбувається наприкінці циклу передачі, вже після фактичної

передачі елемента даних, а кінець передачі фіксується в момент переповнення лічильника (зміна його значення з 0 на 0FFFFh).

### **Регістр режиму (Mode Register).**

Даний регістр задає режими роботи свого каналу контролера.

Кожний з чотирьох каналів ПДП має свій набір регістрів, описаних вище.

Крім того, є наступний набір регістрів, загальних для всіх каналів.

<b>D7</b>	<b>D6</b>	<b>D5</b>	<b>D4</b>	<b>D3</b>	<b>D2</b>	<b>D1</b>	<b>D0</b>
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

D1, D0 - режим роботи

0 - 0-перевірка

0 - 1-запис до пам'яті

1 - 0-читання з пам'яті

1 - 1-неможлива комбінація

D2 - Автоініціалізація: 1 - дозволена, 0 - заборонена

D3 - Зміна поточної адреси при обміні: 0 - збільшення, 1-зменшення

D4, D5 – Тип передачі:

0 – 0 - режим передачі по вимозі

0 – 1 - режим звичайної передачі

1 – 0 - режим блочної передачі

1 – 1 - каскадний режим

### **Регістр команд (Command Register).**

Цей 8-бітний регістр керує роботою контролера. Він програмується, коли контролер знаходиться в стані програмування й очищається командами скидання «Reset» і «Master Clear». Призначення бітів регістра команд приведено на наступному рисунку.

<b>D7</b>	<b>D6</b>	<b>D5</b>	<b>D4</b>	<b>D3</b>	<b>D2</b>	<b>D1</b>	<b>D0</b>
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

D0:0 - заборонити передачу пам'ять - пам'ять; 1 – дозволити

D1:0 - заборонити фіксацію адреси в каналі 0; 1 – дозволити

D2:0 - заблокувати; 1 - розблокувати контролер

D3:0 - нормальна часова діаграма; 1 – стиск в часі

D4:0 - режим фіксованих пріоритетів; 1 – режим циклічного зсуву

D5:0 - затримка при запису, якщо біт 3 встановлений ігнорується режим розширеного запису

D6:0 - активним є високий рівень сигналу запиту DREQ, 1 - низький

D7:0 - активним є високий рівень сигналу підтвердження запиту на ПДП (DACK), 1 - низький

### **Регістр стану (Status Register).**

Регістр відображує поточний стан запитів і передач по всім чотирьом каналах. Біти 0 - 3 встановлюються в одиницю після завершення передачі по каналах 0 - 3 (біт 0 - канал 0, біт 1 - канал 1 і т.д.), якщо не заданий режим автоініціалізації. Ці біти очищаються після команди скидання контролера і після кожної операції зчитування стану з регістра стану. Біти 4 – 7 указують по якому з каналів 0 - 3 активний у теперішній момент сигнал запиту на ПДП.

#### **Регістр масок (Mask Register).**

Кожен біт цього 4-бітового регістра маскує/демаскує свій канал ПДП, при цьому значення 1 маскує канал, значення 0 демаскує канал і дозволяє приймання сигналу запиту по цьому каналу.

#### **Регістр запитів (Request Register).**

Сигнал запиту на ПДП (DREQ) може бути виданий пристроєм, що обслуговується і програмно. Для програмного видання сигналу запиту по одному з 4-х каналів ПДП необхідно встановити відповідний біт у 4-х розрядному регістрі запитів. Запит на ПДП може бути відмінений записом нульового значення у відповідний біт регістра. Біт запиту очищається автоматично при закінченні передачі по даному каналу. Усі біти запиту очищаються при скиданні контролера. Для того, щоб сприймати програмні запити на ПДП, канал повинен знаходитись у режимі блокової передачі.

#### **Робочий регістр (Temporary Register).**

Цей 8-розрядний регістр використовується для збереження елемента даних, переданого в режимі фіксованої адреси при передачі пам'ять-пам'ять або для тимчасового збереження переданого байта при всіх інших режимах передачі.

### **4.5 Програмне керування контролером ПДП**

Програмне керування контролером ПДП здійснюється через порти ВВ. Доступ до кожного регістра контролера може бути здійснений через свої порти ВВ.

Порти 0h - 7h призначені для запису вихідних значень у регістри адреси, початкового і поточного лічильника циклів для всіх 4-х каналів. Порти восьмирозрядні, а регістри, у які через них заносяться дані, 16-ти розрядні і тому запис відбувається в два прийоми. Перед першою командою виводу в необхідний порт потрібно скинути тригер-засувку, для чого виконується команда виведення довільного значення в порт 0Ch, після чого в необхідний порт виводиться молодший байт 16-розрядного значення і потім старший байт наступної команди виводу в той же порт.

Часову діаграму роботи КПДП показано на рисунку 39:

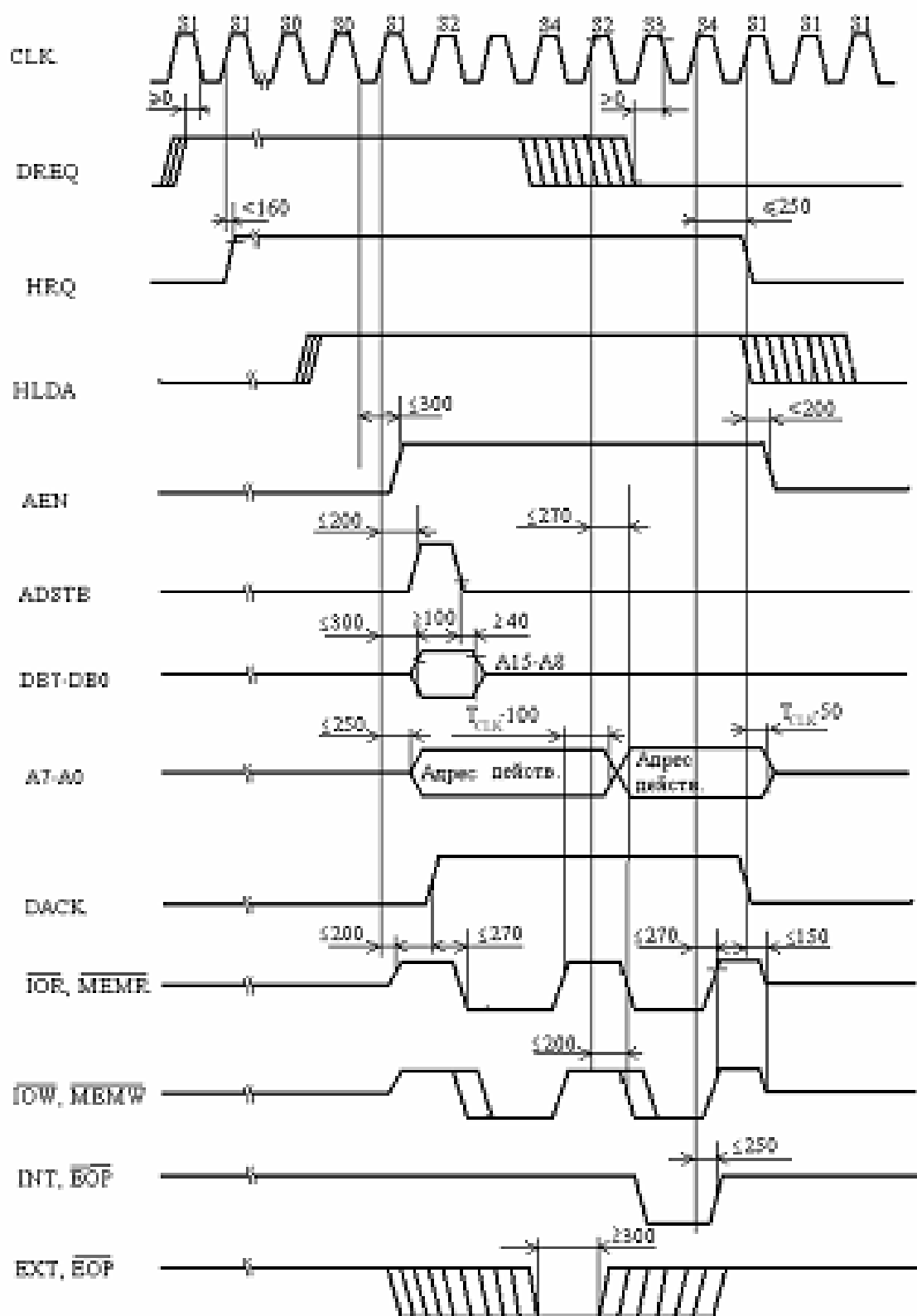


Рисунок 39 – Часова діаграма роботи КПДП

Виведення в порт 8h дозволяє занести значення в реєстр команд ПДП. Читання з порту, 8h зчитує реєстр стану ПДП. Запис у порт 9h дозволяє встановити або скинути біт запиту в реєстр запитів для одного з каналів. Формат команди наступний (рис. 40):

<b>D7</b>	<b>D6</b>	<b>D5</b>	<b>D4</b>	<b>D3</b>	<b>D2</b>	<b>D1</b>	<b>D0</b>
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

D1, D0 – вибір каналу  
 0-0 - канал 0  
 0-1 - канал 1  
 1-0 - канал 2  
 1-1 - канал 3  
 D2 - встановлення біту запиту на КПДП  
 1 - встановити  
 0 - скинути

Рисунок 40 – Формат команди порту 9h

Запис у порт 0Ah дозволяє встановити або скинути біт маски в регістрі масок для одного з каналів. Формат команди (рис.41):

<b>D7</b>	<b>D6</b>	<b>D5</b>	<b>D4</b>	<b>D3</b>	<b>D2</b>	<b>D1</b>	<b>D0</b>
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

D1,D0 – вибір каналу  
 0-0 - канал 0  
 0-1 - канал 1  
 1-0 - канал 2  
 1-1 - канал 3  
 D2 - встановлення біт маски  
 1 - встановити  
 0 - скинути

Рисунок 41 – Формат команди порту 0Ah

Запис у порт 0Bh встановлює значення в регістрі режимів одного з 4-х каналів ПДП. Біти 0 і 1 задають номер каналу (00 - 0, 01 - 1, 10 - 2, 11 - 3). У біти 2 - 7 заносяться значення, передані відповідно в біти 0 - 6 регістра режимів. Запис у порт 0Dh задає програмне скидання контролера (Master Clear). Виведення будь-якого байту в цей порт має той же ефект, що й апаратне скидання контролера. При програмному скиданні очищаються регістри команд, стану, запитів і робочий регістр. Так само скидається тригер-засувка і встановлюються всі біти масок у регістрі масок. Після програмного скидання контролер переходить у цикл чекання.

Вивід будь-якого байта в порт 0Eh очищає регістр масок - скидає біти масок усіх 4-х каналів ПДП і в такий спосіб дозволяє прийом запитів на ПДП по всіх каналах.

Через порт 0Fh можна задати довільне значення регістра масок ПДП. Для цього необхідно в бітах 0 - 3 регістра AL встановити необхідне значення масок каналів 0 - 3 відповідно і вивести це значення в порт.

Сторінкові регістри ПДП призначені для задання параметра сторінки пам'яті, з яким буде проводитись обмін. Під номером сторінки маються на увазі

старші 4 біти повної 20-бітової адреси ОЗП, тобто початкові сегментні адреси сторінок будуть кратні 1000h (0, 1000h, 2000h,..., 9000h). Потрібно відмітити, що архітектура ПДП дозволяє працювати тільки зі сторінкою пам'яті, а логіка, яка забезпечує перемикання сторінок влаштована так, що сторінки мають певні межі. Через цю особливість неможливо за допомогою ПДП здійснити обмін із блоками пам'яті, що перетинають межу між двома такими сторінками.

#### 4.6 Програмна модель КПДП. Програмування КПДП

Для початкового встановлення КПДП необхідно записати відповідну інформацію в 16-ти розрядний регістр адреси каналу (RGA), у 16-ти розрядний лічильник циклів каналу (ЗТ) і в 8-ми розрядний регістр режиму, загальний для всіх каналів. Запис цієї інформації здійснюється за допомогою команди OUT, хоча можливий і інший спосіб звертання до КПДП як до комірок пам'яті. Запис інформації в 16-ти розрядні регістри здійснюється двома командами, починаючи з молодшого байта. Два старших розряди лічильника циклів визначають операцію обміну в такий спосіб:

запис у пам'ять — 01,

читання з пам'яті — 10,

контроль — 00 (комбінація 11 заборонена).

Стан КПДП можна контролювати читанням вмісту RGA, СТ і 8-ми розрядного регістра стану, загального для всіх каналів, за допомогою команди IN. Для читання вмісту 16-ти розрядного регістра використовуються дві команди IN з однієї і тією ж адресною частиною, причому спочатку відбувається зчитування молодшого байта виводів А (3 — 0) і значення сигналу CS для адресації внутрішніх регістрів КПДП. При програмуванні КПДП операції запису керуючих або слів читання станів внутрішніх регістрів визначаються також значеннями сигналів I/OR і I/O W.

При програмуванні КПДП операції запису керуючих або слів читання станів внутрішніх регістрів визначаються також значеннями сигналів I/OR і I/O W.

В таблицях показані коди на А3-А0, що відповідають командам ЦП, і коди на А3-А0, які відповідають адресам регістрів КПДП відповідно.

A3	A2	A1	A0	Команда	Операція
1	0	0	0	Введення	Читання регістру стану
1	0	0	0	Виведення	Запис до регістру команд керування
1	0	0	1	Те саме	Запис до регістру запитів
1	0	1	0	>>	Встановлення всіх розрядів маски
1	0	1	1	>>	Запис до регістру режиму
1	1	0	0	>>	Встановлення режиму введення молодшого байта
1	1	0	1	Введення	Читання регістру тимчасового зберігання
1	1	0	1	Виведення	Загальне скидання
1	1	1	0	Те саме	Скидання всіх розрядів маски
1	1	1	1	>>	Встановлення розряду маски

A3	A2	A1	A0	Команда	Операція
0	0	0	0	Виведення	Заван. мол/ст байту в рег. BAR і CAR в к.0
0	0	0	0	Введення	Читання змісту CAR каналу 0
0	0	0	1	Виведення	Заван. мол/ст байту в рег. WCR і CWR в к.0
0	0	0	1	Введення	Читання змісту CWR каналу 0
0	0	1	0	Виведення	Заван. мол/ст байту в рег. BAR і CAR в к.1
0	0	1	0	Введення	Читання змісту CAR каналу 1
0	0	1	1	Виведення	Заван. мол/ст байту в рег. WCR і CWR в к.1
0	0	1	1	Введення	Читання змісту CWR каналу 1
0	1	0	0	Виведення	Заван. мол/ст байту в рег. BAR і CAR в к.2
0	1	0	0	Введення	Читання змісту CAR каналу 2
0	1	0	1	Виведення	Заван. мол/ст байту в рег. WCR і CWR в к.2
0	1	0	1	Введення	Читання змісту CWR каналу 2
0	1	1	0	Виведення	Заван. мол/ст байту в рег. BAR і CAR в к.3
0	1	1	0	Введення	Читання змісту CAR каналу 3
0	1	1	1	Виведення	Заван. мол/ст байту в рег. WCR і CWR в к.3
0	1	1	1	Введення	Читання змісту CWR каналу 3

Формат керуючого слова, записується в регістр режиму( Mode Register ) - 0Bh. Даний регістр задає режими роботи каналу контролера.

Розряди команди записаної в регістр команд (Command Register)-08h D3-DO(EN3 - EN(J) задають дозвіл обміну по відповідному каналу, запис нуля в розряд забороняє обмін. Інші розряди визначають режими роботи каналу.

Розряд D4(RP) встановлює порядок обслуговування запитів від каналів. При RP = 0 задається фіксований пріоритет каналів і канал 0 має вищий пріоритет. У режимі циклічного пріоритету (RP= 1) після обслуговування каналу йому привласнюється нижчий пріоритет, а наступному за ним по номеру каналу — вищий. Причому циклічне зміщення пріоритетів здійснюється після кожного циклу прямого доступу.

Режим розширеного запису (EW=i) збільшує за рахунок зсуву переднього фронту тривалість сигналів 1/OW і MEMW, які генерує КПДП. Це дозволяє ВП, що формує сигнал READY по фронті сигналу запису, зменшити час охолодження і збільшити швидкість обміну.

Якщо  $TCS=1$ , то поява сигналу  $TC$  в одному з каналів скидає відповідний розряд  $D3$  —  $D0$ , у результаті чого канал відключається. Подальша робота цього каналу можлива після перезавантаження регістра режиму. Якщо  $TCS=0$ , то поява сигналу  $TC$  не впливає на розряд дозволу роботи каналу і закінчувати передачу повинне ВП за рахунок припинення генерування сигналу  $DRQ$ .

У режимі автозавантаження ( $AL=1$ ) може працювати тільки другий канал, використовуючи вміст своїх внутрішніх регістрів і внутрішніх регістрів третього каналу. Після передачі даних відповідно до параметрів регістрів другого каналу і появи сигналу  $TC$  параметри з регістрів третього каналу автоматично завантажуються в регістри другого каналу; прапор відновлення ( $UF$ ) у регістрі стану каналів встановлюється в 1. Потім передача даних продовжується відповідно до нових параметрів регістрів другого каналу, а наприкінці першого ж циклу прямого доступу з новими параметрами прапор  $UF$  скидається. Режим автозавантаження дозволяє організувати повторювані пересилання блоків даних з однаковими параметрами або з'єднати кілька блоків з різними параметрами.

У початковому стані  $S1$  запрограмований на визначений режим КПДП очікує запиту  $DRQ$  від ВП. Переходячи в стан  $S0$ , він генерує сигнал  $HRQ$  і очікує надходження від МП сигналу  $HLDA$ . Після надходження сигналу підтвердження  $HLDA$  починається цикл обміну. У стані  $S1$  формується сигнал  $AEN$  для блокування інших пристроїв системи від шин даних і керування, видається код молодших розрядів на виходи  $A(7—0)$ , а код старших розрядів — на виходи  $D(7—0)$ .

Видача старших розрядів адреси супроводжується стробуючим сигналом  $ADSTB$  для запису їх у зовнішній буферний регістр. У стані  $S2$  генерується сигнал  $DACK$ , що вказує на початок обміну, а також формуються пари сигналів  $MEMR, I/OR$  і  $MEMW, I/OW$ , що визначають напрямок обміну. У стані  $S3$  відбувається передача даних у ЗП або ВП.

Стан  $S4$  завершує цикл прямого доступу. У цьому стані при передачі останнього байта видається сигнал  $TC$ , а у випадку кінця блоку — сигнал  $MARK$ . При необхідності узгодження швидкодії ЗП і ВП за допомогою сигналу  $READ Y$  між станами  $S_3$  і  $S_4$  вводиться необхідне число станів очікування  $SW$ . У режимі контролю перехід у стан  $SW$  не дозволяється.

Часова діаграма КПДП в циклі запису представлена на рисунку 42.

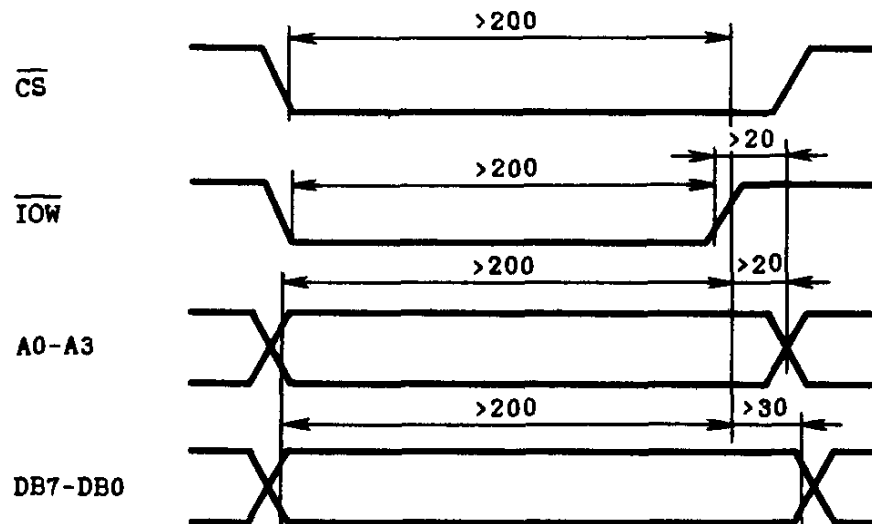


Рисунок 42 – Часова діаграма КПДП в циклі запису

В початковому стані S1 запрограмований на визначений режим КПДП очікує запиту DRQ від ВП. Переходячи в стан S0, він генерує сигнал HRQ і очікує надходження від МП сигналу HLDA. Після надходження сигналу підтвердження HLDA починається цикл обміну. У стані S1 формується сигнал AEN для блокування інших пристроїв системи від шин даних і керування, видається код молодших розрядів на виходи A (7 — 0), а код старших розрядів — на виходи D(7 — 0).

Часова діаграма КПДП в циклі читання представлена на рисунку 43.

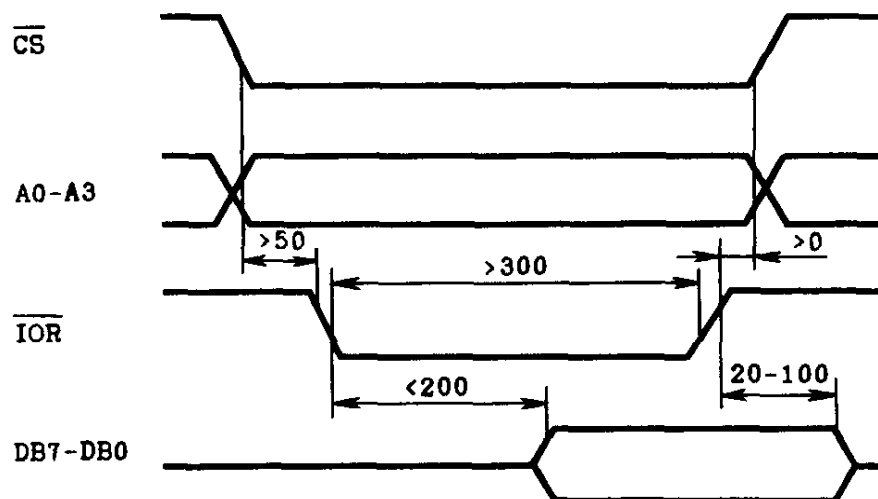


Рисунок 43 – Часова діаграма роботи КПДП в циклі читання

Видача старших розрядів адреси супроводжується стробуючим сигналом ADSTB для запису їх у зовнішній буферний регістр. У стані S2 генерується сигнал DACK, що вказує на початок обміну, а також формуються пари сигналів MEMR, I/OR і MEMW, I/OW, що визначають напрямок обміну. У стані S3

відбувається передача даних у ЗП або ВП. Стан S4 завершує цикл прямого доступу. У цьому стані при передачі останнього байта видається сигнал TC, а у випадку кінця блоку — сигнал MARK. При необхідності узгодження швидкодії ЗП і ВП за допомогою сигналу READ Y між станами S<sub>3</sub> і S<sub>4</sub> вводиться необхідне число станів очікування SW. У режимі контролю перехід у стан SW не дозволяється. Часова діаграма роботи КПДП в циклах обміну представлена на рисунку 44:

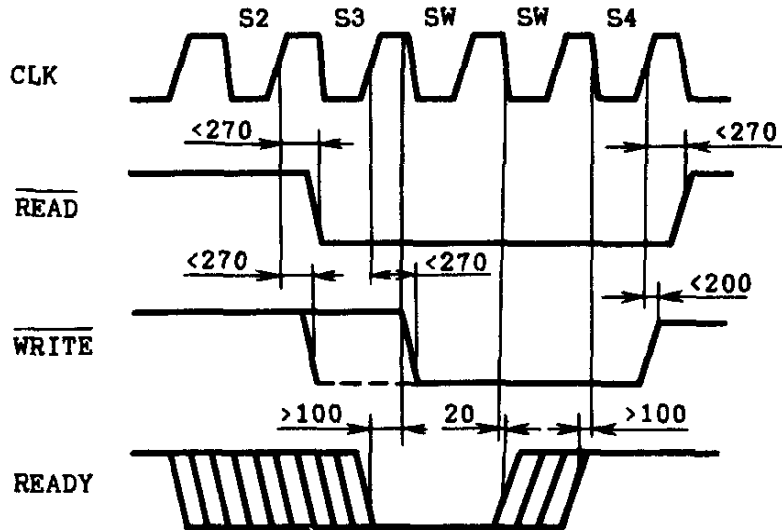


Рисунок 44 – Часова діаграма роботи КПДП в циклах обміну

Підключення КПДП до системної шини показано на рисунку 45:

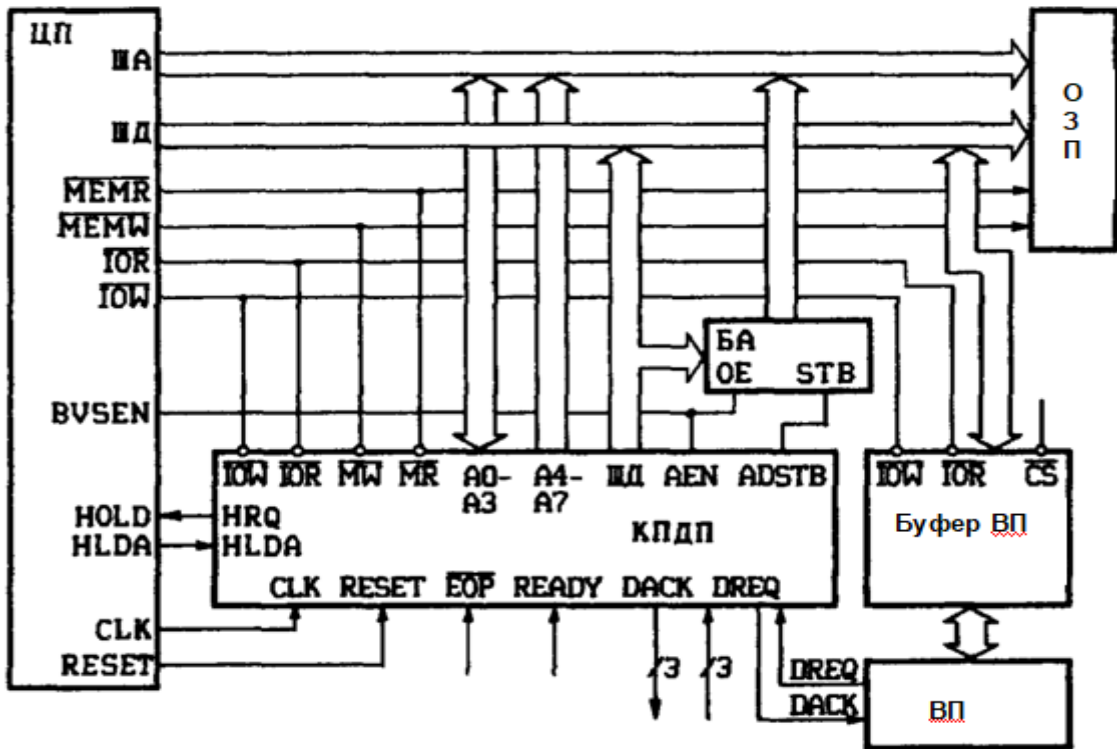


Рисунок 45 – Підключення КПДП до системної шини

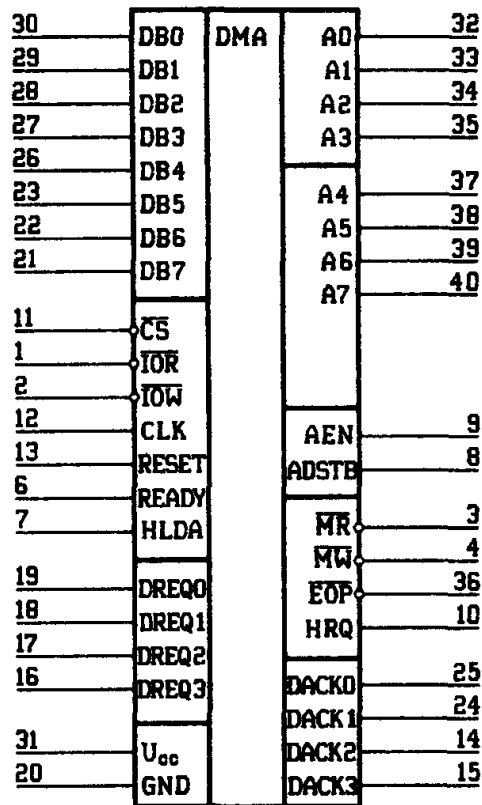


Рисунок 46 – Мікросхема КПДП

Фрагмент програми початкової встановлення КПДП при звертанні до нього як до зовнішнього пристрою може мати наступний вид:

```

“      ;
“      ;
“      ;
DI      ;заборона переривань
MVI A,28H ; завантаження керуючого слова
OUT 0BH  ; у реєстр режиму
MVI A,1AH ; завантаження керуючого слова
OUT 08H  ; у реєстр команд
MVI A,00H ; завантаження молодшого байта адреси
OUT 00H(0кан.) ; масиву в порт
      02H(1кан.)
      04H(2кан.)
      06H(3кан.)
MVI A,40H ; завантаження старшого байта адреси
OUT 00H(0кан.) ; масиву в порт
MVI A,01H ; завантаження номера сторінки пам'яті
OUT 81H(2кан.) ; в порт каналу
      82H(3кан.)
      83H(1кан.)
      87H(0кан.)
MVI A,F4H ; завантаження молодшого байта лічильника циклів

```

```

OUT  01H(0кан.)   ; в порт канала
      03H(1кан.)
      05H(2кан.)
      07H(3кан.)
MVI  A,00H        ; завантаження старшого байта лічильника циклів
OUT  01H          ; в порт канала
EI    ;дозвіл переривань

```

У програмі відбувається налаштування каналу на запис масиву в область ОЗП. Для передачі N байт у лічильник каналу завантажуються число N-1. Після програмного налаштування каналу аналогічно набудовуються інші канали. У регістр режиму завантажуються керуюче слово. Для попередження помилок перед програмуванням КПДП або читанням вмісту його регістрів необхідно заборонити переривання.

Підключення контролера до системної шини (рис. 45). Вісім старших розрядів адреси видаються на ШД і повинні бути записані сигналом ADSTB в зовнішній регістр. Лінія AEN використовується для того, щоб розряди адреси залишилися чинними на ША протягом трьох тактових періодів циклу ПДП. Лінії A7 - A0 підключаються безпосередньо до ША. Сигнали MEMR, MEMW, IOR, IOW керують у циклах ПДП відповідно ОЗП і буфером ВП.

За допомогою емулятора програмованого контролера прямого доступу до пам'яті I8257 прослідкуємо процес програмування пристрою. Загальний вигляд емулятора представлений на рис.47 (em dma, prj\_dma.exe).

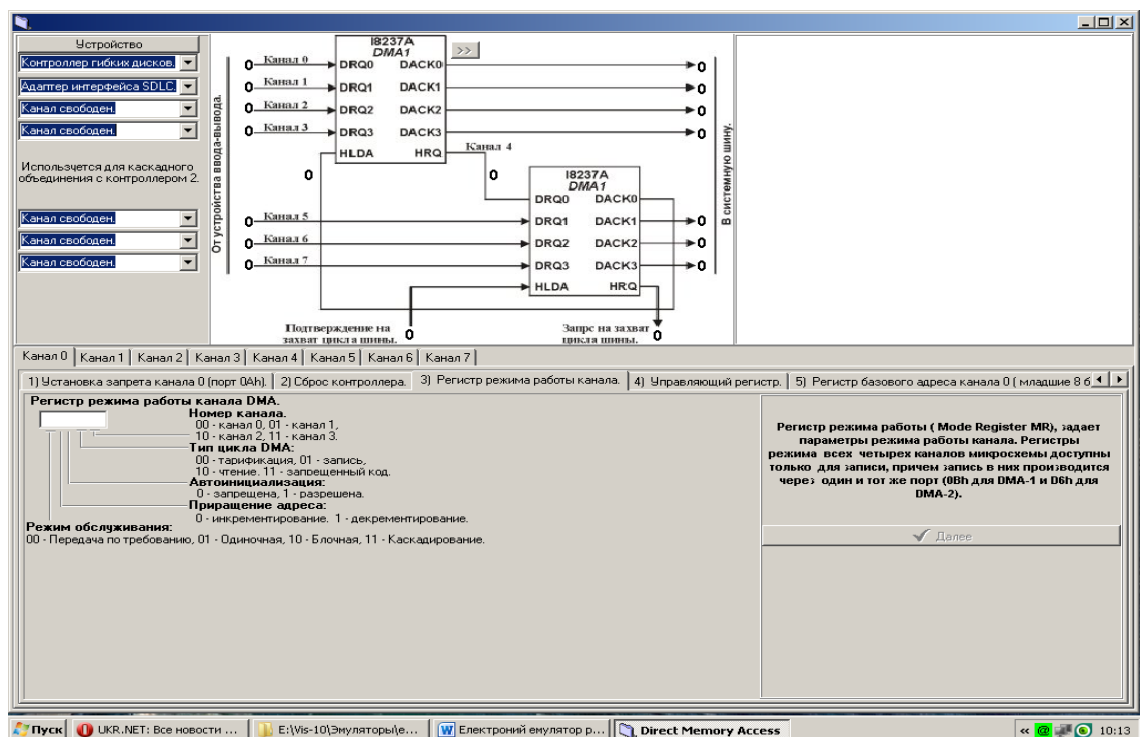


Рисунок 47 - Загальний вигляд емулятора контролера I8257

## **Завантаження програми до емулятора**

1. Для початку роботи емулятора необхідно завантажити послідовно всі керуючі слова вибраного режиму роботи . Спочатку виберемо один з доступних каналів одного з задіяних контролерів. У інформаційних віконцях з першого по десяте вказується значення керуючого слова.

2. Натиснемо кнопку 1(завантаження першого керуючого слова). У вікні зліва пояснюється значення кожного біту керуючого слова, а у вікні справа дається пояснення подій, які відбудуться після введення набраної команди. Задамо команду, якщо буде введений неправильний формат команди , з'явиться вікно «Внимание ошибка». Перевірте формат команди і відредагуйте у відповідності з підказкою. Натисніть кнопку «Далее», емулятор автоматично перейде до завантаження наступного керуючого слова 2 і так далі поки не завантажимо усі керуючі слова.

3. Після завантаження керуючих слів декілька разів натиснемо кнопку «>>». У вікні відобразиться робота контролера, а потім з'являться інформаційні вікна «Данные устройства (DRQ)», з кнопками «Ввод» та «Скрыть окно ввода данных» та «Регистры канала контроллера DMA» з кнопками «Повтор», «Пауза», «Выход». У вікні «Данные устройства (DRQ)» відобразиться процес читання або запису даних по вказаним адресам а у вікні «Регистры канала контроллера DMA» значення відповідних регістрів. Вихід з режиму виконується за допомогою відповідних кнопок зупинки процесу.

## Контрольні питання та завдання

1. У яких режимах працюють КПДП і його функції в системі?
2. Визначте програмно-доступні регістри і їх адресацію.
3. Складіть схему підключення I8257 до шин адреси і даних МПС.
4. Складіть програму ініціалізації контролера для блочного обміну по одному каналу.
5. Поясніть реалізацію режиму ПДП на I8257.
6. Поясніть адресацію до регістрів КПДП при програмуванні.
7. Як формується 16-бітна адреса КПДП при управлінні обміном?
8. Які пріоритети запитів підтримує I8257 ?
9. В якій послідовності необхідно робити завантаження регістрів контролера при його програмуванні?
10. На лабораторній використовуючи емулятор КПДП написати програму передачі або приймання даних до пам'яті відповідно варіанту і відлагодити її.

## Лекція 5

### 5.1 Програмований інтервальний таймер KP580BI53

#### Принцип роботи таймера

БІС програмованого таймера KP580BI53 призначена для організації роботи мікропроцесорних систем у режимі реального часу і дозволяє формувати сигнали з різними часовими і частотними характеристиками.

Мікросхема являє собою однокристальний трьохканальний програмований пристрій, призначений для отримання програмно-керованих часових затримок і виконання заданих часових функцій в мікропроцесорних системах. БІС містить три канали. В кожному каналі є регістр керуючого слова, 16-ти розрядний програмований і працюючий у двійковому або двійково-десятковому коді (реверсивний), 2-х байтний буферний регістр, в який по спеціальній команді переписується код лічильника.

CLK - тактові входи лічильника;

GATE - входи блокування роботи;

OUT - виходи лічильника;

RD/ - читання лічильника;

WR/ - запис команди або даних;

CS/ - вибір корпусу мікросхеми;

A0, A1 - вибір лічильника;

Програмування кожного каналу таймера зводиться до наступних операцій:

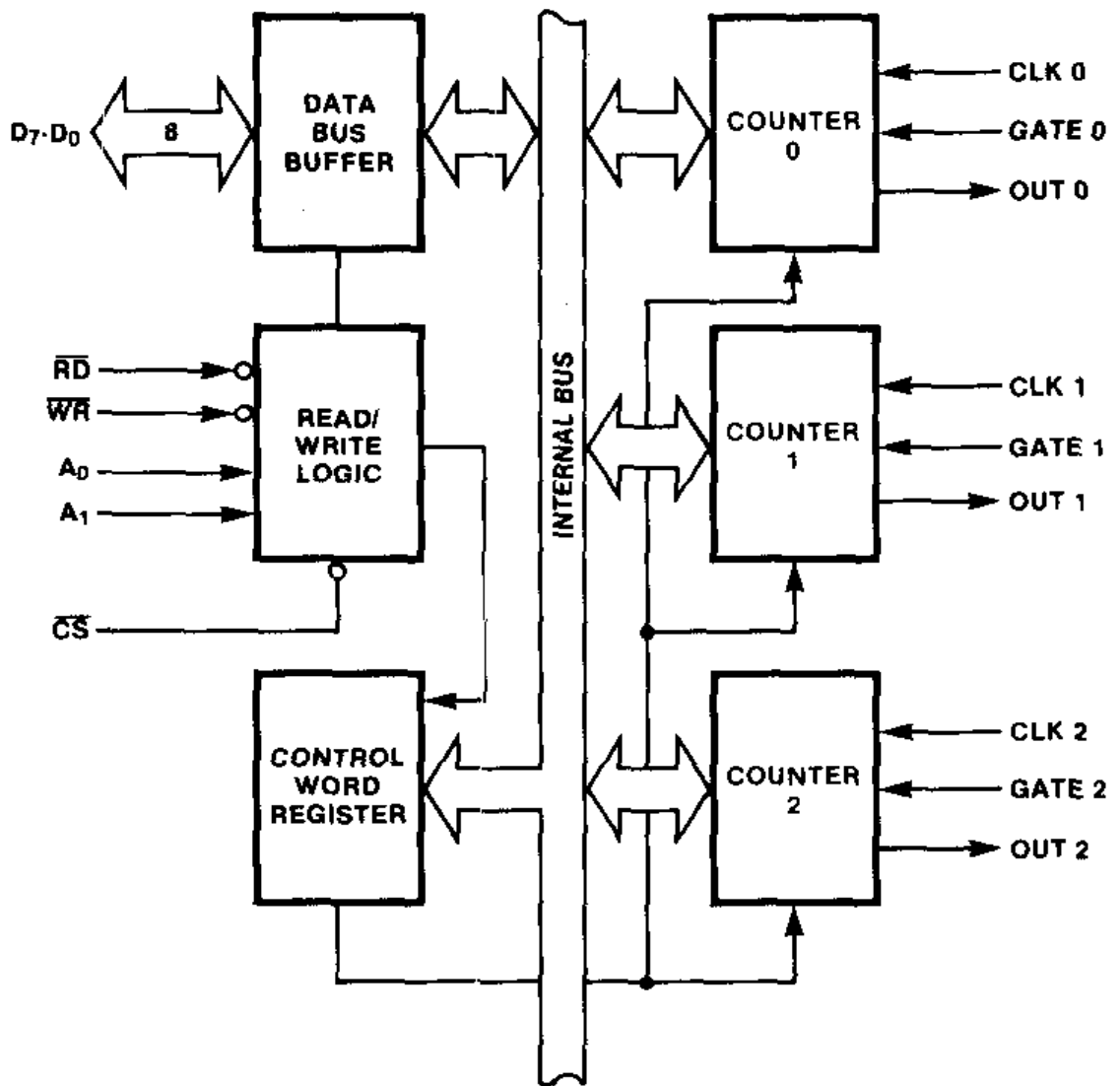
- запис у регістр керуючого слова індивідуального слова керування каналом;
- запис в 16-й програмований лічильник необхідного коду перерахунку;
- запис індивідуального слова керування каналом здійснюється по єдиній для усіх каналів адресі (A0=1, A1=1);

#### 5.2 Блок схема таймера

Спрощена структурна схема таймера приведена на рисунку 45. До складу БІС входять:

- буфер даних (BD), призначений для обміну даними і керуючими словами між МП і таймером;
- схема керування читанням-записом (RWCU), що забезпечує виконання операцій ВВ інформації в таймер;
- регістр керуючого слова (RGR), призначений для запису керуючих слів, що задають режими роботи лічильників;
- лічильник каналів(СТ0-СТ2).

Таймер містить три незалежних канали. У кожному каналі є регістр слова керування, 16-ти розрядний програмований лічильник, що працює в двійковому або двійково-десятковому коді, двобайтний буферний регістр, через який контакти регістра і команди управління зв'язані із системною шиною.



DBB - буфер даних  
 RWL - схема керування читанням-записом  
 CWR - регістр керуючого слова  
 CT0-CT2 – лічильник каналів

Рисунок 48 - Спрощена структурна схема таймера

Кожен канал містить наступні регістри:

- стану каналу RS (8 розрядів);
- керуючого слова RSW (8 розрядів);
- буферний регістр OL (16 розрядів);
- регістр лічильника CE (16 розрядів);
- регістр констант перерахунку CR (16 розрядів).

Канали таймера підключаються до зовнішніх пристроїв за допомогою трьох ліній:

GATE - керуючий вхід;

CLK - вхід тактової частоти;

OUT - вихід таймера.

Регістр лічильника CE працює в режимі рахунку. Його вміст зменшується по задньому фронті сигналу CLK за умови, що на вході GATE встановлений рівень логічної 1. У залежності від режиму роботи таймера при досягненні лічильником CE нуля тим або іншим способом змінюється вихідний сигнал OUT.

Буферний регістр OL призначений для запам'ятовування поточного вмісту регістра лічильника CE без зупинки процесу рахунку. Після запам'ятовування буферний регістр доступний програмі для читання.

Регістр констант перерахунку CR може завантажуватися в регістр лічильника, якщо це потрібно в поточному режимі роботи таймера. Регістри стану каналу і керуючого слова призначені, відповідно, для визначення поточного стану каналу і для задання режиму роботи таймера.

Вибір необхідного лічильника здійснюється за допомогою адрес  $A_0 \dots A_1$  по наступній таблиці.

A0	A1	Найменування
0	0	Лічильник каналу 0
0	1	Лічильник каналу 1
1	0	Лічильник каналу 2
1	1	Регістр керуючого слова

На вхід таймера надходять сигнали: GATE (для блокування роботи лічильника) і імпульси заданої частини CLK. На виході кожного каналу таймера є сигнал OUT, що по закінченню перерахунку ( $const=0$ ) змінює своє положення з «0» у «1» або з «1» у «0» у залежності від заданого режиму таймера.

Підключення таймера до шин мікропроцесора показано на рисунку 46:

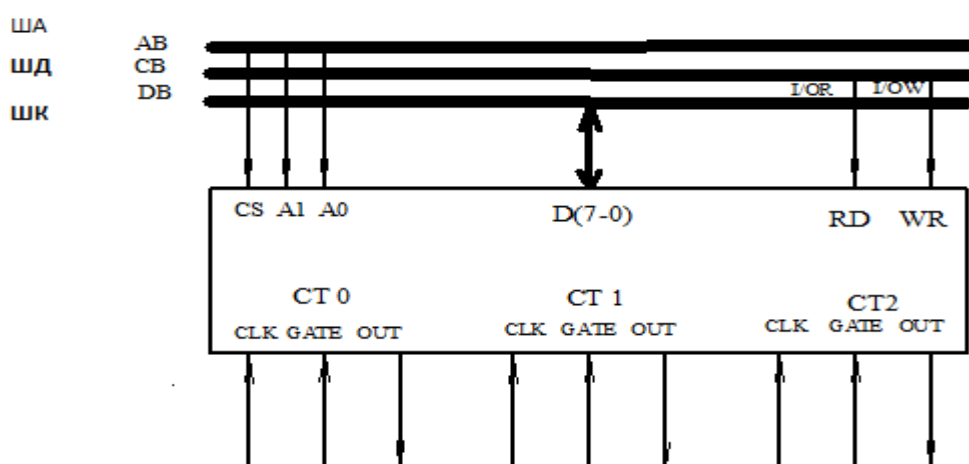


Рисунок 49 - Підключення таймера до системних шин

Умовне графічне зображення таймера представлено на рисунку 50:

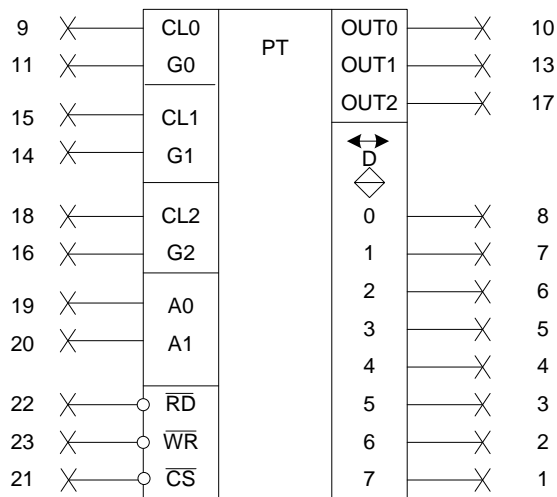


Рисунок 50 - Графічне зображення таймера

### 5.3 Режими роботи таймера

Таймер має шість режимів роботи:

- режим «нуль» - формування сигналу по закінченню інтервалу часу;
- режим «1» формування імпульсу заданої тривалості;
- режим «2» генератор імпульсі потрібної частоти;
- режим «3» програмований генератор прямокутних імпульсів;
- режим «4» формування імпульсу по закінченню заданого часу;
- режим «5» лічильник подій.

ВІС I8253/54 містить три 16-ти розрядних двійково-десяткових регістри з попереднім заповненням. Функції конфігурації його входів і виходів встановлюються за допомогою засобів програмного забезпечення.

Лічильники повністю незалежні. Кожен може мати свій режим роботи і тип лічби (двійковий або десятковий). Завантаження лічильників початковим значенням лічби визначається програмно. Зчитування змісту кожного лічильника здійснюється за допомогою команди введення.

Таймер також містить додаткове обладнання і команду, які дозволяють читати вміст лічильника "на льоту", не заважаючи його роботі. Програмування таймера відбувається за допомогою команди виводу. Кожен лічильник індивідуально налагоджується відповідно вмісту регістра керуючого слова.

Усі шість режимів роботи таймера можна розділити на три групи:

**Режими 0, 4** – режими одноразового виконання функцій. Якщо під час рахунку на вхід GATE подати нуль, то це приведе до зупинки рахунку. Новий позитивний сигнал на GATE не викликає перезавантаження лічильника / таймера, а тільки дозволяє продовження рахунку. По необхідності повторення функції необхідно нове програмування-завантаження нової константи.

GATE 1 – дозвіл

Перехід GATE з 1 в 0 - зупиняє рахунок

Перехід GATE 0 в 1 - дозволяє продовження рахунку.

**Режими 1, 5** - режими з перезапуском. Тут є можливість повторення запрограмованих функцій без нового перепрограмування. Завантажена константа зберігається, а її передача здійснюється по фронту сигналу GATE незалежно від завершення рахунку.

GATE 1 – дозвіл

Перехід GATE з 1 в 0 - дозвіл

Перехід GATE з 0 в 1 - перезавантаження.

**Режими 2, 3** – режими автозавантаження. Завантаження здійснюються автоматично при виконанні умов рахунку (імпульсний генератор і генератор меандру), оскільки це режими з зацикленням рахунку. Вихід OUT відкривається позитивним сигналом на GATA.

GATE 1 – дозвіл

GATE 0 – зупинка

Перехід GATE з 0 в 1 - перезавантаження.

Розглянемо роботу таймера в різних режимах.

### **Режим 0. Формування сигналу по закінченню заданого інтервалу часу**

Після запису за допомогою сигналу /WR керуючого байта CWR, який переводить відповідний канал таймера у режим 0 вихід таймера OUT встановлює значення логічної «1».

Режим передбачає зміну виходу таймера OUT з низького рівня  $OUT=0$  до високого  $OUT=1$  після закінчення лічби (досягнення значення 0 константи перерахунку).

Після запису за допомогою сигналу /WR керуючого байта CWR, який переводить канал N таймера у режим 0, вихід OUT встановлюється в 0. Після запису константи перерахунку (наприклад,  $N=4$ ), канал N починає роботу.

Коли константа перерахунку досягне нульового значення, вихід таймера встановлюється у високий рівень  $OUT=1$ .

Високий рівень входу  $GATE=1$  дозволяє роботу лічильника, а  $GATE=0$  – зупиняє його роботу. Таким чином, час видачі сигналу по закінченню лічби залежить від введеної константи перерахунку і може збільшуватись на тривалість  $\Delta t$  за допомогою сигналу керування GATE.

GATE 1 – дозв.

Перехід GATE з 1 в 0 - зупиняє рахунок

Перехід GATE 0 в 1 - дозволяє продовження рахунку.

### **Режим 1. Формування імпульсу заданої тривалості (строб)**

Після запису за допомогою сигналу /WR керуючого байта CWR, який переводить відповідний канал таймера у режим 1. Вихід таймера OUT встановлюється в 1. Після завантаження константи перерахунку ( $N=4$ ) таймер починає роботу і вихід таймера встановлюється у низький рівень  $OUT=0$ . Коли константа перерахунку досягне нульового значення, вихід таймера встановлюється у високий рівень  $OUT=1$ . Таким чином формується інтервал часу

$\Delta t$  (строб в залежності від константи перерахунку). Високий рівень входу  $GATE=1$ , а також перехід в 0 і його утримання, дозволяє роботу лічильника, а перехід сигналу  $GATE$  з низького рівня у високий, призводить до перезавантаження константи перерахунку лічильника на початкове значення, що дозволяє змінювати інтервал стробу.

$GATE=1$  – дозвіл

Перехід  $GATE$  з 1 в 0 - дозвіл

Перехід  $GATE$  з 0 в 1 - перезавантаження.

### **Режим 2. Програмований генератор імпульсів потрібної частоти**

Після запису за допомогою сигналу  $/WR$  керуючого байта  $CWR$ , який переводить відповідний канал таймера у режим 2. Вихід таймера  $OUT$  встановлюється в 1.

Після завантаження константи перерахунку ( $N=4$ ) таймер починає роботу. Коли константа перерахунку досягне значення 1, вихід таймера встановлюється у низький рівень  $OUT=0$  і тримається у цьому стані доки константа перерахунку досягне нульового значення, після чого вихід таймера приймає значення 1 ( $OUT=1$ ) і відбувається апаратне перезавантаження константи перерахунку на початкове значення  $N=4$ . Високий рівень  $GATE=1$ , а також перехід сигналу  $GATE$  з 1 в 0 і його утримання, дозволяє роботу таймера, а перехід сигналу  $GATE$  з низького рівня у високий, призводить до перезавантаження константи перерахунку лічильника на початкове значення, що дозволяє змінювати частоту імпульсів.

$GATE=1$  – дозвіл

$GATE=0$  – зупинка.

Перехід  $GATE$  з 0 в 1 - перезавантаження.

### **Режим 3. Програмований генератор прямокутних імпульсів**

Після запису керуючого байта  $CWR$ , відповідний канал таймера налаштовується на режим 3. Вихід таймера  $OUT$  встановлюється в 1. Після завантаження константи перерахунку таймер починає роботу і рівень логічної одиниці на виході таймера  $OUT=1$  буде триматися до половини значення константи перерахунку  $N/2$  (якщо  $N$  парне), після цього встановлюється логічний 0, який утримується до досягнення нульового значення константи перерахунку  $N$ , після чого відбувається автоматичне перезавантаження константи перерахунку на початкове значення і таймер починає свою роботу спочатку. Таким чином на виході таймера формується сигнал прямокутної форми, в якому тривалість логічного нуля дорівнює тривалості логічної одиниці (меандр). Якщо число  $N$  непарне то на виході таймера логічна одиниця буде протягом  $(n+1)/2$  і логічний нуль протягом  $(n-1)/2$ .

Високий рівень  $GATE=1$  дозволяє роботу таймера, низький рівень  $GATE=0$  зупиняє, а перехід сигналу  $GATE$  з 0 в 1 перезавантажує константу перерахунку на початкове значення і дозволяє роботу таймера.

$GATE=1$  – дозвіл.

GATE 0 – зупинка.

Перехід GATE з 0 в 1 - перезавантаження.

#### **Режим 4. Формування імпульсу по закінченню заданого часу**

Після запису за допомогою сигналу /WR керуючого байта CWR, який переводить відповідний канал таймера у режим 4. Вихід таймера OUT встановлюється в 1.

Після запису константи перерахунку ( $N=4$ ) таймер починає роботу.

При досягненні константи  $N=0$  на один період CLK стан виходу таймера OUT встановлюється в 0, після чого OUT знову встановлюється в 1 ( тобто через період часу  $(n+1) T_{clk}$ )  $OUT=0$  на період одного такту.

Високий рівень входу  $GATE=1$  дозволяє роботу лічильника,  $GATE=0$  – зупиняє його роботу, а перехід сигналу GATE з низького рівня у високий, призводить до продовження рахунку.

GATE 1 – дозв.

Перехід GATE з 1 в 0 - зупиняє рахунок

Перехід GATE 0 в 1 - дозволяє продовження рахунку.

#### **Режим 5. Лічильник подій**

Після запису за допомогою сигналу /WR керуючого байта CWR, який переводить відповідний канал таймера у режим 5. Вихід таймера OUT встановлюється в 1. Після запису константи перерахунку ( $N=4$ ) таймер починає роботу. При досягненні константи  $N=0$  на один період CLK стан виходу таймера OUT встановлюється в 0, після чого OUT знову встановлюється в 1 ( тобто через період часу  $(n+1) T_{clk}$ )  $OUT=0$  на період одного такту.

Високий рівень входу  $GATE=1$ , а також перехід в 0 і його утримання, дозволяє роботу лічильника, а перехід сигналу GATE з низького рівня у високий, призводить до перезавантаження константи перерахунку лічильника на початкове значення, що дозволяє змінювати інтервал стробу.

GATE 1 –дозв.

Перехід GATE з 1 в 0 - дозвіл

Перехід GATE з 0 в 1 - перезавантаження.

Керуючі слова можуть бути записані в програмований таймер у довільному порядку. У будь-який наступний момент часу записуються початкові коди лічильників у відповідності зі значеннями розрядів D5, D4 керуючих слів.

Виходячи з особливості впливу сигналу GATE на роботу лічильника каналу інтервального таймера, режими його роботи можна поділити на дві групи:

- з при зупинкою (паузою) по сигналу GATE (режим 0, 3, 4);
- з перезавантаженням по сигналу GATE (режим 1, 2, 5).

Часову діаграму роботи таймера в режимі 0 представлено на рисунку 51:

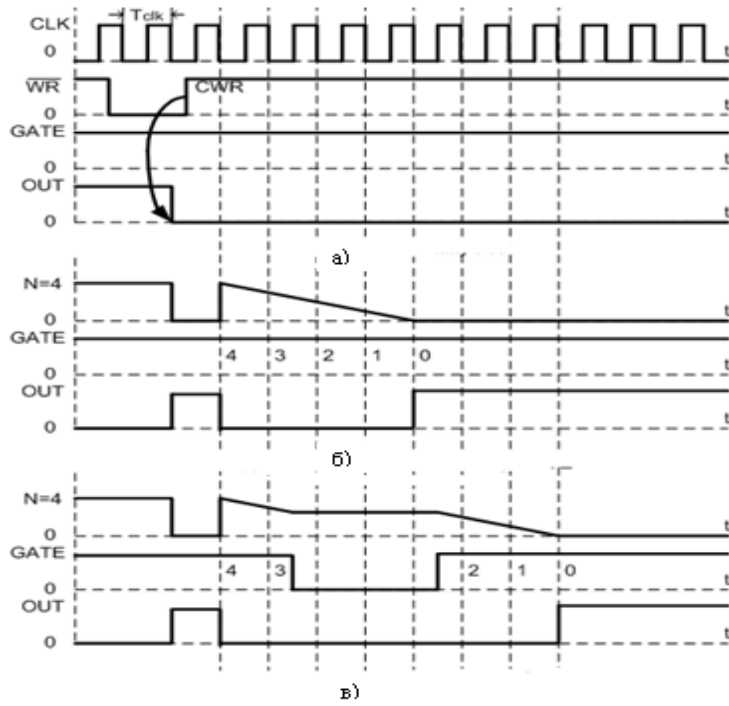


Рисунок 51 – Часова діаграма роботи таймера в режимі 0  
 а) завантаження керуючого байта;  
 б) робота таймера з N=4, GATE=1;  
 в) робота таймера при зміні GATE.

Часову діаграму роботи таймера в режимі 1 представлено на рисунку 52:

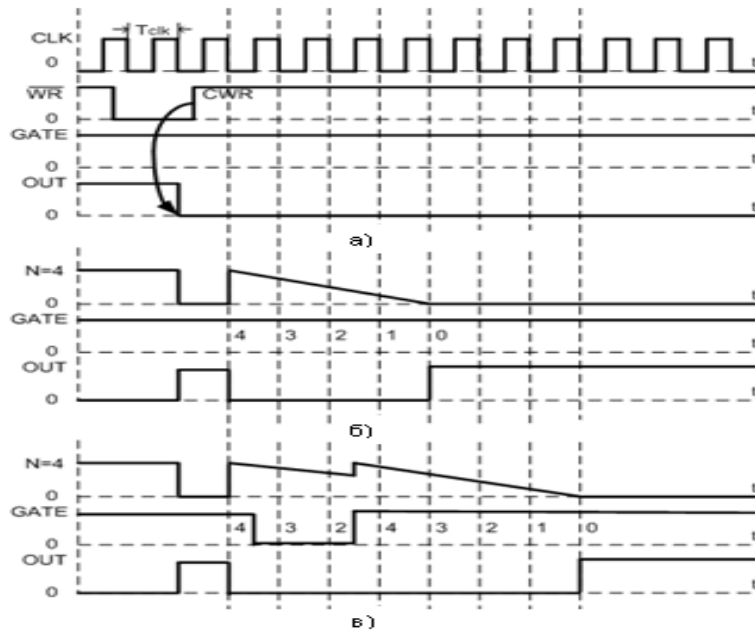


Рисунок 52 – Часова діаграма роботи таймера в режимі 1  
 а) завантаження керуючого байта;  
 б) робота таймера з N=4, GATE=1;  
 в) робота таймера при зміні GATE.

Часову діаграму роботи таймера в режимі 2 представлено на рисунку 53:

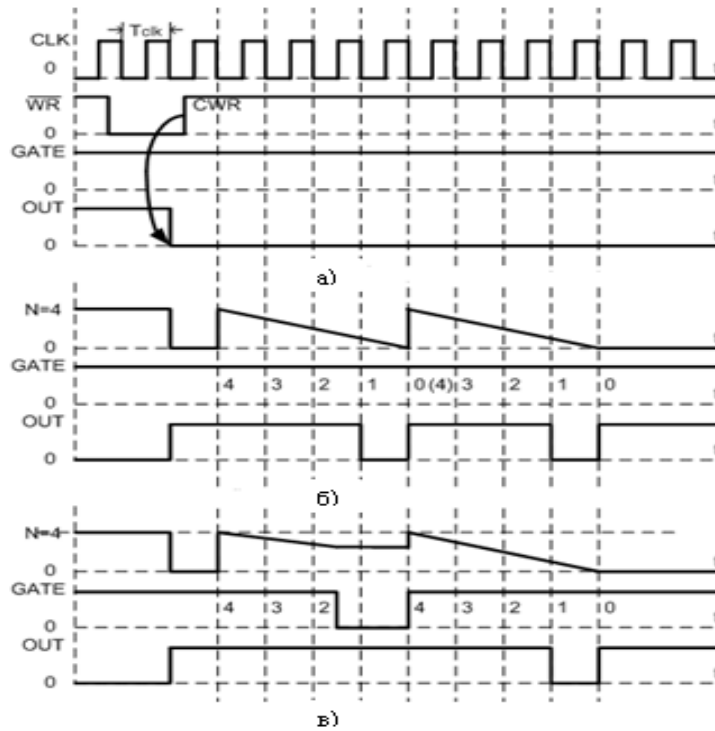


Рисунок 53 – Часова діаграма роботи таймера в режимі 2  
 а) завантаження керуючого байта;  
 б) робота таймера з  $N=4$ ,  $GATE=1$ ;  
 в) робота таймера при зміні  $GATE$ .

Часову діаграму роботи таймера в режимі 3 представлено на рисунку 54:

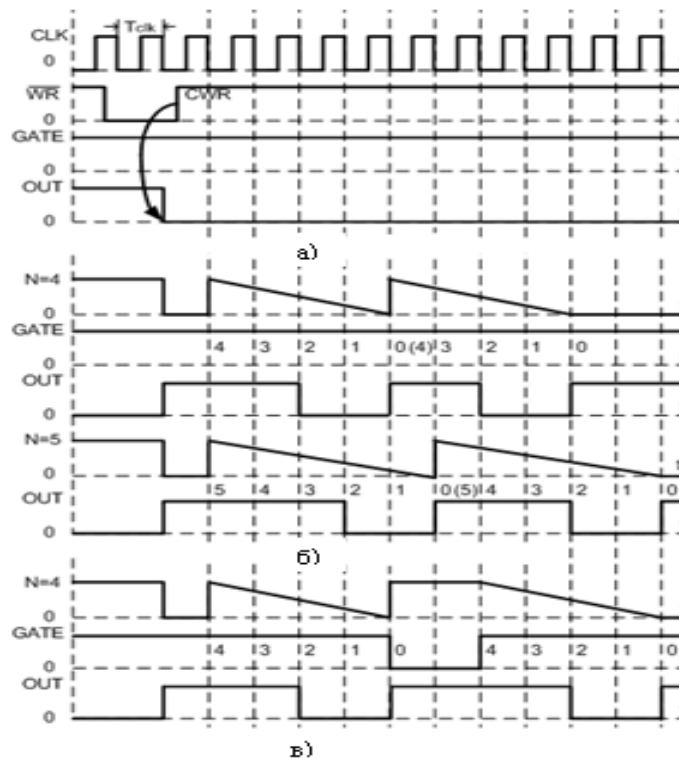


Рисунок 54 – Часова діаграма роботи таймера в режимі 3  
 а) завантаження керуючого байта;  
 б) робота таймера з  $N=4$ ,  $GATE=1$ ;  
 в) робота таймера при зміні  $GATE$ .

Часову діаграму роботи таймера в режимі 4 представлено на рисунку 55:

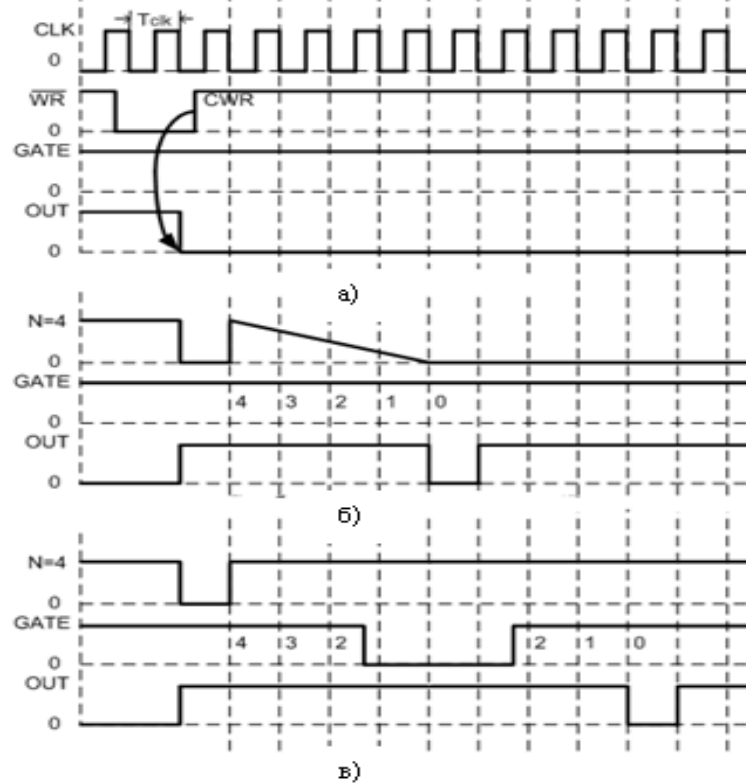


Рисунок 55 – Часова діаграма роботи таймера в режимі 4  
 а) завантаження керуючого байта;  
 б) робота таймера з  $N=4$ ,  $GATE=1$ ;  
 в) робота таймера при зміні  $GATE$ .

Часову діаграму роботи таймера в режимі 5 представлено на рисунку 56:

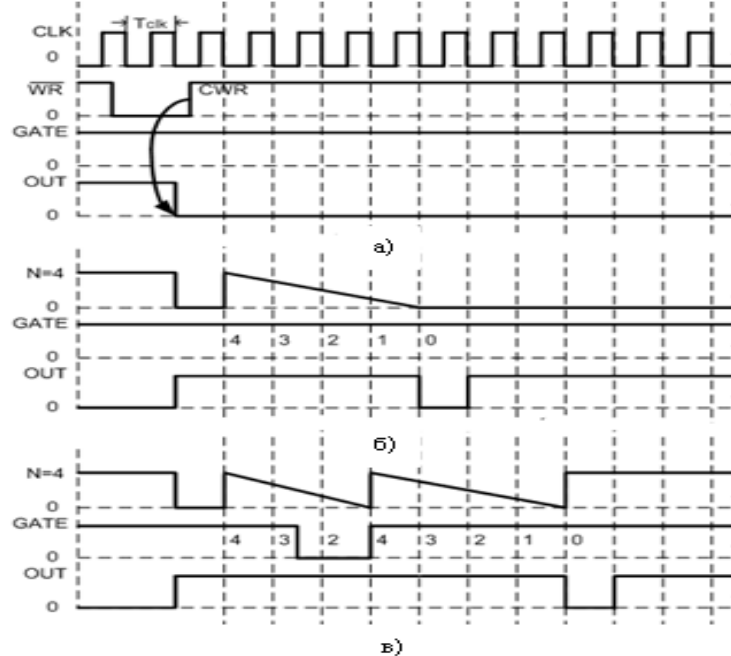


Рисунок 56 – Часова діаграма роботи таймера в режимі 5  
 а) завантаження керуючого байта;  
 б) робота таймера з  $N=4$ ,  $GATE=1$ ;  
 в) робота таймера при зміні  $GATE$ .

## 5.4 Програмування таймера

Програмування таймера досить гнучке і немає значення послідовність завантаження керуючих слів режиму окремих лічильників, тобто необов'язково повинно бути першим керуюче слово лічильника 0. Регістр керуючого слова режиму кожного лічильника має свою адресу і може завантажуватись незалежно. Однак завантаження змісту лічильника повинно відповідати послідовності, запрограмованій в керуючому слові режиму ( старший байт, молодший байт ).

При програмуванні таймера програміст вирішує наступні задачі:

1. Ініціалізація каналів інтервального таймера. Здійснюється за допомогою

- Запису керуючого слова CWR,
- Запису константи перерахунку.

2.-Читання регістру стану інтервального таймера, або константи перерахунку RBC ( Read Back Command) здійснюється за допомогою

- запису керуючого байту RBC,
- читання регістру стану ініціалізації таймеру RST,
- читання константи перерахунку:

- а) з зупинкою таймера
- б) без зупинки таймера.

Керуюче слово задає один з шести режимів роботи, тип рахунку (двійковий чи двійково-десятковий), порядок завантаження і розмірність (один чи два байти) константи. Воно завантажується в регістри RSW каналів таймера і зберігається до наступного перепрограмування. Після цього завантажуються константи, які приймають значення:

- двійковий - 0 - 65536
- двійково-десятковий - 0 - 9999

Константи перерахунку завантажуються у таймер при A0, A1 ( 00, 01, 10 ). Існує 2 варіанти завантаження: спочатку всі слова керування, а потім константи, або слово керування і константу для кожного каналу послідовно (рис.57):



Рисунок 57 - Формат керуючого слова

### Читання вмісту лічильника

Якщо потрібно знати значення лічби у процесі роботи (особливо це використовується при визначенні кількості подій), то можна використати два способи читання, які не пошкоджують інформацію:

1) Використовуючи команду введення із обраного лічильника (READ), керуючи входами A0-A1 програміст може вибирати лічильник значення якого потрібно знати.

При читанні значення лічильника потрібно заборонити роботу лічильника за допомогою команди GATE або шляхом заборони тактових імпульсів. При читанні записів лічильника першим читається молодший байт LSB, другим - старший байт MSB. Слід пам'ятати, що процедура читання повинна бути обов'язково доведена до кінця. Якщо запрограмовано читати два байти, то вони повинні бути прочитані до наступної команди завантаження.

2) Другий спосіб - читання під час лічби по команді CLC (на льоту), без зупинки. Для реалізації цього способу таймер має додатковий регістр, звернення до якого здійснюється за допомогою команди виведення. При цьому для читання змісту лічильника "на льоту" в регістр керуючого слова завантажується спеціальний код, під дією якого зміст лічильника без впливу на його роботу фіксується у додатковому регістрі. Після чого використовується звичайна операція ВВЕДЕННЯ для читання змісту регістра.

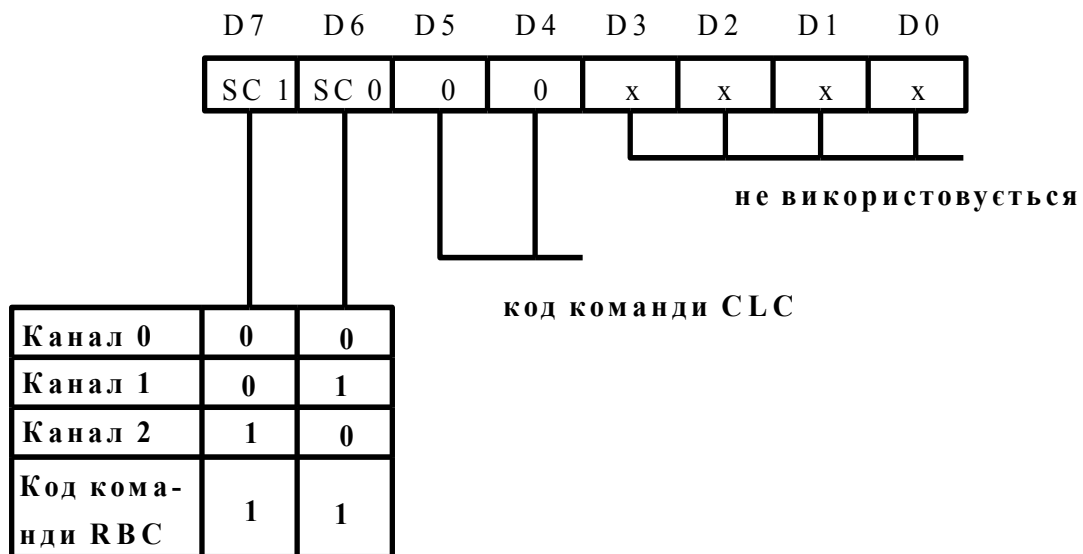


Рисунок 58 - Формат команди CLC

Таймеру відповідають чотири порти введення/виводу з наступними адресами наприклад:

- 40h - канал 0;
- 41h - канал 1;
- 42h - канал 2;
- 43h - керуючий регістр.

### *Програмування таймера*

Приклад:

```
MVI A, 00110000B ;формування керуючого слова лічильника 0
OUT PORT_RUS    ;запис 2-х байт конст., режим 0, рахунок двійковий-десятковий
MVI A, DATA_LOW ; завантаження молодшого байта конст. У лічильник 0
OUT PORT_0      ; при A0,A1=00
MVI A, DATA_HIGH ; завантаження старшого байта конст. У лічильник 0
OUT PORT_0      ; при A0, A1=00
MVI A, 00000000B ; формування керуючого слова для читання
OUT PORT_RUS    ; на льоту слова стану і завантаження в таймер
IN PORT_0       ; читання молодшого байта лічильника 0
IN PORT_0       ; читання старшого байта лічильника 0
```

За допомогою емулятора програмованого таймеру PIT I8253/54 прослідкуємо процес програмування пристрою. Загальний вигляд емулятора представлений на рис.59(timer1.exe):

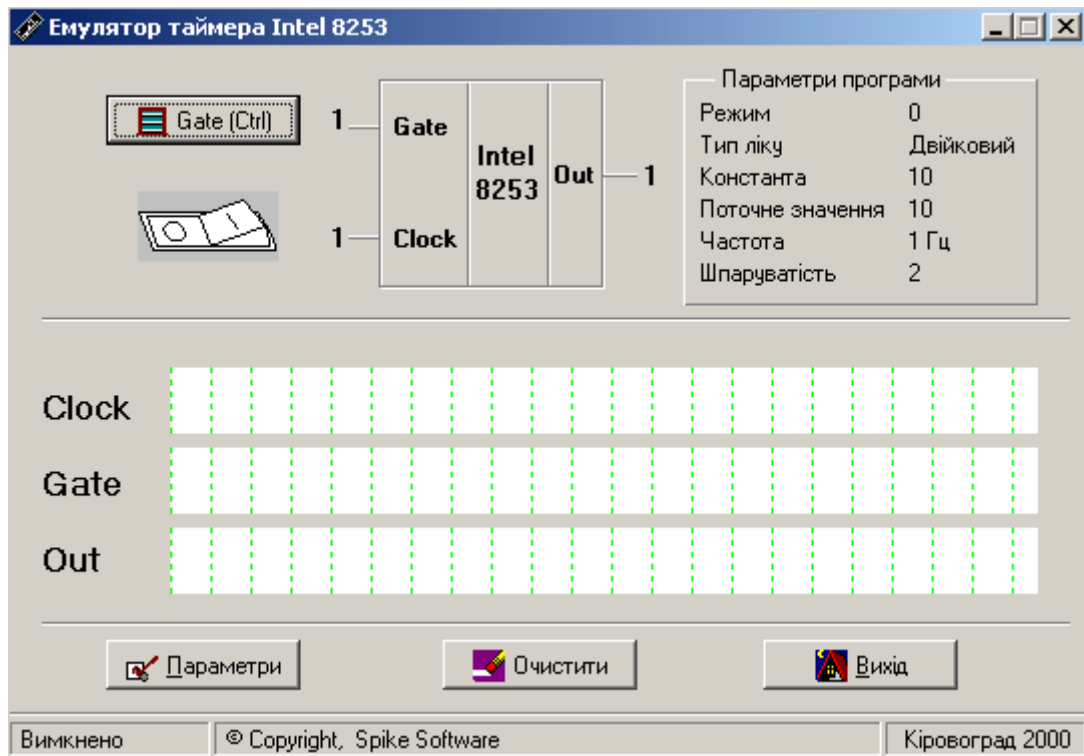
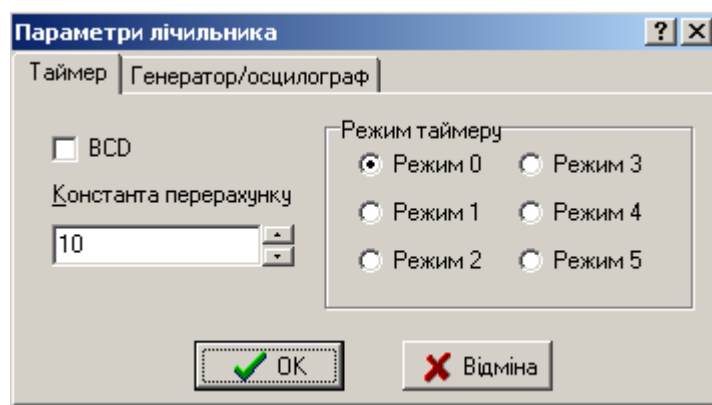


Рисунок 59 - Загальний вигляд емулятора PIT

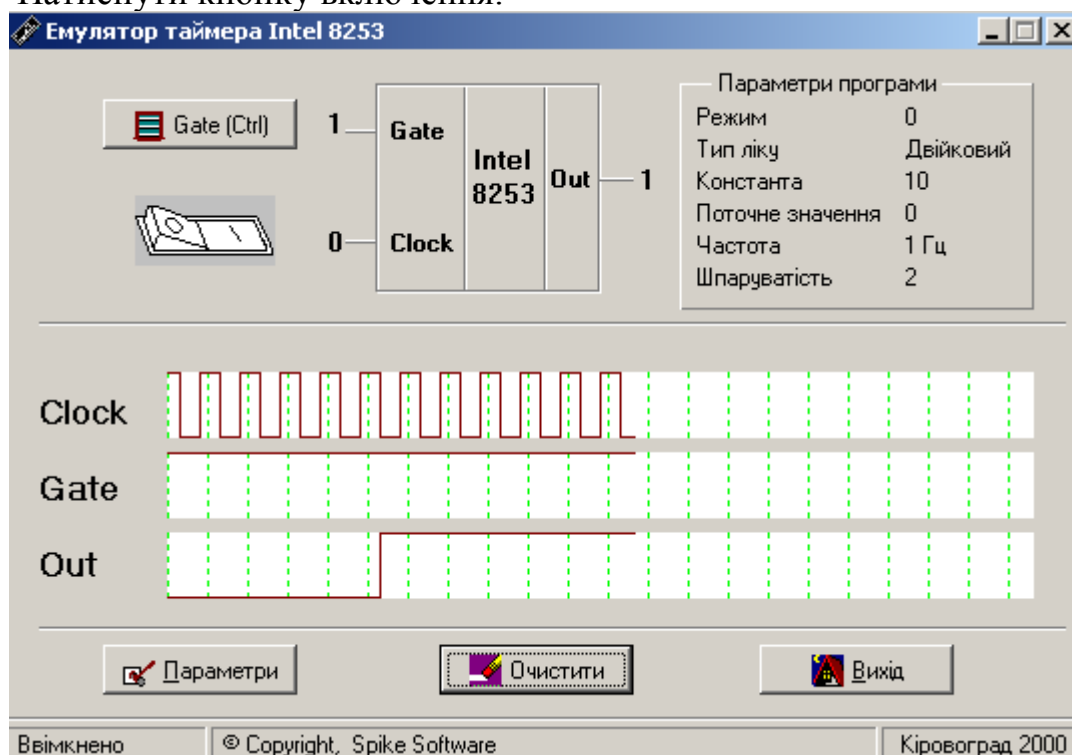
Завантаження програми до емулятора роботи таймера

1. Для початку роботи емулятора необхідно завантажити керуюче слово режиму до відповідного каналу та константу перерахунку. У вікні емулятора натиснемо кнопку «Параметри».



2. У вікні «Параметри лічильника» вибрати режим та вказати константу перерахунку і натиснути кнопку «ОК».

3. Натиснути кнопку включення.



4. В вікні емулятора відображаються сигнали Clock, Gate та Out у заданому режимі заданого каналу. При натисненні кнопки Gate робота таймера буде змінюватися в залежності від режиму роботи.

Використавши комплекс для програмування таймеру загальний вигляд якого представлений на рисунку 60 можна набрати та відлагодити програму роботи пристрою. Під час запуску програми можна в покроковому режимі відслідкувати завантаження всіх команд та роботу пристрою, який програмується на індикаторах формату команд, даних та вхідних і вихідних сигналів ВІС.



Рисунок 60 - Загальний вигляд комплексу програмування таймера

Загальний вигляд вікна компілятора представлено на рисунку 61:

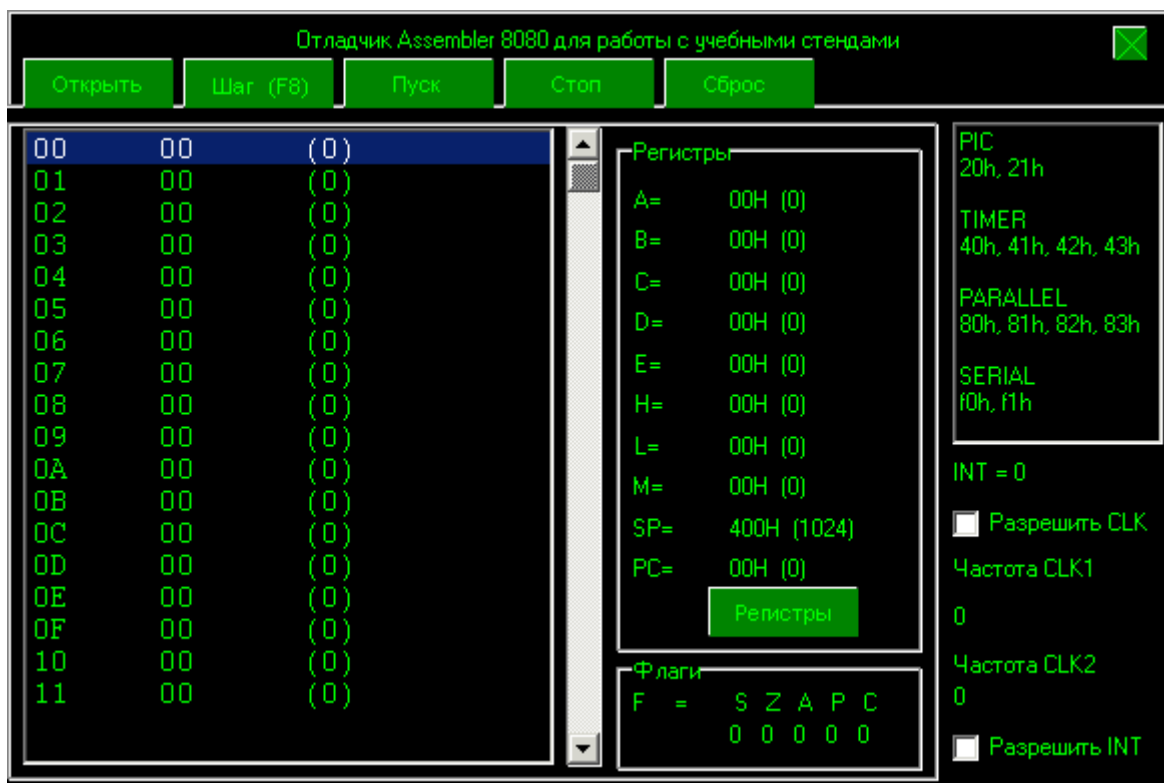


Рисунок 61 - Загальний вигляд вікна компілятора

### Програмування таймера I8253\I8254 (КР 580ВН53) ASM-80

Org 800h

```

Lda rej0          ; перший заданий режим
Out 43h          ; до RUS
Lxi h,c0         ; адреса константи
Mov a,m          ; зчитування константи
Out 40h          ; до каналу 0
Inx h            ;
Mov a,m          ; зчитування старшого байту конст.
Out 40h          ; до каналу 0

```

Читання слова стану

```

Mvi a,xx00xxxxb ; керуюче слово для читання слова стану
                  ; на льоту (D7, D6-номер каналу якого треба
                  ; прочитати слово стану, D5,D4-прикмета
                  ; керуючого слова

```

```

out 43h
in xxh           ; де xx –номер каналу ,який ми читаємо (мол.б)
in xxh           ; старший байт константи

```

де у слові стану - D0 - тип рахунку, D1; D2; D3 - код режиму, D4, D5 - код послідовності завантаження константи , D - дає інформацію чи завантажена константа, D7 - стан виходу

```

hlt
Rej0: db 00110000b ;керуючі слова заданих режимів реж.0,кан.0
Rej1: db 00110010b ; реж.1,кан.0
Rej2: db 01110100b ; реж.2, кан 1
Rej3: db 01110110b ; реж.3, кан. 1
Rej4: db 10111000b ; реж.4, кан. 2
Rej5: db 10111010b ; реж.4. кан.2
C0: dW xxh         ; константа
end

```

## ASM-86

Masm

Model small

. stack 256h

. data

```

Rej0: db 00110000b ;керуючі слова заданих режимів реж.0, кан.0
Rej1: db 00110010b ; реж.1,кан.0
Rej2: db 01110100b ; реж.2, кан 1
Rej3: db 01110110b ; реж.3, кан. 1
Rej4: db 10111000b ; реж.4, кан. 2
Rej5: db 10111010b ; реж.4. кан.2
Cons : dw xxh      ; константа

```

.code

main:

```

mov ax, @data    ; налаштування на сегмент даних
mov ds, ax
mov cx, 6

mov al, rej0     ; канал 0, режим 0

```

```

out 43h, al
mov ax, cons
out 40h, al
out 40h, ah
mov al, rej1 ; канал 0, режим 1
out 43h, al
mov ax, cons
out 40h, al
out 40h, ah

mov al, rej2 ; канал 1, режим 2
out 43h, al
mov ax, cons
out 41h, al
out 41h, ah

mov al, rej3 ; канал 1, режим 3
out 43h, al
mov ax, cons
out 41h, al
out 41h, ah

mov al, rej4 ; канал 2, режим 4
out 43h, al
mov ax, cons
out 42h, al
out 42h, ah

mov al, rej5 ; канал 2, режим 5
out 43h, al
mov ax, cons
out 42h, al
out 42h, ah

```

```

exst:
    mov ax, 4c00h
    int 21h
end main

```

## Ci

```

#include <stdio.h>           підключення бібліотеки
#include <dos.h>             підключення бібліотеки
void main ()                головна підпрограма
{
    unsigned c0=0xFFFF;     // константа
    unsigned char rej0=0x30; // слово керування
                           // запис керуючого слова
    outportb (0x43,rej0);
                           // запис константи
    outportb (0x40,c0&ff);  // запис молодшого байту константи
    outportb (0x40,c0>>8); // запис старшого байту константи
}

```

## **Контрольні питання та завдання**

1. Архітектура програмованого таймера (ПТ) КР580ВИ53. Призначення, структурна схема, призначення основних блоків, вхідних/вихідних сигналів.
2. Система мікрокоманд, формати команд, формат слова-стану. Режими роботи ВІС КР580ВИ53 (програмований таймер). Підключення до шин МП.
3. Поясніть адресацію регістрів таймера.
4. Намалюйте схему таймера та поясніть його роботу.
5. На лабораторній роботі створити фрагменти програм ініціалізації таймера, в різних режимах.

## Лекція 6

### Програмований адаптер послідовного інтерфейсу

#### 6.1 Послідовні інтерфейси

Послідовний інтерфейс для передачі даних використовує одну сигнальну лінію, по якій інформаційні біти передаються один за одним послідовно. Звідси – назва інтерфейсу і порту. Англійські терміни - Serial Interface і Serial Port (іноді їх неправильно перекладають як "серійні"). Послідовна передача дозволяє скоротити кількість сигнальних ліній і збільшити дальність зв'язку. В ряді послідовних інтерфейсів використовується гальванічна розв'язка зовнішніх (звичайно вхідних) сигналів від схемної землі пристрою, що дозволяє з'єднання пристроїв, які знаходяться під різними потенціалами.

Обмін інформацією з пристроями ВВ називається програмно керованим обміном. Програмно керований обмін здійснюється за допомогою команд ВВ одним із трьох способів:

- синхронним
- асинхронним
- з перервою програми.

Синхронний обмін здійснюється тоді, коли по часу співпадають готовність зовнішнього пристрою і МП до обміну інформацією. Забезпечення співпадіння по часу готовності зовнішнього пристрою і ЕОМ є основною складністю синхронного обміну. Синхронний обмін використовується рідко і тільки для процесів суворо фіксованих по часу і повністю визначених. Перевагою цього обміну є простота апаратної і програмної реалізації.

Асинхронний обмін здійснюється під керуванням програми в тому випадку, коли ЗП знаходиться в стані готовності. ЕОМ перевіряє стан ЗП програмно і визначає його готовність.

Обмін з перервою програми здійснюється тоді, коли ЗП видає сигнал готовності (запит переривання) по якому МП здійснює перехід до підпрограми обслуговування запиту переривання. По закінченні підпрограми МП повертається до виконання основної програми. На рисунку 62 представлено блок-схему КР580ВВ51:

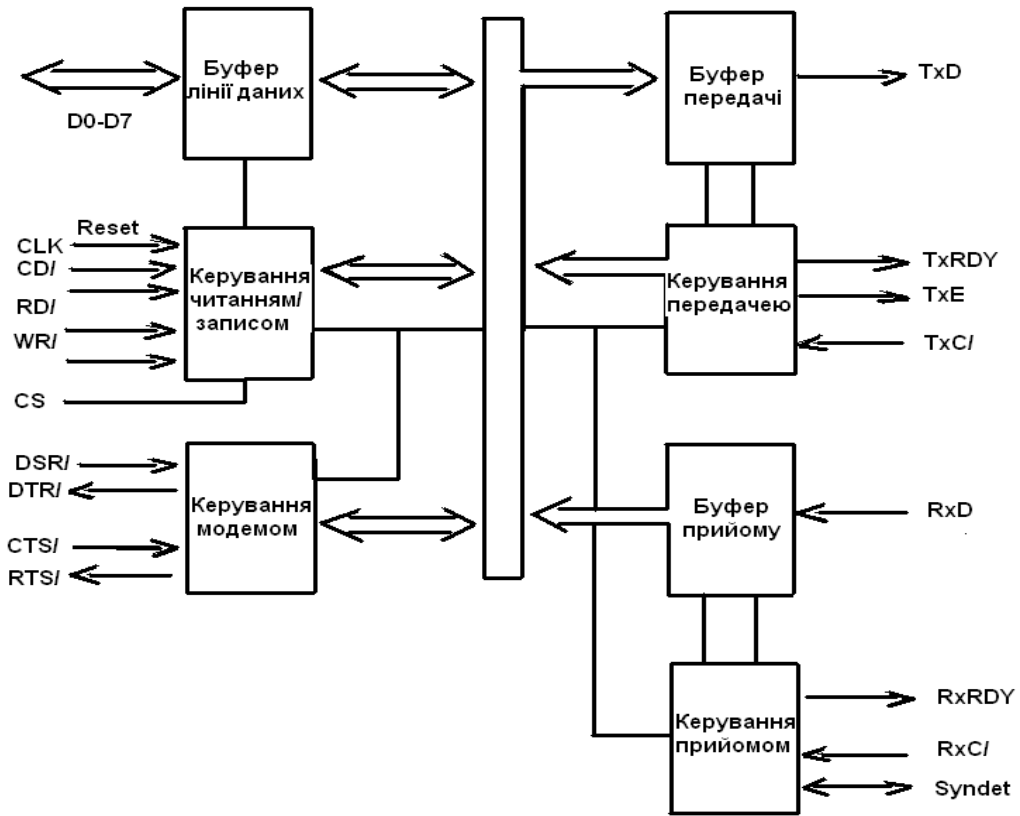


Рисунок - 62 Блок-схема УСАПМ KP580BB51

63: Підключення УСАПМ до шин мікропроцесора представлено на рисунку

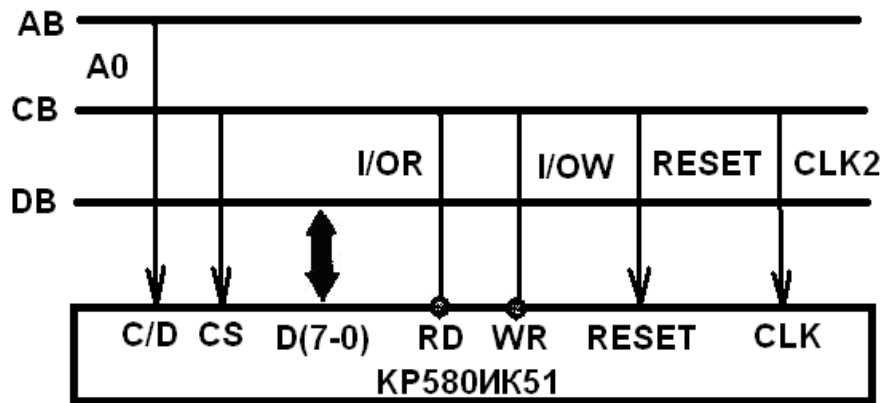


Рисунок 63 - Підключення УСАПМ до шин мікропроцесора

Часові діаграми сигналів керування показані на рисунку 64:

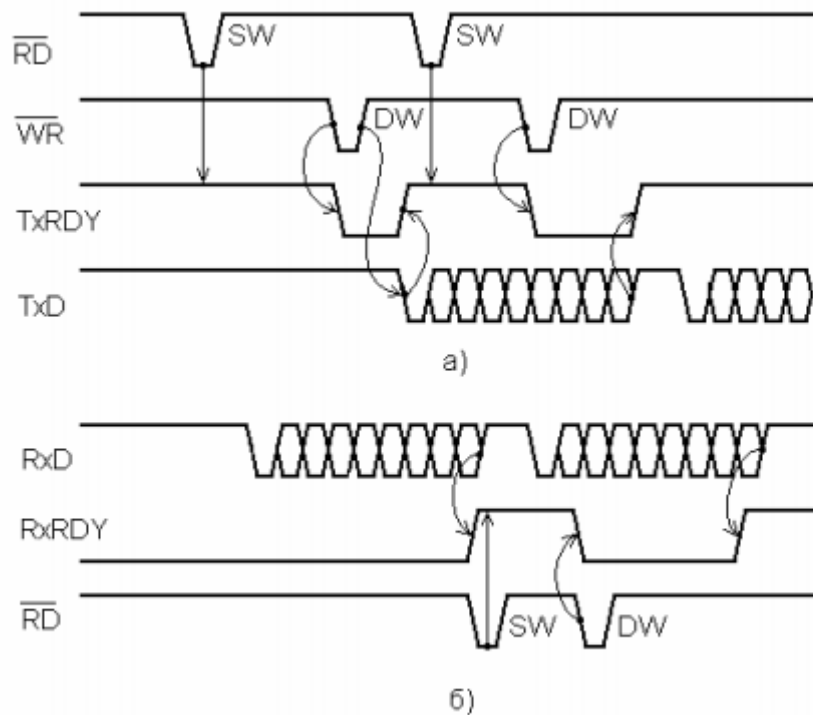


Рисунок 64 – Часові діаграми сигналів керування  
а) передачі б) прийому в асинхронному режимі

Графічне представлення мікросхеми УСАПП КР580ВВ51 показано на  
рисунку 65:

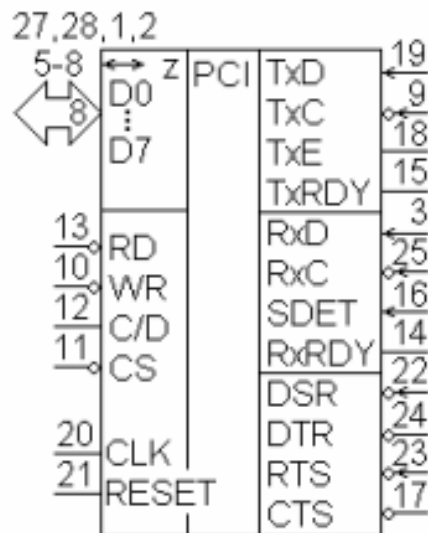


Рисунок 65 - Мікросхема I8251

DSR - (Data Set Ready) - готовність модему. Вхідний сигнал DSR є  
сигналом загального призначення. Його стан може бути перевірений за  
допомогою ЦП і регістра стану. Вхід зазвичай використовується для перевірки  
стану готовності модему.

DTR - (Data Terminal Ready) - основна функція ввімкнути або вимкнути лінію. Програмно він може бути встановлений в стан логічного 0 за допомогою команди керування. Вхідний сигнал DTR використовується для керування і перевірки готовності терміналу, а також вибору швидкості.

RTS - (Request To Send) - основна функція: переключити модем на приймання або передачу. Програмно він може бути встановлений в стан логічного 0 за допомогою відповідного розряду в команді керування.

CTS - (Clear To Send) - характеризує стан комутатора. Стан логічного 0 на цьому вході дозволяє УСАПП передачу даних, якщо розряд TXEN у команді керування знаходиться в 1.

TxRDY - готовність передавача. Цей вихід повідомляє ЦП про готовність передавача прийняти символ даних і може бути використаний для формування запиту в контролер переривань. ЦП може контролювати TxRDY за допомогою читання регістру стану. TxRDY автоматично встановлюється в логічний 0 при завантаженні символу і ЦП. Сигнал TxRDY встановлюється в стан логічної 1 тільки в тому випадку коли CTS встановлено в 0. Цим сигналом TxRDY відрізняється від сигналу TxE.

TxE - відсутність даних у передавача. Коли УСАПП не має даних для передачі, на виході TxE встановлюється стан логічної 1. TxE автоматично встановлюється в стан логічного 0 при отриманні символу із ЦП. TxE може бути використаний для вказівки закінчення режиму передачі.

TxC - імпульс тактування передавача. Керує швидкістю передачі символу. В режимі синхронної передачі частота TxС дорівнює дійсній частоті передачі. У режимі асинхронної передачі частота TxС кратна дійсній частоті.

RxRDY - готовність приймача. Цей вихід показує, що УСАПП має символ готовий до передачі в ЦП. RxRDY може бути використаний в структурі організації перерви або при організації опитування. RxRDY автоматично встановлюється в 0 після передачі символу в ЦП.

RxC - імпульс тактування приймача, керує швидкістю прийому символу.

SYNC - знаходження синхроімпульса. Використовується тільки в синхронному режимі.

D0-D7 - лінії даних.

RESET - гасіння.

CLK - тактовий імпульс.

CD/ - логічний 0 - на шині даних дані; логічна 1 - команда.

RD/ - читання або виведення.

WR/ - запис або введення.

CS/ - вибір корпусу.

TXD - вихід даних передавача.

RXD - вхід даних приймача.

## 6.2 Програмування послідовного адаптера

Функціональне призначення УСАПП визначається способами програмного забезпечення. Для цього ЦП передає в УСАПП ряд керуючих слів після встановлення його в початковий стан.

Керуючі слова розділяються на дві групи:

- команда режиму;
- команда керування;

Команда режиму слідує відразу за встановленням в початковий стан, тобто появою команди RESET. Команда керування повинна слідувати за командою режиму або символами SYNC. Послідовність програмування УСАПП представлено на рисунку 66:

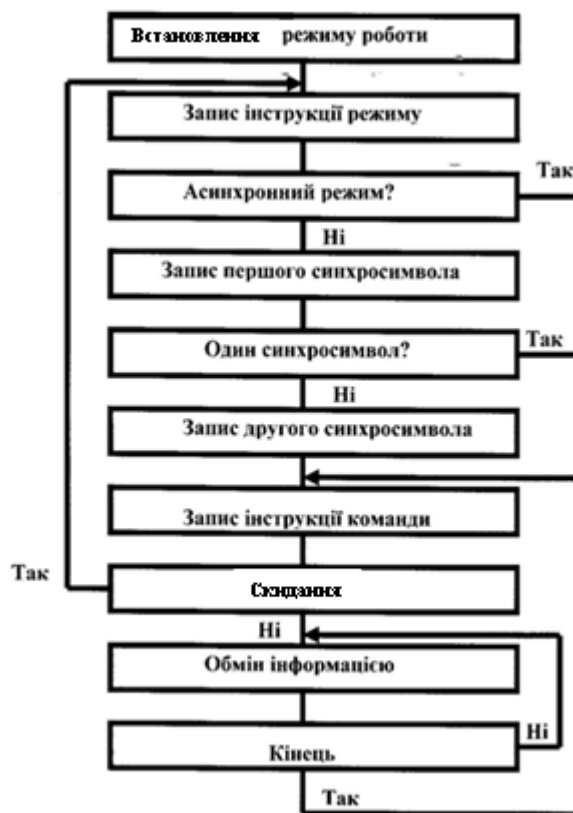


Рисунок 66 - Послідовність програмування УСАПП

Програмну модель адаптера I8051 представлено на рисунку 67:

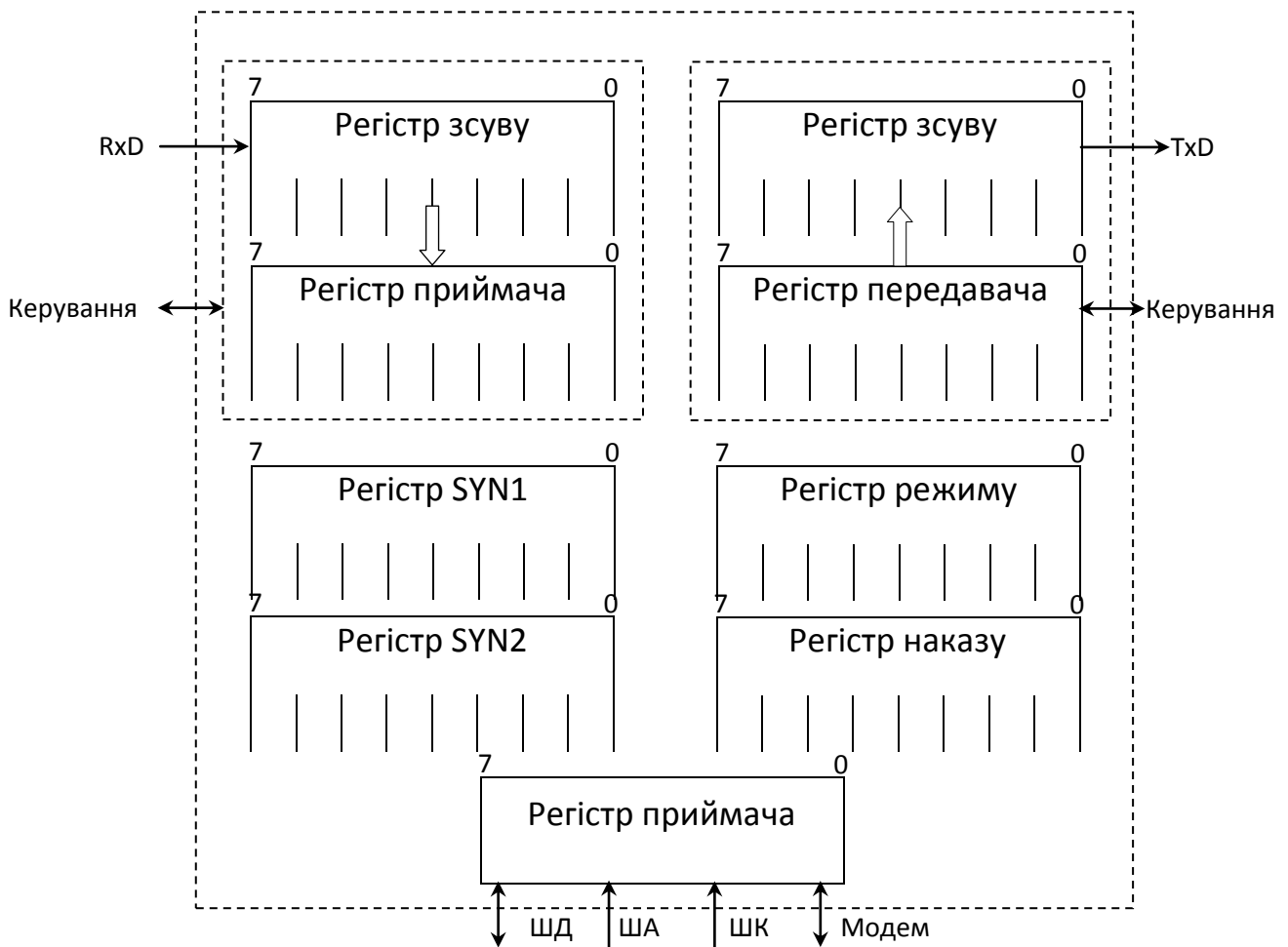


Рисунок 67 - Програмна модель адаптера I8051

Операції, обумовлені сигналами керування від МП

Операція	Сигнали керування			
	C/D	RD	WR	CS
Читання даних з УСАПП на D(7-0)	0	0	1	0
Запис даних з D(7-0) в УСАПП	0	1	0	0
Читання слова стану з УСАПП на D(7~0)	1	0	1	0
Запис керуючого слова D(7-0) в УСАПП	1	1	0	0
Відключення УСАПП від D(7-0)	X	1	1	0
Те ж	X	X	X	1

Примітка X — будь-який стан сигналу.

Команди керування можуть бути записані в будь-який час роботи. Для повернення до команди встановлення нового режиму використовують регістр команди керування. Якщо IR у команді керування встановити в 1, то відбудеться повернення до команди режиму.

Команда режиму визначає основні робочі характеристики адаптера. Команда керування - виконує наступні функції:

- а) встановлення операції введення або виведення;
- б) скидання тригерів прапорців помилок;
- в) керування модемом.

Читання стану

Ця операція дозволяє ЦП в довільний час роботи читати стан адаптера з метою виявлення помилок, а також сканувати запити зовнішніх пристроїв. Читання стану виконується якщо сигнал CD встановлений в стан логічної 1.

Передача/приймання даних.

Після того як слово режиму запрограмує потрібний режим адаптера і при необхідності будуть завантажені один або два синхроімпульси – адаптер готовий до обміну даними. Завантаження відповідного керуючого слова визначає передачу або приймання інформації адаптером. Рівень логічної 1 на TxRDY сигналізує ЦП про те, що адаптер готовий до прийому сигналу. Після запису символу в адаптер рівень TxRDY встановлюється в логічний 0. Адаптер може також приймати послідовні дані від модему або пристрою ВВ і після завершення прийому адаптер встановлює на RxRDY рівень логічної 1, що служить сигналом для ЦП про готовність адаптера передати йому цей символ. Адаптер не може почати передачу до тих пір поки розряд TXEN у команді керування не встановлений в логічну 1 і не отримав сигнал гасіння даних.

Асинхронний режим (передача).

При передачі даних адаптер до перетвореного послідовного коду слова даних додає спочатку стартовий біт, а в кінці стоповий. Крім того, якщо контроль парності передбачений у команді режиму, перед стоп-бітом встановлюється біт перевірки на парність, або непарність. Передача даних здійснюється через вихід TXD. Послідовні дані видаються передавачем на вихід по спаду ТХС/ з частотою 1.16 або 1.64 від частоти ТХС/. Вихід TXD після передачі слова даних і при відсутності наступного символу переходить в стан логічної 1, поки нові дані не надійдуть від ЦП. У команді керування передбачена можливість переводу виходу TXD, при відсутності даних, у стан логічного 0. Формат команди асинхронного режиму представлено на рисунку 68:

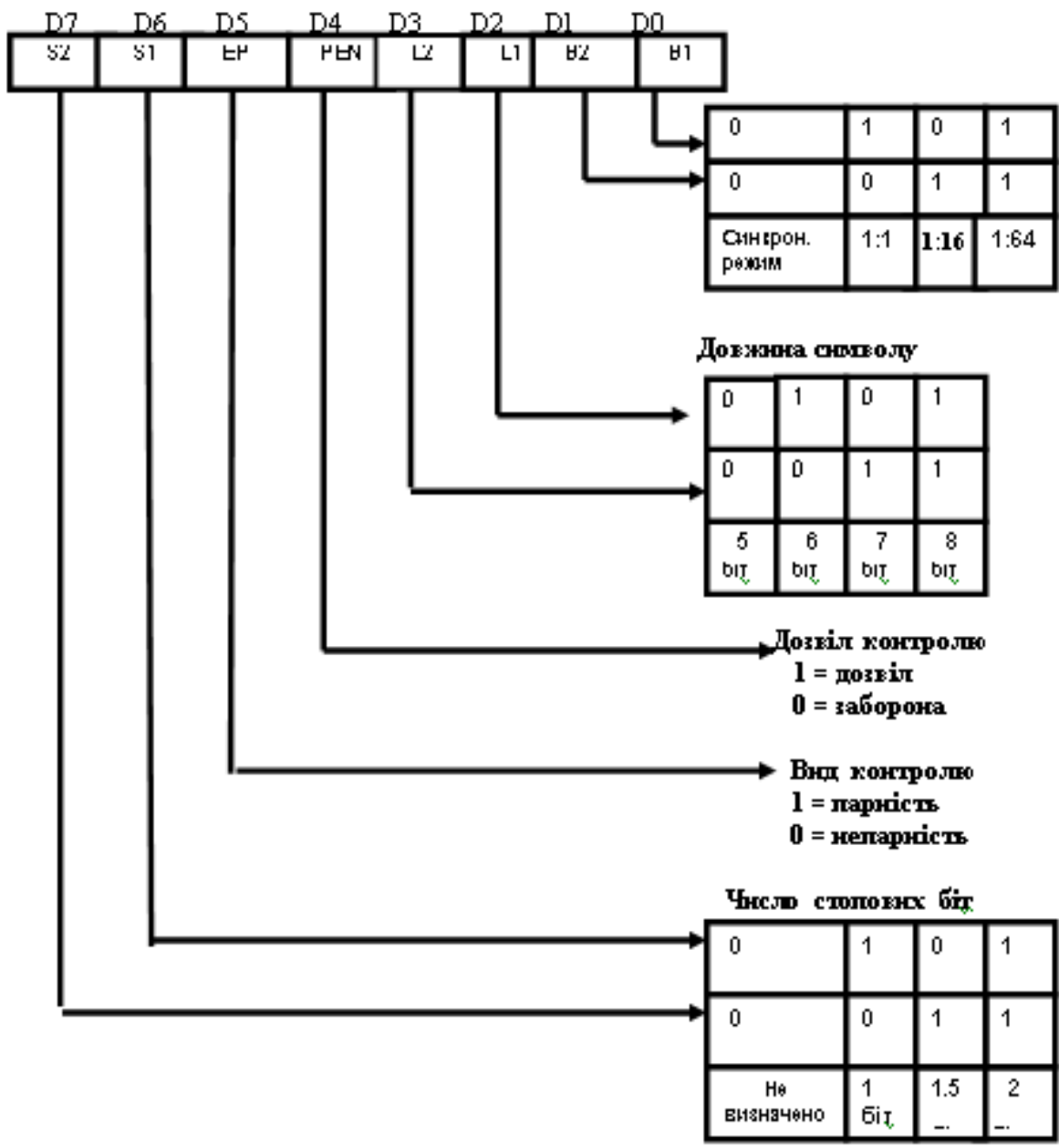


Рисунок 68 - Формат команди асинхронного режиму

Формат команди синхронного режиму представлено на рисунку 69:

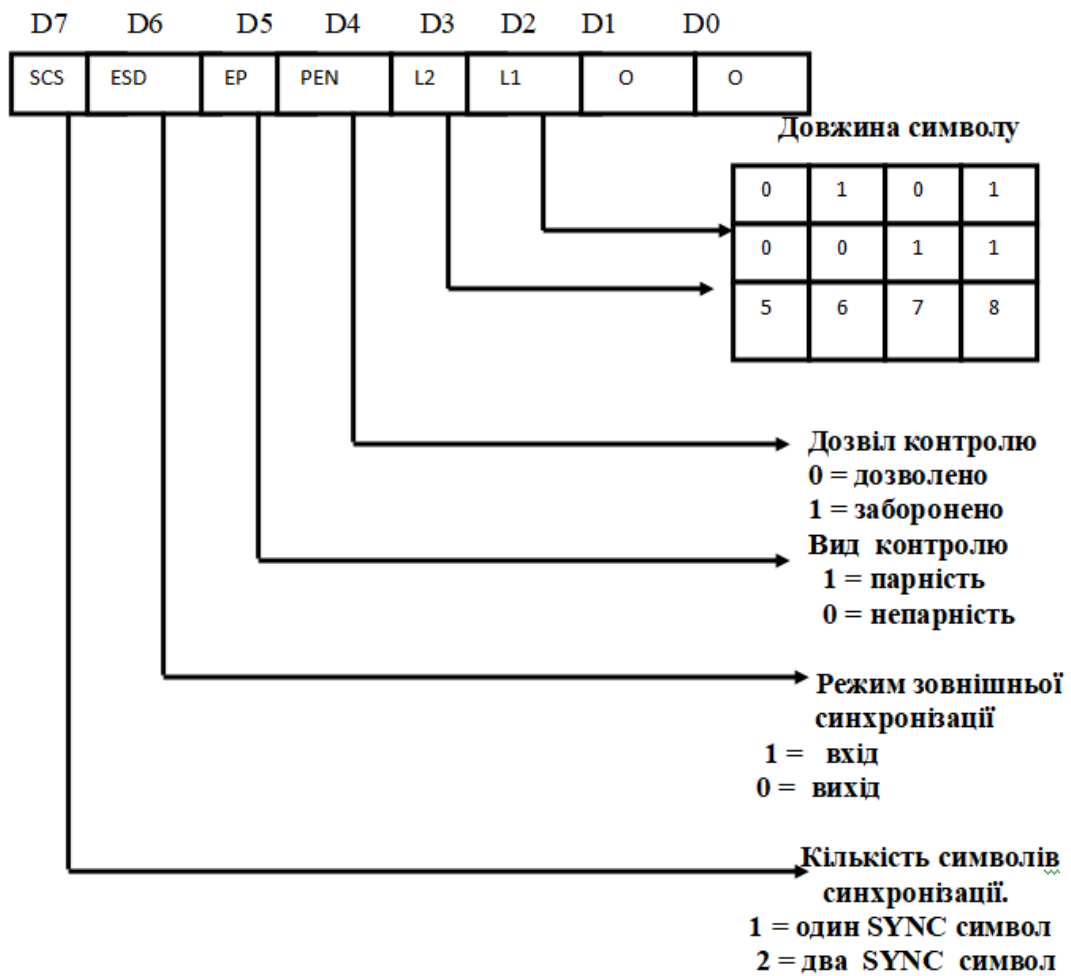


Рисунок 69 - Формат команди синхронного режиму

Формат команди керування представлено на рисунку 70:

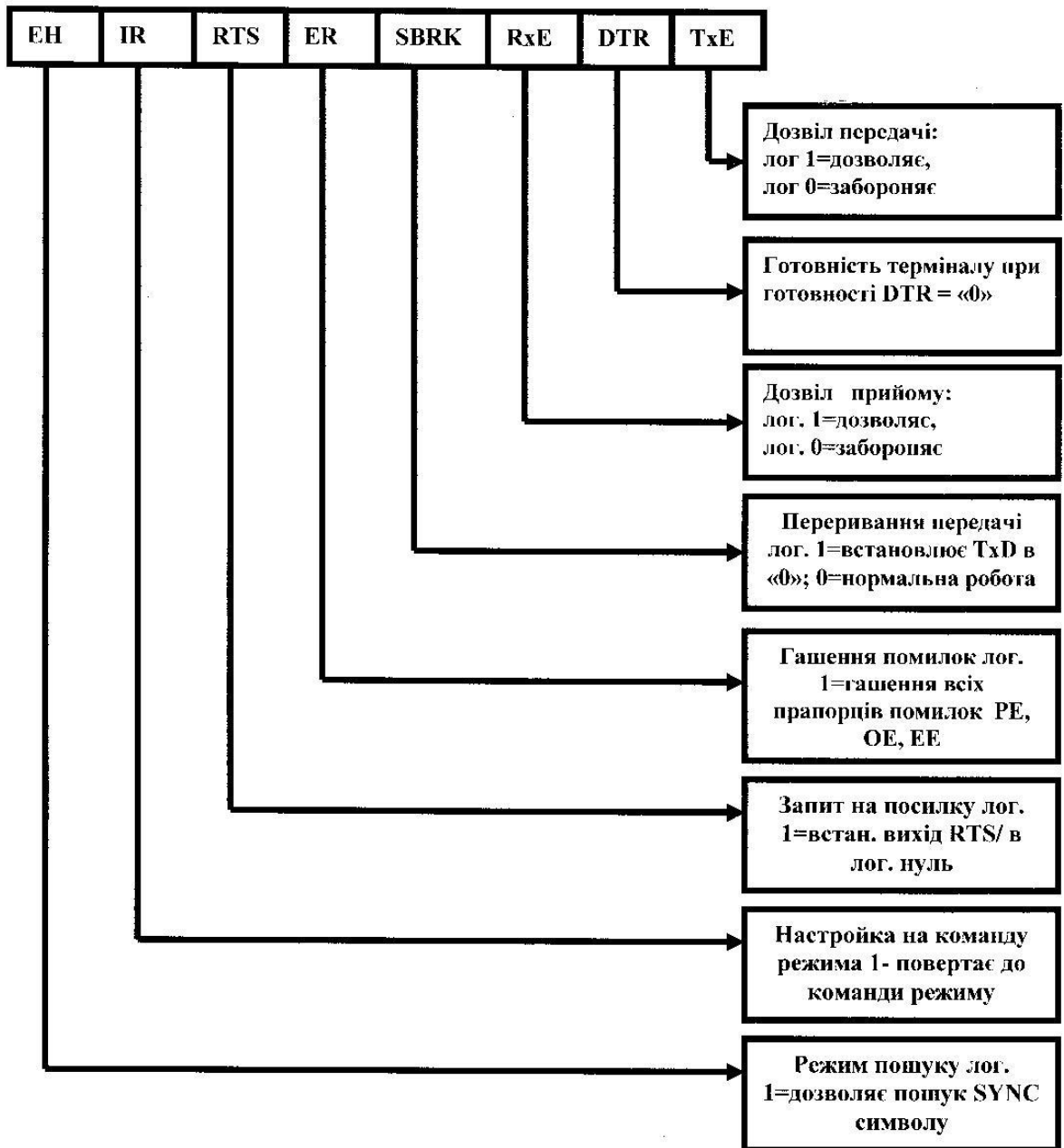


Рисунок 70 - Формат команди керування

Формат слова стану показано на рисунку 71:

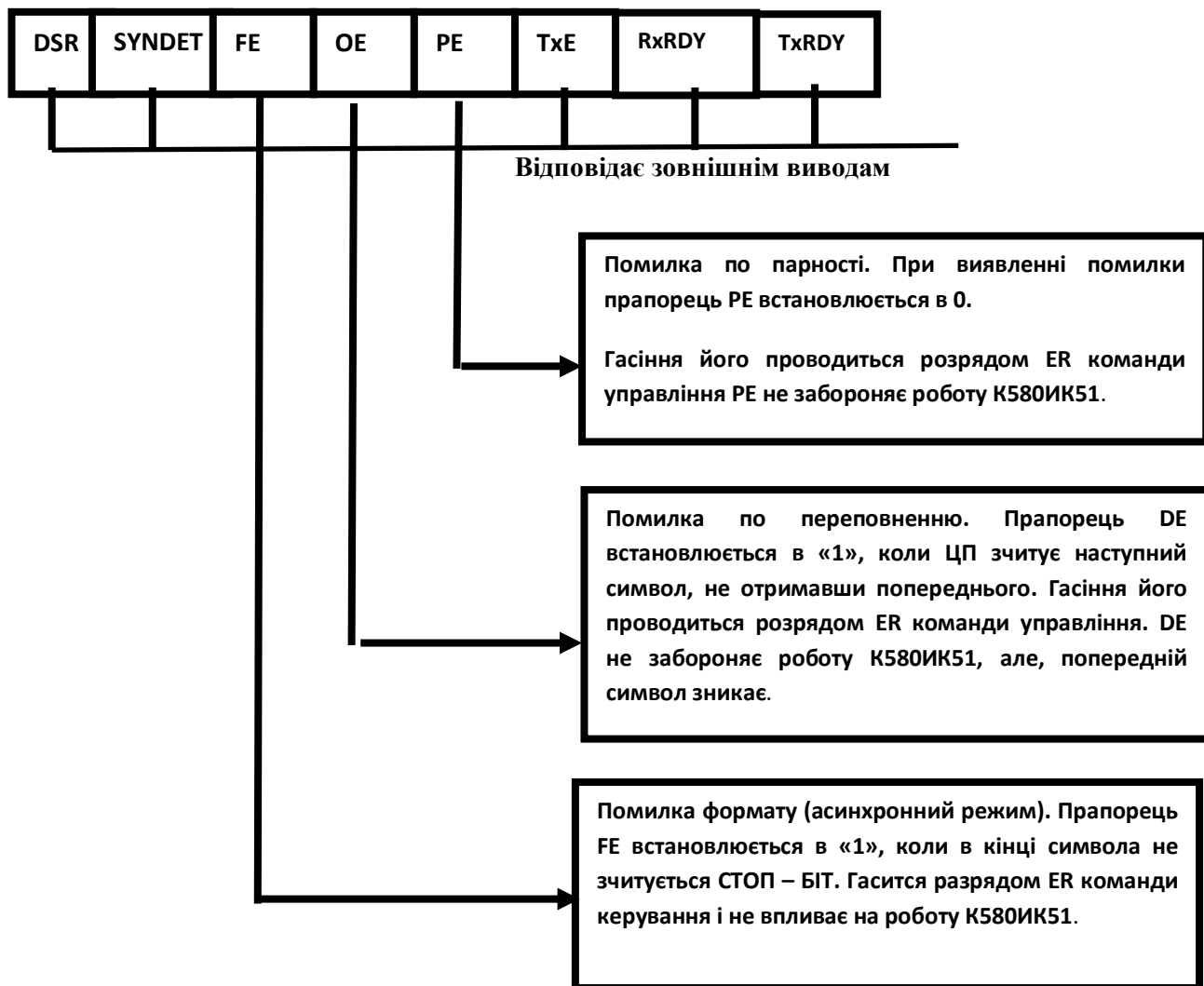
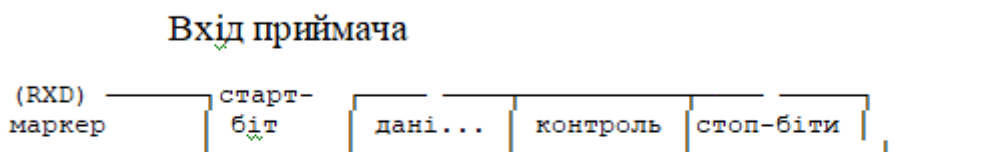
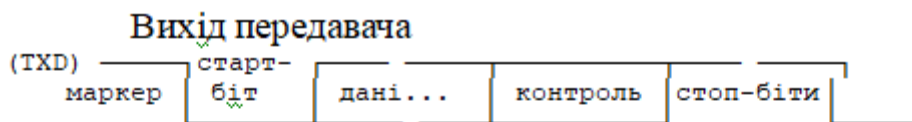


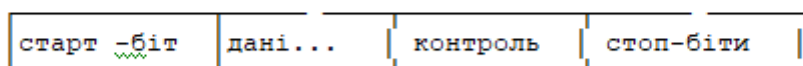
Рисунок 71 - Формат слова стану

### Асинхронний режим (прийом).

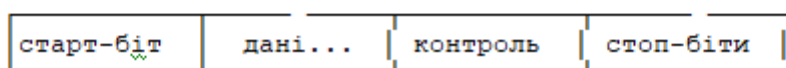
Рівень на вході RXD знаходиться в стані логічної 1. Спад по сигналу на цьому вході свідчить про появу старт-біта. Достовірність цього біта контролюється повторним стробуванням. Повторне знаходження рівня логічного 0 свідчить про достовірність стартового біта. При цьому по спаду RXD/ запускається лічильник, який відраховує задану програмно довжину слова даних, біт парності і стоп-біти. Дані приймаються в послідовному коді по фронту RXC/. При знаходженні помилки по парності встановлюється прапорець помилки по парності. Після прийняття стоп-біта адаптер здійснює передачу прийнятого символу у паралельному коді у буфер даних для передачі у ЦП. При цьому сигнал RxRDY встановлюється в логічну 1 і поточна інформація записується у буфер, стираючи попередню. Всі прапорці помилок скидаються за допомогою команди керування.



**Вихід послідовних даних (TXD)**



**Вхід послідовних даних (RXD)**



**Адресація портів KP580BB51**

БІС KP580BB51 ініціюється при рівні логічного 0 на контакті вибору корпусу, який з'являється коли адреса порту BB на адресних лініях знаходяться в діапазоні EC..EF. Для адресації використовуються розряди з 2-го по 7-й.

Молодший розряд адреси A0 керує входом CD.

адреса пристрою BB	команда	функція	направлення
ED або EF	виведення	команда	ЦП -> УСАПП
EC або EE	виведення	дані	ЦП -> УСАПП
ED або EF	введення	стан	УСАПП -> ЦП
EC або EE	введення	дані	УСАПП -> ЦП

**Організація переривання за допомогою KP580BB51.**

Для організації переривання можна використати вихід готовності приймача RxRDY. Крім цього можна використати вихід готовності передавача TxRDY, а також вихід TXE.

TxRDY – логічна 1 коли адаптер готовий до прийому символів від ЦП.

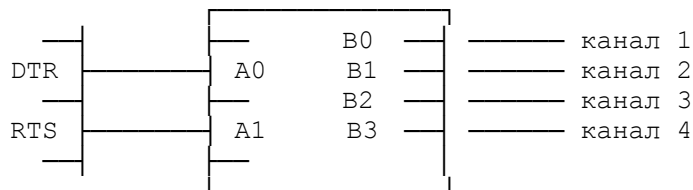
RxRDY – логічна 1 коли адаптер містить символ готовий до передачі в ЦП.

TXE - логічна 1 коли в буфері відсутні дані.

Виходи TxRDY, TXE керуються бітом дозволу прийому у команді керування.

Комутація каналів приймання/передачі.

Програмований послідовний інтерфейс забезпечує передачу і приймання інформації рівнями TTL, або тактовими сигналами. Вибір каналів приймання/передачі здійснюється завдяки програмно-керованим рівням DTR і RTS адаптера.



За допомогою емулятора програмованого адаптера послідовного інтерфейсу I8251 прослідкуємо процес програмування пристрою. Загальний вигляд емулятора представлений на рисунку 72 (vv51.exe).

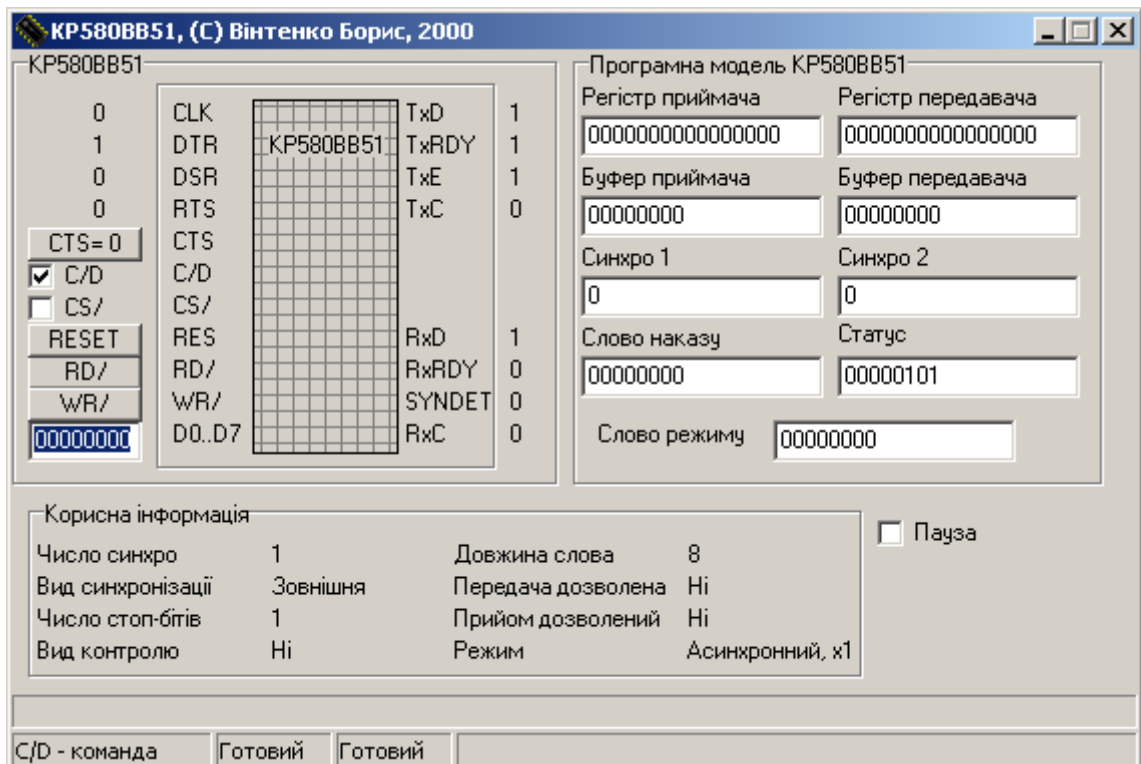
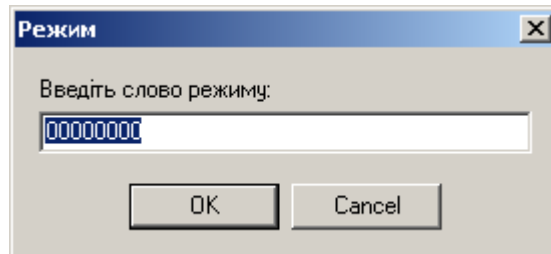


Рисунок 72 - Загальний вигляд емулятора УСАПП I8251

Завантаження програми до емулятора:

1. Для початку роботи емулятора необхідно завантажити керуюче слово (інструкцію) режиму та керуюче слово (інструкцію) команди. В вікні емулятора натиснемо кнопку «Параметри».

2. У відповідному віконці встановимо  $C\backslash D=1(\checkmark)$  та натиснемо "Reset" тим самим виконаємо скидання адаптеру. Після виконаних дій з'явиться вікно, в яке повинні ввести керуюче слово вибраного режиму.

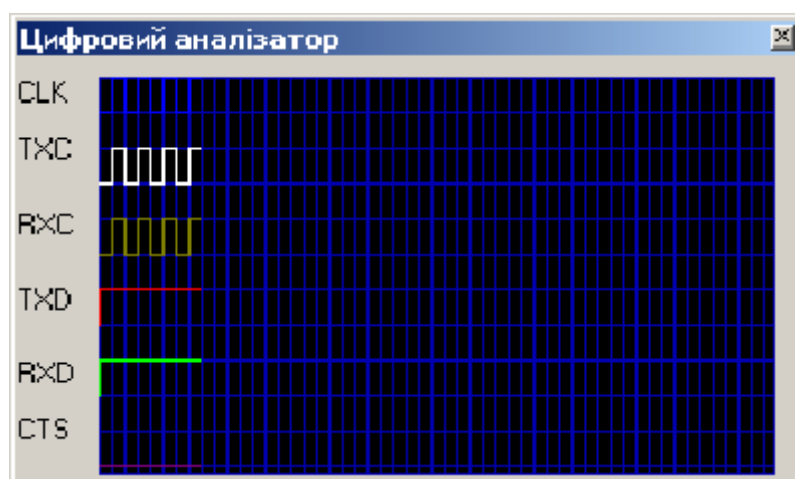


3. Ввести КСР (керуюче слово режиму) і натиснути кнопку "Ok". У віконці «Слово режиму» з'явиться набрана команда.

4. У виділене віконце «D0-D7» введемо КСК (керуюче слово команди) і натиснемо кнопку "WR". Набрана команда з'явиться у віконці «Слово команди». В вікні «Корисна інформація» відобразиться вибраний режим та параметри посилання.

5. Встановимо  $C\backslash D=0$ . Введемо слово даних (до віконця «D0-D7») і натиснемо кнопку "WR". Формат даних відобразиться у віконці «Буфер передавача».

5. Натиснемо кнопку CTS (встановимо  $CTS=0$ ) таким чином дозволивши передачу. Дані послідовно будуть передаватися до зовнішнього пристрою (віконце «Регістр передавача»). У вікні «Цифровий аналізатор» можна відслідкувати процес передачі. В вікні з зображенням мікросхеми та головними сигналами можна спостерігати за зміною сигналів в процесі роботи адаптера.



Використавши комплекс для програмування адаптера, загальний вигляд якого представлений на рисунку 73, можна набрати та відлагодити програму роботи пристрою. В процесі запуску програми можна в покроковому режимі відслідкувати завантаження всіх команд та роботу пристрою, який програмується на індикаторах формату команд, даних та вхідних і вихідних сигналів ВІС.



Рисунок 73 - Загальний вигляд комплексу програмування адаптера

Загальний вигляд вікна компілятора представлено на рисунку 74:

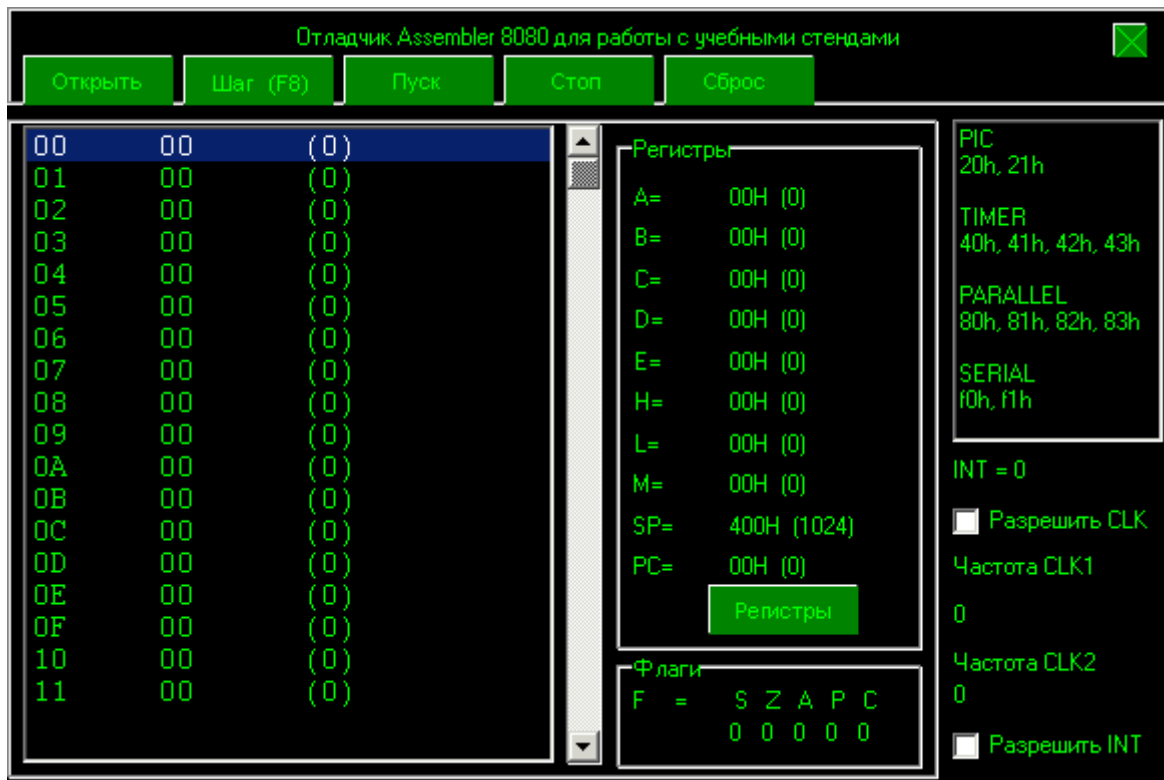


Рисунок 74 - Загальний вигляд вікна компілятора

## Приклад програмування

DI                    заборона переривань  
Mvi A,B6h; налаштування таймера  
Out DFh;    3 режим, лічильник 2  
Mvi A,80h; введення молодшої частини константи  
Out Deh;    по адресі  
Mvi A, 04h; старшої частини, швидкість 50 біт/сек  
Out Deh;    по адресі  
XRa A;       скидання буфера адаптера обнулити A, очистити буфер  
Out EDh  
Out EDh        в буфер надсилаємо нуль  
Out Edh        скидання буфера адаптера  
Out Edh        часова затримка

M1: IN Edh; перевірка очищення буфера  
    Ani 04h; TxE=1  
    jz m1;

movi A,40h; налаштування на режим IR=1  
out edh;

movi A,4eh; режим стоп-біт 1, без контролю  
out Edh; 8 біт інформації, кратність 1\16

Mvi A,15h; команда CI дозвіл передачі та приймання, RTS\=0  
Out edh  
EI                    дозвіл переривань

m2: IN edh читати стан регістра  
    ani 01h    D=0, якщо вихідний буфер порожній і  
    готовий до передачі символів TxRDY=1

JZ m2  
M3:Lxi h,mass                    завантаження масиву  
Lda len                    довжина масиву  
Mov A,C                    встановлення лічильника  
Out EC  
Mov a,m  
Out edh  
Inx h  
Dcr c  
Jz m3

Приймання даних  
M4 IN edh  
ani 04h  
jz m4

IN ech  
Mov M,A  
OUT ECH  
HLT

ADR : DB FFh  
END

## Контрольні питання та завдання

1. Архітектура ВІС послідовного інтерфейсу КР580ИК51. Структурна схема, призначення основних блоків і вхідних/вихідних сигналів.
- 2 Інструкції режиму, команди, слово стану. Режими роботи послідовного інтерфейсу КР580ИК51. Підключення до шин МП.
3. Розробити програми введення та виведення інформації
4. Поясніть синхронізацію приймання даних в синхронному та асинхронному режимах.
5. Чому з збільшенням швидкості передачі збільшується вірогідність похибок даних, що приймаються?
6. Намалюйте структурну схему УСАПП та поясніть режими роботи.
7. Намалюйте схему підключення УСАПП до мікропроцесорної шини.
8. Поясніть послідовність операцій ініціалізації УСАПП.
9. На лабораторній роботі напишіть програми прийому та передачі інформації.

## **Лекція 7**

### **Контролер прямого доступу до пам'яті**

Контролер прямого доступу до пам'яті (ПДП, DMA - Direct Memory Access) забезпечує високошвидкісний обмін даними між пристроями ВВ і ОЗП без використання центрального процесора, що дозволяє звільнити процесор для виконання обчислень паралельно з обміном і незалежно від нього. Відчутні переваги надає використання ПДП у процесі обміну з пристроями, що приймають або передають дані досить великими порціями з високою швидкістю.

У IBM PC – подібних комп'ютерах функції КПДП виконує мікросхема 8237 фірми INTEL. Контролер має 4 незалежних канали, кожний із яких здатен обслуговувати один периферійний пристрій.

#### **7.1 Принцип роботи контролера ПДП ВВ з прямим доступом до пам'яті**

Обмін даними з повільно діючими периферійними пристроями, наприклад принтером створюється за допомогою переривань і якщо підпрограма обслуговування переривань триває 50 мксек, а швидкість передачі 100 символів\с то на ВВ витрачається 0,5% часу процесора і не виникає питань про збільшення продуктивності процесора, але при передачі даних між основною і зовнішньою пам'яттю ЕОМ великих блоків даних продуктивність процесора у режимі переривань стає недостатньою.

Тому для передачі даних безпосередньо між пристроями зовнішньої пам'яті і оперативною пам'яттю, створено спеціальний метод передачі даних без участі ЦП, який отримав назву прямого доступу до пам'яті (ПДП). Апаратні засоби реалізації каналу ПДП називають КПДП.

Існує два різновиди ПДП. У режимі ідентифікації стану пам'яті передавання ПДП виконуються без інформування ЦП, для чого використовуються ті інтервали машинних циклів, коли ЦП не звертається до системної шини, а виконує внутрішні перетворення даних. Такими інтервалами, наприклад у мікропроцесора І8080 є такти  $T_u$  і  $T_s$ . ЦП ( або додаткова схема) ідентифікує ці інтервали для КПДП спеціальним сигналом, який дозволяє доступ до системної шини. Продуктивність ЦП у такому режимі не зменшується.

У режимі з пропусканням тактів (вірніше «запозиченням» тактів) КПДП при необхідності передачі даних сигналом запиту REQ або HOLD процесор відключається від системної шини на декілька тактів. Продуктивність ЦП у такому режимі зменшується.

#### **7.2 Блок схема КПДП**

Типовим прикладом КПДП є ВІС І8257/І8237. Мікросхема може керувати роботою 4 незалежних каналів ПДП, з урахуванням пріоритетів периферійних пристроїв. Для ЦП КПДП являє собою декілька паралельних 8–ми бітних портів вводу-виводу. Після ініціалізації контролер керує передачею блоку даних до 16к

без втручання ЦП. Допускається програмування будь-якого з трьох режимів роботи:

- зчитування (передача з основної пам'яті у зовнішню);
- запис (передача з зовнішньої пам'яті до основної);
- перевірка ПДП.

Спрощена структурна схема КПДП приведена на рисунку 75:

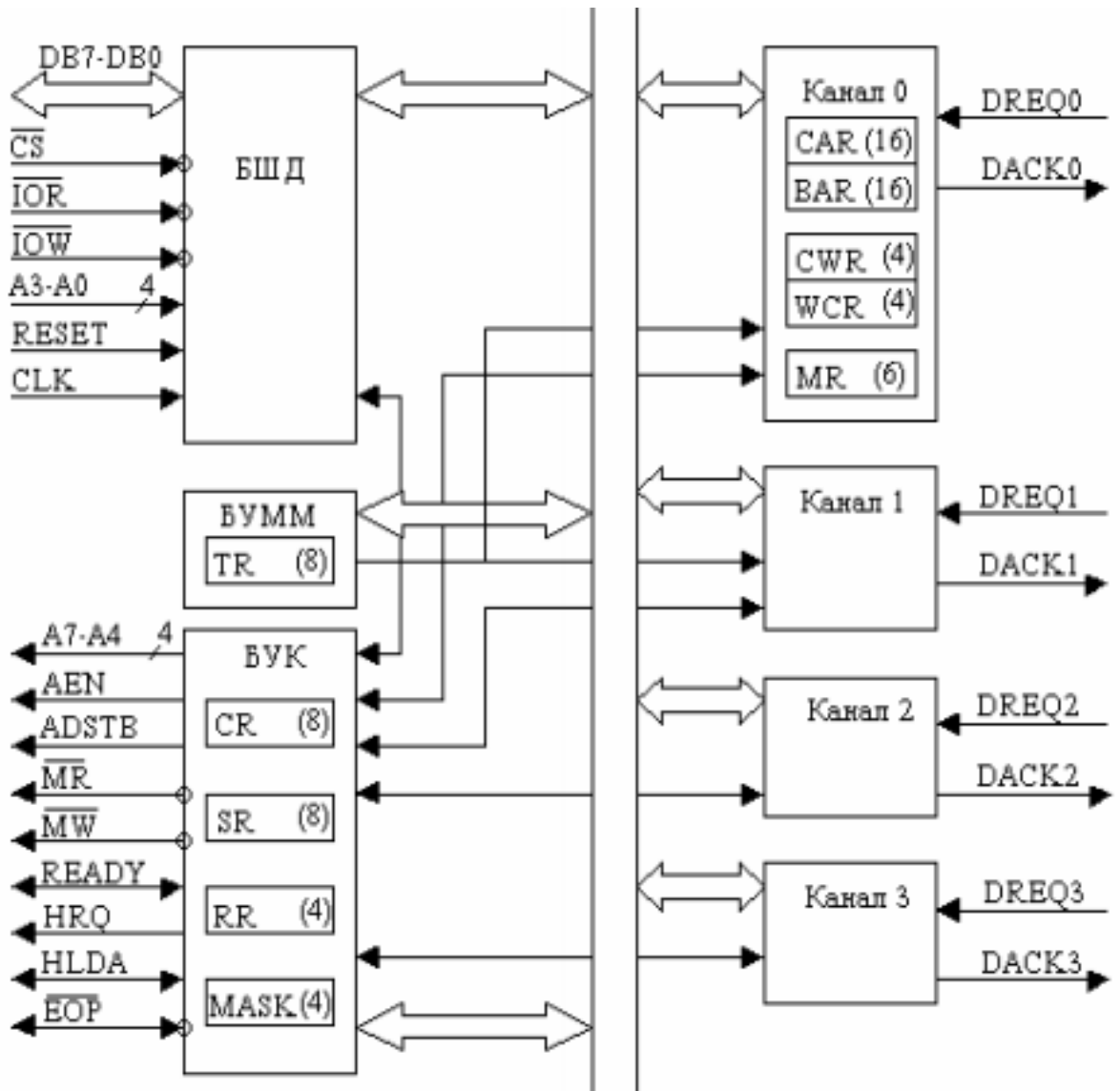


Рисунок 75 – Структурна схема КПДП

До складу ВІС входять: двонаправлений двостабільний буфер даних (BD), призначений для обміну інформацією між МП і КПДП; схема керування читанням/записом (RWCU), яка адресує внутрішні регістри КПДП і керує обміном по шині D (7-0); блок керування (CU), що містить регістри режиму і стану КПДП і забезпечує послідовність операцій, необхідну для організації режиму прямого доступу до пам'яті; блок керування пріоритетами (PCU), забезпечуючий визначений порядок обслуговування запитів зовнішніх пристроїв; чотири канали прямого доступу (СН0-СН3), кожний з яких містить регістр адреси

комірки пам'яті, з яким проводиться обмін, і лічильник циклів обміну, два старших розряди якого відведені для задання операції обміну.

Призначення вхідних, вихідних і керуючих сигналів КПДП:

CLK - вхід для підключення тактового генератора  $F_{CLK} = 3$  МГц.

CS - вибір кристала. 0 CS = дозволяє роботу КПДП.

RESET - скидання. Сигнал високого рівня переводить КПДП в початковий стан, встановлюючи в нуль регістри команд, умов, тимчасового зберігання, а також встановлюючи в одиницю всі розряди маски.

READY - готовність. Вхідний сигнал, що використовується для синхронізації роботи КПДП з повільнодіючими пристроями.

HLDA - підтвердження захоплення. Вхідний сигнал, що використовується ЦП для повідомлення КПДП про можливість виконання циклів ПДП.

DREQ3 - DREQ0 - входи запитів на ПДП від зовнішніх пристроїв. Полярність запитів задається програмно. Сигнали на цих входах повинні утримуватися до приходу сигналу DACK. У початковому стані пріоритет запитів природній, DREQ0 має найвищий пріоритет.

DB7 - DB0 - двонаправлена шина даних з буфером, що мають z-стан. У циклах ПДП на ці лінії видається вісім старших розрядів адресного коду, які необхідно «заклацнути» на зовнішньому регістрі сигналом ADSTB. У режимі роботи з ЦП по цих лініях здійснюється приймання/передача даних.

IOR - читання; як вхід використовується ЦП для читання вмісту внутрішніх регістрів КПДП; як вихід у режимі ПДП дозволяє видачу даних із зовнішніх пристроїв.

IOW - запис; як вхід використовується ЦП для завантаження даних в регістри КПДП; як вихід у режимі ПДП дозволяє запис даних в регістри зовнішніх пристроїв.

EOP - закінчення процесу. Вхід/вихід, який використовується для вказання закінчення процесу передачі даних у режимі ПДП. Подаючи на цей вхід сигнал низького рівня, можна припинити передачу даних. Після завершення передачі даних по одному з каналів на виході встановлюється сигнал 0 EOP =. За цим сигналом знімається запит, і обслуговування припиняється. Якщо встановлено режим автоініціалізації, то відбувається завантаження робочих регістрів даного каналу вмістом базових регістрів, а розряди регістра маски не змінюються. У режимах без автоініціалізації розряди маски і розряд T3 у слові-стану встановлюються відповідно до стану обслуговуваного каналу. При передачі пам'ять - пам'ять вивід EOP орієнтований на вихід, і після закінчення рахунку на цьому виході формується сигнал. Якщо вивід EOP не використовується, то він повинен бути підключений через резистор до шини живлення (+5 В) для запобігання формування хибних сигналів закінчення процесу.

A3 - A0 - адресні входи/виходи. Використовуються як вхідні в режимі роботи з ЦП і для адресації каналів і регістрів каналів КПДП. У режимі ПДП є виходами, по яких передаються чотири молодших розряди адреси ОЗП.

A7 - A4 - адресні виходи, на які в режимі ПДП передаються відповідні розряди адреси ОЗП. У режимі роботи з ЦП переходять у z-стан.

HRQ - вихід запиту захоплення шин. Запит до ЦП для переходу в режим ПДП.

DACK3 - DACK0 - підтвердження ПДП. Вихідні лінії, на які видаються повідомлення для ВП про можливість виконання циклів ПДП. Полярність сигналу задається програмно. Після сигналу RESET на виходах DACK встановлюється нуль.

AEN - дозвіл адреси. AEN = 1 встановлюється на час видачі восьми старших розрядів адреси ОЗП на лінії DB7 - DB0.

ADSTB - строб адреси. Вихід, на якому формується імпульс (строб), який здійснює запис старших розрядів (A15 - A8) адреси ОЗП з шин DB7 - DB0 у зовнішній буферний регістр.

MEMR - читання з пам'яті. Вихід, який використовується в режимі ПДП для управління операцією читання з пам'яті.

MEMW - запис в пам'ять. Вихід, який використовується в режимі ПДП для управління операцією запису в пам'ять.

Ucc - шина живлення (+5 В).

GND - загальний.

### 7.3 Внутрішні регістри КПДП

Контролер має 344 біта внутрішньої пам'яті, організованої у вигляді регістрів. Опис внутрішніх регістрів ПДП приведено в таблиці:

Таблиця - Регістри контролера ПДП

Найменування регістра	Розрядність (біт)	Число регістрів
Регістр початкової адреси (Base Address Register)	16	4
Регістр початкового лічильника циклів (Base Word Count Register)	16	4
Регістр поточної адреси (Current Address Register)	16	4
Регістр поточного лічильника циклів (Current Word Count Register)	16	4
Робочий регістр адреси (Temporary Address Register)	16	1
Робочий регістр лічильника циклів (Temporary Word Count Register)	16	1
Регістр стану (Status Register)	8	1
Регістр команд (Command Register)	8	1
Регістр режиму (Mode Register)	6	4
Робочий регістр (Temporary Register)	8	1

Регістр масок (Mask Register)	4	1
Регістр запитів (Request Register)	4	1

### **Регістр початкової адреси (Base Address Register).**

У цьому регістрі задається стартова адреса ОЗП, з якого починається передача. Регістр містить 16 розрядів і визначає адресу всередині заданої сторінки пам'яті розміром 64к. Задання номера сторінки пам'яті здійснюється через спеціальні сторінкові регістри (Page Registers), підтримувані зовнішньою логікою. Кожен канал ПДП має свій регістр початкової адреси і сторінковий регістр. Такий розподіл пам'яті на сторінки не дозволяє здійснити обмін із блоком пам'яті, що знаходиться на перетинанні двох сторінок. Кожна сторінка починається із сегментної адреси, кратного 1000h (0, 1000h, 2000h, ..., 9000h).

### **Регістр початкового лічильника циклів (Base Word Count Register).**

У цьому регістрі задається початкове число циклів передачі для програмованого каналу. Фактичне число переданих під час роботи ПДП елементів даних на одиницю перевищує задане число циклів, тобто якщо задати 100 циклів передачі, а розмір елемента буде дорівнює 1 байтові, то за сеанс обміну буде переданий 101 байт інформації.

### **Регістр поточної адреси (Current Address Register).**

Початкове значення заноситься в цей регістр одночасно з регістром початкової адреси. Надалі в ході передачі значення поточної адреси автоматично збільшується або зменшується (конкретний напрямок зміни задається при програмуванні в регістрі режиму). Якщо дозволено автоініціалізацію, то після закінчення передачі в регістр автоматично встановлюється значення з регістра початкової адреси.

### **Регістр поточного лічильника циклів (Current Word Count Register).**

Регістр містить поточне значення лічильника циклів (число циклів передачі, що залишилися). Відображуване в ньому число циклів завжди на одиницю менше числа ще не переданих елементів даних тому, що зміна значення в цьому регістрі відбувається наприкінці циклу передачі, вже після фактичної передачі елемента даних, а кінець передачі фіксується в момент переповнення лічильника (зміна його значення з 0 на 0FFFFh).

### **Регістр режиму (Mode Register).**

Даний регістр задає режими роботи свого каналу контролера. Кожний з чотирьох каналів ПДП має свій набір регістрів, описаних вище. Крім того, є наступний набір регістрів, загальних для всіх каналів:

Формат команди

<b>D7</b>	<b>D6</b>	<b>D5</b>	<b>D4</b>	<b>D3</b>	<b>D2</b>	<b>D1</b>	<b>D0</b>
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

D1, D0-режим роботи  
0-0-перевірка  
0-1-запис до пам'яті  
1-0-читання з пам'яті  
1-1-неможлива комбінація

D2- автоініціалізація: 1-дозволена, 0-заборонена  
D3-зміна поточної адреси при обміні: 0-збільшення, 1-зменшення  
D4, D5 – тип передачі:  
0-0-режим передачі по вимозі  
0-1-режим звичайної передачі  
1-0-режим блочної передачі  
1-1-каскадний режим

### Регістр команд (Command Register).

Цей 8-бітний регістр керує роботою контролера. Він програмується, коли контролер знаходиться в стані програмування й очищається командами скидання «Reset» і «Master Clear». Призначення бітів регістра команд приведено на рисунку:

Формат команди

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

D0:0-заборонити передачу пам'ять-пам'ять; 1 – дозволити  
D1:0-заборонити фіксацію адреси в каналі 0; 1 – дозволити  
D2:0-заблокувати; 1 - розблокувати контролер  
D3:0-нормальна часова діаграма; 1 – стискування в часі  
D4:0-режим фіксованих пріоритетів; 1 – режим циклічного зсуву  
D5:0-затримка при запису, якщо біт 3 встановлений ігнорується режим розширеного запису  
D6:0-активним є високий рівень сигналу запиту DREQ, 1 - низький  
D7:0-активним є високий рівень сигналу підтвердження запиту на ПДП (DACK), 1 - низький

### Регістр стану (Status Register).

Регістр відбиває поточний стан запитів і передач по всім чотирьох каналам. Біти 0 - 3 встановлюються в одиницю після завершення передачі по каналах 0 - 3 (біт 0 - канал 0, біт 1 - канал 1 і т.д.), якщо не заданий режим автоініціалізації. Ці біти очищаються після команди скидання контролера і після кожної операції зчитування стану з регістра стану. Біти 4 – 7 вказують по якому з каналів 0 - 3 активний у даний момент сигнал запиту на ПДП.

### Регістр масок (Mask Register).

Кожен біт цього 4-х бітового регістра маскує/демаскує свій канал ПДП, при цьому значення 1 маскує канал, значення 0 демаскує канал і дозволяє приймання сигналу запиту по цьому каналі.

### Регістр запитів (Request Register).

Сигнал запиту на ПДП (DREQ) може бути виданий пристроєм, що обслуговується і програмно. Для програмного видання сигналу запиту по одному з 4-х каналів ПДП необхідно встановити відповідний біт у 4-х розрядному регістрі запитів. Запит на ПДП може бути відмінений записом нульового значення у відповідний біт регістра. Біт запиту очищається автоматично після закінчення передачі по даному каналу. Усі біти запитів очищаються при скиданні контролера. Для того, щоб сприймати програмні запити на ПДП, канал повинний знаходитися у режимі блокової передачі.

### **Робочий регістр (Temporary Register).**

Цей 8-розрядний регістр використовується для збереження елемента даних, переданого в режимі фіксованої адреси при передачі пам'ять-пам'ять або для тимчасового збереження переданого байта при всіх інших режимах передачі.

## **7.4 Програмне керування контролером ПДП**

Програмне керування контролером ПДП здійснюється через порти ВВ. Доступ до кожного регістра контролера може бути здійснений через свої порти ВВ.

Порти 0h - 7h призначені для запису вихідних значень у регістри початкової і поточної адреси, початкового і поточного лічильника циклів для всіх 4-х каналів. Порти восьмирозрядні, а регістри, у які через них заносяться дані, 16-розрядні, тому запис відбувається в два прийоми. Перед першою командою виводу в необхідний порт необхідно скинути тригер-засувку, для чого виконується команда виведення довільного значення в порт 0Ch, після чого в необхідний порт виводиться молодший байт 16-розрядного значення і потім старший байт наступної команди виведення в той же порт. Часова діаграма роботи КПДП представлена на рисунку 76:

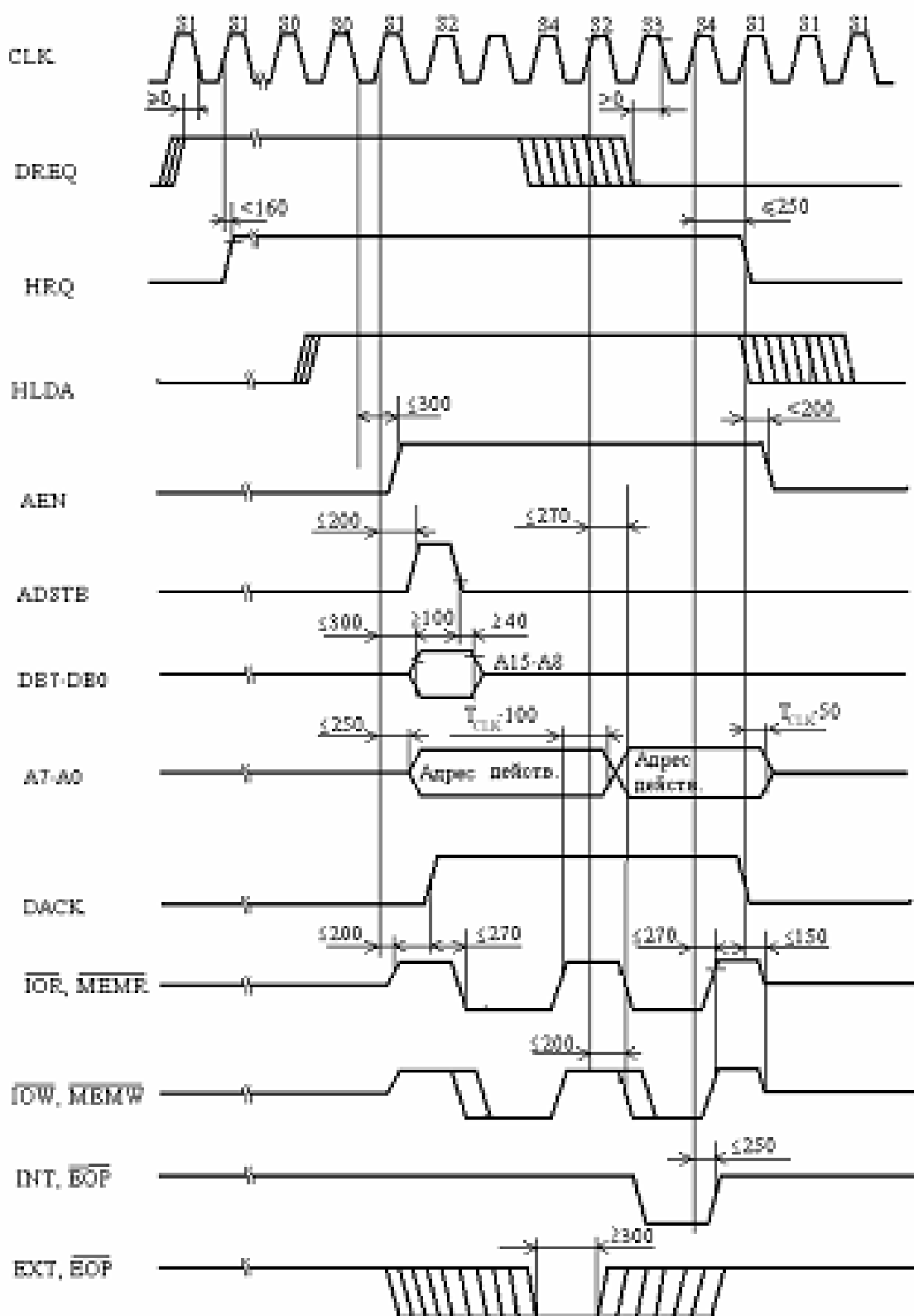


Рисунок 76 – Часова діаграма роботи КПП

Виведення у порт 8h дозволяє занести значення в реєстр команд ПДП. Читання з порту 8h зчитує реєстр стану ПДП. Запис у порт 9h дозволяє встановити або скинути біт запиту в реєстр запитів для одного з каналів. Формат команди наступний:

<b>D7</b>	<b>D6</b>	<b>D5</b>	<b>D4</b>	<b>D3</b>	<b>D2</b>	<b>D1</b>	<b>D0</b>
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

D1,D0 – вибір каналу  
 0-0-канал 0  
 0-1-канал 1  
 1-0-канал 2  
 1-1-канал 3  
 D2-встановлення біту запиту на КПДП  
 1-встановити  
 0-скинути

Запис у порт 0Ah дозволяє установити або скинути біт маски в регістрі масок для одного з каналів. Формат команди:

<b>D7</b>	<b>D6</b>	<b>D5</b>	<b>D4</b>	<b>D3</b>	<b>D2</b>	<b>D1</b>	<b>D0</b>
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

D1,D0 – вибір каналу  
 0-0-канал 0  
 0-1-канал 1  
 1-0-канал 2  
 1-1-канал 3  
 D2-встановлення біт маски  
 1-встановити  
 0-скинути

Запис у порт 0Vh встановлює значення в регістрі режимів одного з 4-х каналів ПДП. Біти 0 і 1 задають номер каналу (00 - 0, 01 - 1, 10 - 2, 11 - 3). У біти 2 - 7 заносяться значення, передані відповідно в біти 0 - 6 регістра режимів.

Запис у порт 0Dh задає програмне скидання контролера (Master Clear). Виведення будь-якого байта в цей порт має той же ефект, що й апаратне скидання контролера. При програмному скиданні очищуються регістри команд, стану, запитів і робочий регістр. Так само скидається тригер-засувка і встановлюються всі біти масок у регістрі масок. Після програмного скидання контролер переходить у цикл очікування.

Вивід будь-якого байта в порт 0Eh очищає регістр масок - скидає біти масок усіх 4-х каналів ПДП і в такий спосіб дозволяє прийом запитів на ПДП по всіх каналах.

Через порт 0Fh можна задати довільне значення регістра масок ПДП. Для цього необхідно в бітах 0 - 3 регістра AL встановити необхідне значення масок каналів 0 - 3 відповідно і вивести це значення в порт.

Сторінкові регістри ПДП призначені для задання параметра сторінки пам'яті, з яким буде відбуватись обмін. Під номером сторінки розуміються старші 4 біти повної 20-бітової адреси ОЗП, тобто початкові сегментні адреси сторінок будуть кратні 1000h (0, 1000h, 2000h, ..., 9000h). Архітектура ПДП дозволяє працювати тільки зі сторінкою пам'яті розміром 64ДО, а логіка, що забезпечує перемикання сторінок влаштована так, що сторінки мають певні суворі межі,

описані вище. Через цю особливість неможливо за допомогою ПДП здійснити обмін із блоками пам'яті, що перетинають межу між двома такими сторінками.

## 7.5 Програмна модель КПДП. Програмування КПДП

Для початкового встановлення КПДП необхідно записати відповідну інформацію в 16-розрядний регістр адреси каналу (RGA), у 16-розрядний лічильник циклів каналу і в 8-ми розрядний регістр режиму, загальний для всіх каналів. Запис цієї інформації відбувається за допомогою команди OUT, хоча можливий і інший спосіб звертання до КПДП як до комірок пам'яті. Запис інформації в 16-ти розрядні регістри здійснюється двома командами, починаючи з молодшого байта. Два старших розряди лічильника циклів визначають операцію обміну в такий спосіб:

запис у пам'ять — 01

читання з пам'яті — 10

контроль — 00 (комбінація 11 заборонена).

Стан КПДП можна контролювати читанням вмісту RGA, CT і 8-ми розрядного регістра стану, загального для всіх каналів, за допомогою команди IN. Для читання вмісту 16-розрядного регістра використовуються дві команди IN з однієї і тією ж адресною частиною, причому спочатку відбувається зчитування молодшого байта виходів A (3 — 0) і значення сигналу CS для адресації внутрішніх регістрів КПДП. При програмуванні КПДП операції запису керуючих або слів читання станів внутрішніх регістрів визначаються також значеннями сигналів I/OR і I/O W. При програмуванні КПДП операції запису керуючих або слів читання станів внутрішніх регістрів визначаються також значеннями сигналів I/OR і I/O W.

В таблиці 7.1 показано коди на A3-A0, що відповідають командам ЦП, а в таблиці 7.2 - коди на A3-A0, які відповідають адресам регістрів КПДП.

Таблиця 7.1 - Коди на A3-A0 що відповідають командам ЦП

A3	A2	A1	A0	Команда	Операція
1	0	0	0	Введення	Читання регістру стану
1	0	0	0	Виведення	Запис до регістру команд керування
1	0	0	1	Те саме	Запис до регістру запитів
1	0	1	0	>>	Встановлення всіх розрядів маски
1	0	1	1	>>	Запис до регістру режиму
1	1	0	0	>>	Встановлення режиму введення мол. байта
1	1	0	1	Введення	Читання регістру тимчасового зберігання
1	1	0	1	Виведення	Загальне скидання
1	1	1	0	Те саме	Скидання всіх розрядів маски
1	1	1	1	>>	Встановлення розряду маски

Таблиця 7.2 - Коды на А3-А0, які відповідають адресам регістрів КПДП

А3	А2	А1	А0	Команда	Операція
0	0	0	0	Виведення	Заван. мол/ст байту в рег. BAR і CAR в к.0
0	0	0	0	Введення	Читання змісту CAR каналу 0
0	0	0	1	Виведення	Заван. мол/ст байту в рег. WCR і CWR в к.0
0	0	0	1	Введення	Читання змісту CWR каналу 0
0	0	1	0	Виведення	Заван. мол/ст байту в рег. BAR і CAR в к.1
0	0	1	0	Введення	Читання змісту CAR каналу 1
0	0	1	1	Виведення	Заван. мол/ст байту в рег. WCR і CWR в к.1
0	0	1	1	Введення	Читання змісту CWR каналу 1
0	1	0	0	Виведення	Заван. мол/ст байту в рег. BAR і CAR в к.2
0	1	0	0	Введення	Читання змісту CAR каналу 2
0	1	0	1	Виведення	Заван. мол/ст байту в рег. WCR і CWR в к.2
0	1	0	1	Введення	Читання змісту CWR каналу 2
0	1	1	0	Виведення	Заван. мол/ст байту в рег. BAR і CAR в к.3
0	1	1	0	Введення	Читання змісту CAR каналу 3
0	1	1	1	Виведення	Заван. мол/ст байту в рег. WCR і CWR в к.3
0	1	1	1	Введення	Читання змісту CWR каналу 3

Формат керуючого слова, записується в регістр режиму( Mode Register ) - 0Bh. Даний регістр задає режими роботи каналу контролера. Розряди команди записаної в регістр команд (Command Register)-08h D3-DO(EN3 - EN(J) задають дозвіл обміну по відповідному каналу, запис нуля в розряд забороняє обмін. Інші розряди визначають режими роботи каналу.

Розряд D4(RP) встановлює порядок обслуговування запитів від каналів. При RP = 0 задається фіксований пріоритет каналів і канал 0 має вищий пріоритет. У режимі циклічного пріоритету (RP= 1) після обслуговування каналу йому привласнюється нижчий пріоритет, а наступному за ним по номеру каналу — вищий. Причому циклічна зміна пріоритетів відбувається після кожного циклу прямого доступу.

Режим розширеного запису (EW=i) збільшує за рахунок зсуву переднього фронту тривалість сигналів 1/OW і MEMW, які генеруються КПДП. Це дозволяє ВП, що формує сигнал READY по фронту сигналу запису, зменшити час охолодження і збільшити швидкість обміну.

При TCS= 1 поява сигналу TC в одному з каналів скидає відповідний розряд D3 — DO, у результаті чого канал відключається. Подальша робота цього каналу можлива після перезавантаження регістра режиму. Якщо TCS=0, то поява сигналу TC не впливає на розряд дозволу роботи каналу і закінчувати передачу повинен ВП за рахунок припинення дії сигналу DRQ.

У режимі автозавантаження (AL= 1) може працювати тільки другий канал, використовуючи вміст своїх внутрішніх регістрів і внутрішніх регістрів третього каналу. Після передачі даних відповідно до параметрів регістрів другого

каналу і появи сигналу TC параметри з регістрів третього каналу автоматично завантажуються в регістри другого каналу; прапорець відновлення (UF) у регістрі стану каналів встановлюється в 1. Потім передача даних продовжується відповідно до нових параметрів регістрів другого каналу, а наприкінці першого ж циклу прямого доступу з новими параметрами прапорець UF скидається. Режим автозавантаження дозволяє організувати повторювані пересилання блоків даних з однаковими параметрами з'єднувати кілька блоків з різними параметрами.

У початковому стані S1 запрограмований на визначений режим КПДП і очікує запит DRQ від ВП. Переходячи в стан S0, він генерує сигнал HRQ і очікує надходження від МП сигналу HLDA. Після надходження сигналу підтвердження HLDA починається цикл обміну. У стані S1 формується сигнал AEN для блокування інших пристроїв системи від шин даних і керування, видається код молодших розрядів на виходи A (7 — 0), а код старших розрядів — на виходи D(7 — 0).

Видача старших розрядів адреси супроводжується стробуючим сигналом ADSTB для запису їх у зовнішній буферний регістр. У стані S2 генерується сигнал DACK, що вказує на початок обміну, а також формуються пари сигналів MEMR, I/OR і MEMW, I/OW, які визначають напрямок обміну. У стані S3 відбувається передача даних у ЗП або ВП.

Стан S4 завершує цикл прямого доступу. У цьому стані при передачі останнього байта видається сигнал TC, а у випадку кінця блоку — сигнал MARK. При необхідності узгодження швидкодії ЗП і ВП за допомогою сигналу READ Y між станами S<sub>3</sub> і S<sub>4</sub> вводиться необхідне число станів чекання SW. У режимі контролю перехід у стан SW не дозволяється. Часова діаграма КПДП в циклі запису представлена на рисунку 77:

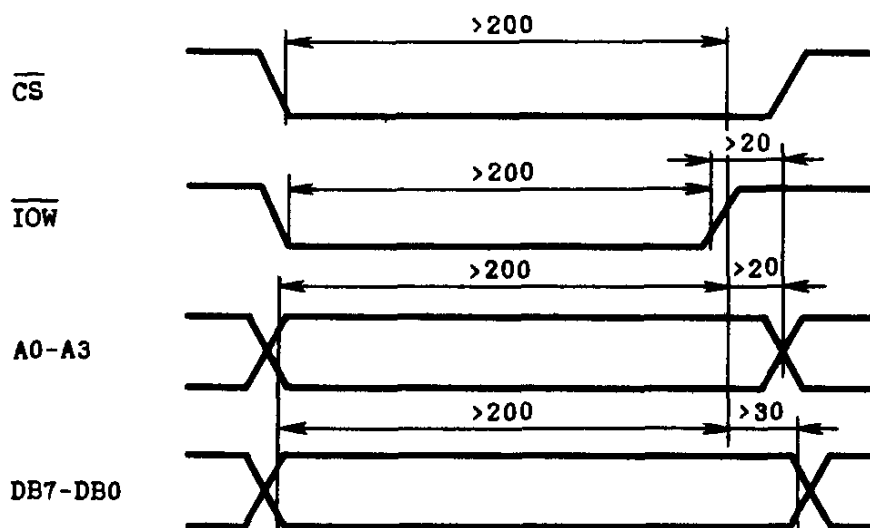


Рисунок 77 – Часова діаграма КПДП в циклі запису

В початковому стані S1 запрограмований на визначений режим КПДП і очікує запит DRQ від ВП. Переходячи в стан S0, він виробляє сигнал HRQ і очікує надходження від МП сигналу HLDA. Після надходження сигналу

підтвердження HLDA починається цикл обміну. У стані S1 формується сигнал AEN для блокування інших пристроїв системи від шин даних і керування, видається код молодших розрядів на виходи A (7 — 0), а код старших розрядів — на виходи D(7 — 0). Часову діаграму роботи КПДП в циклі читання показано на рисунку 78:

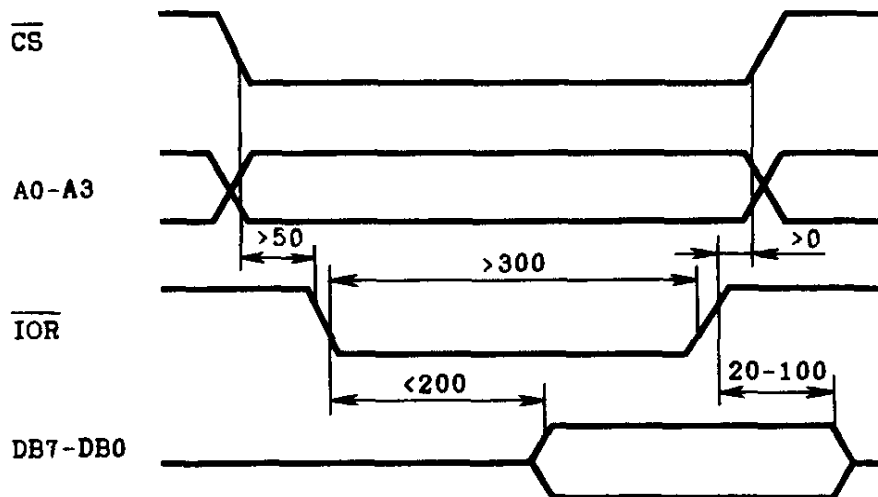


Рисунок 78 – Часова діаграма роботи КПДП в циклі читання

Видача старших розрядів адреси супроводжується стробуючим сигналом ADSTB для запису їх у зовнішній буферний регістр. У стані S2 виробляється сигнал DACK, що вказує на початок обміну, а також формуються пари сигналів MEMR, I/OR і MEMW, I/OW, що визначають напрямок обміну. У стані S3 відбувається передача даних у ЗП чи ВП. Стан S4 завершує цикл прямого доступу. У цьому стані при передачі останнього байта видається сигнал TC, а у випадку кінця блоку — сигнал MARK. При необхідності узгодження швидкодії ЗП і ВП за допомогою сигналу READ Y між станами S<sub>3</sub> і S<sub>4</sub> вводиться необхідне число станів очікування SW. У режимі контролю перехід у стан SW не дозволяється. Часову діаграму роботи КПДП в циклах обміну представлено на рисунку 79:

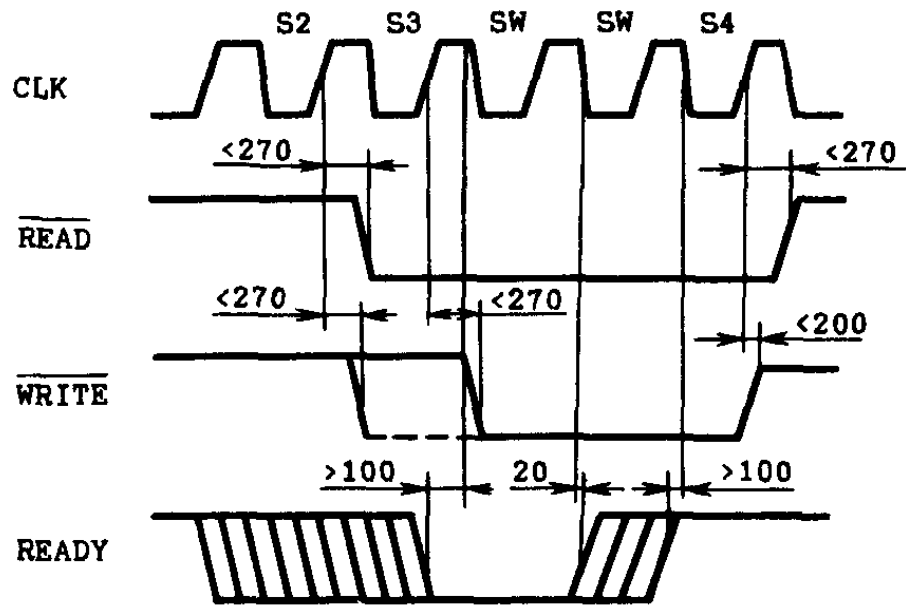


Рисунок 79 – Часова діаграма роботи КПДП в циклах обміну

Підключення КПДП до системної шини представлено на рисунку 80:

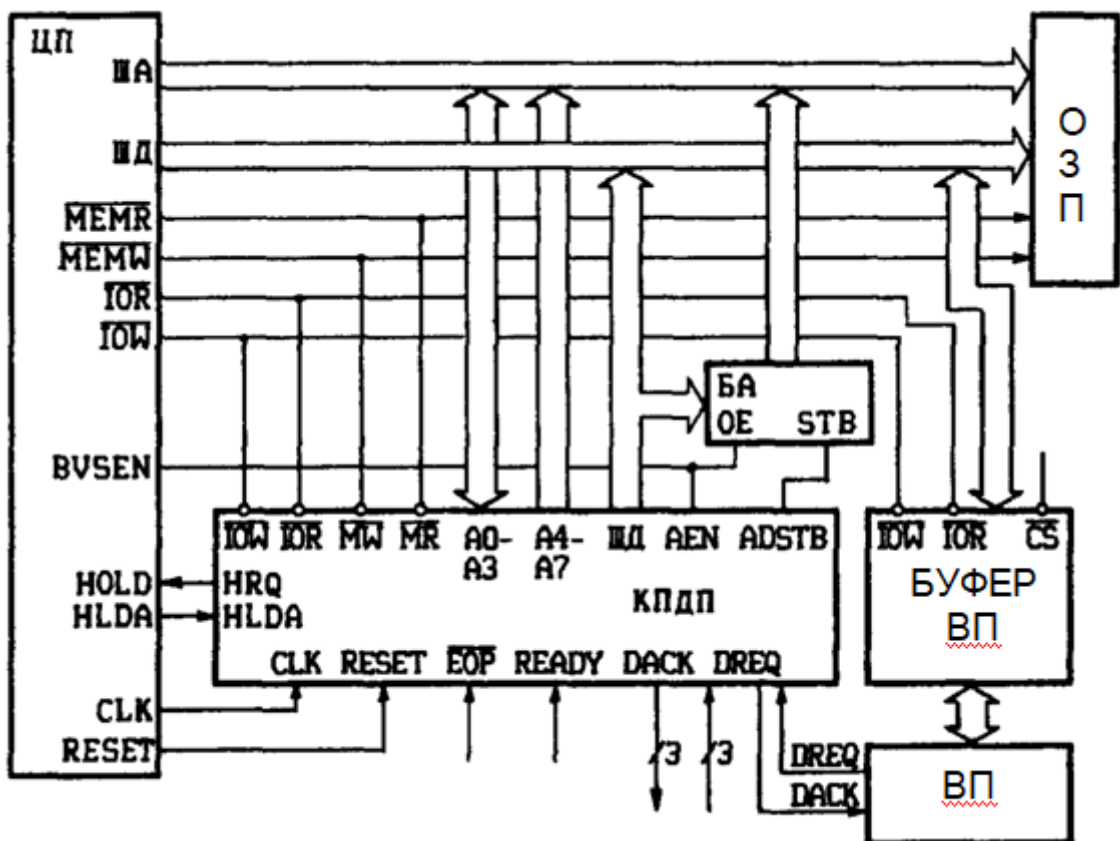


Рисунок 80 – Підключення КПДП до системної шини

Графічне зображення корпусу мікросхеми КПДП представлено на рисунку 81:

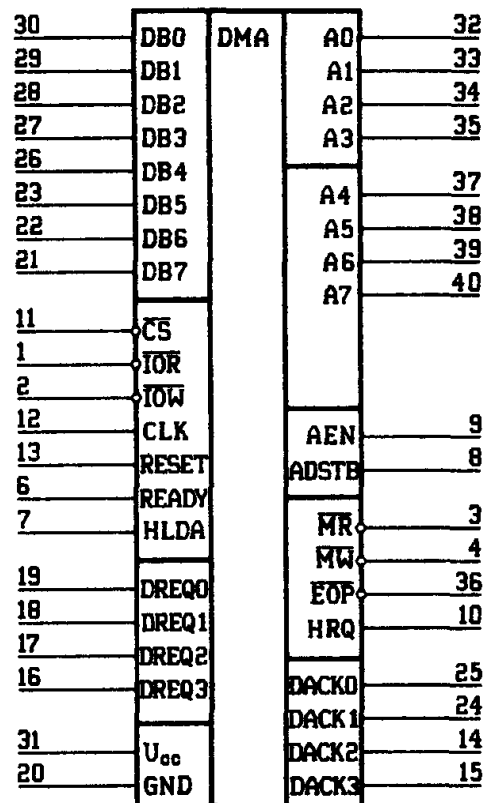


Рисунок 81 – Графічне зображення корпусу мікросхеми КПДП

Фрагмент програми початкового встановлення КПДП при звертанні до нього як до зовнішнього пристрою може мати наступний вигляд:

#### Приклад програмування КПДП

```

DI ;заборона переривань
MVI A,28H ; завантаження керуючого слова у регістр режиму
; звичайна передача, поточна адреса-зменшення,
; автоініціалізація- заборонена, режим-перевірка

OUT 0BH ; до регістра режиму
MVI A,1AH ; завантаження керуючого слова
; DACK- вис. рівень, DREQ-вис. рівень,затримка,при
; запису, циклічний зсув пріоритетів, стискання часу
; передачі, блокування контролера, ;дозвіл фіксації адреси,
; заборона передачі; пам'ять-пам'ять

OUT 08H ; у регістр команд
MVI A,00H ; завантаження молодшого байта адреси
OUT 00H(0кан.) ; масиву в порт
; 02H(1 кан.)
; 04H(2кан.)
; 06H(3кан.)

MVI A,40H ;завантаження старшого байта адреси

```

```

OUT  00H(0кан.) ; масиву в порт
MVI  A,01H      ; завантаження номера сторінки пам'яті
OUT  87H(0кан.) ; у порт каналу
                        83H(1кан.)
                        81H(2кан.) ;
                        82H(3кан.)
MVI  A,F4H      ; завантаження молодшого байта лічильника циклів
OUT  01H(0кан.) ; у порт каналу
                        03H(1кан.)
                        05H(2кан.)
                        07H(3кан.)
MVI  A,00H      ; завантаження старшого байта лічильника циклів
OUT  01H        ; у порт каналу
                        03H(1кан.)
                        05H(2кан.)
                        07H(3кан.)

EI          ;дозвіл переривань

```

У програмі відбувається налаштування каналу на запис масиву в область ОЗП. Для передачі N байт у лічильник каналу завантажується число N- 1. Після програмного налаштування каналу аналогічно набудовуються інші канали. У регістр режиму завантажується керуюче слово. Для попередження помилок перед програмуванням КПДП або читанням вмісту його регістрів необхідно забороняти переривання.

Підключення контролера до системної шини (рис. 80). Вісім старших розрядів адреси видаються на ШД і повинні бути записані сигналом ADSTB в зовнішній регістр. Лінія AEN використовується для того, щоб розряди адреси залишилися чинними на ША протягом трьох тактових періодів циклу ПДП. Лінії A7 - A0 підключаються безпосередньо до ША. Сигнали MEMR, MEMW, IOR, IOW керують у циклах ПДП відповідно ОЗП і буфером ВП.

За допомогою емулятора програмованого контролера прямого доступу до пам'яті I8257 прослідкуємо процес програмування пристрою. Загальний вигляд емулятора представлений на рисунку 82 (DMA\DMA\DMA.exe).

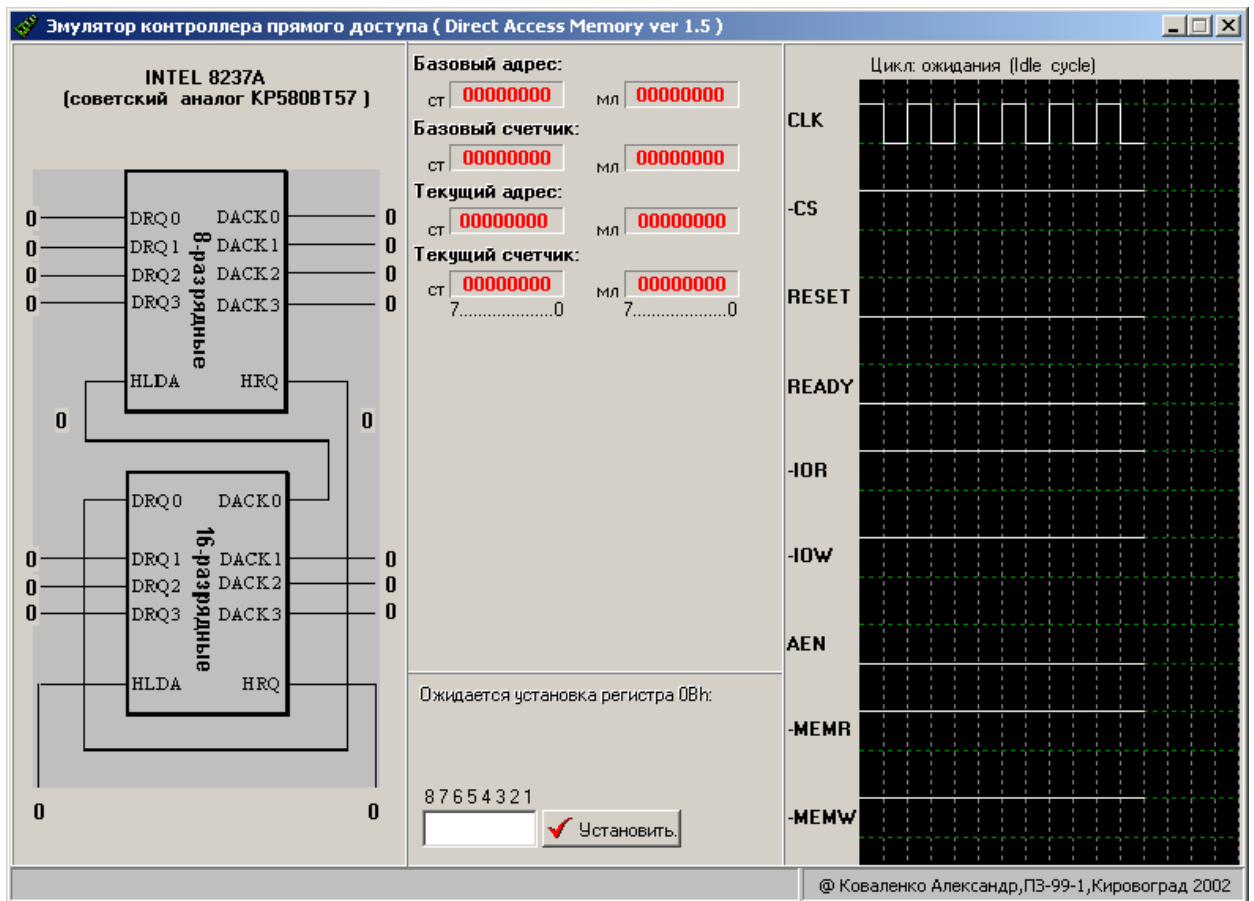


Рисунок 82 - Загальний вигляд емулятора контролера I8257

Завантаження програми до емулятора (приклад програми та послідовність завантаження команд вказана вище)

1. Для початку роботи емулятора необхідно завантажити послідовно всі керуючі слова вибраного режиму роботи. Керуючі слова та команди завантажуються послідовно одне за одним у вікно «87654321» відповідно підказці (зверху є підказка яке керуюче слово очікується).

2. Після введення кожного керуючого слова необхідно натиснути кнопку «Установить». У інформаційних віконцях вказується значення молодшого та старшого байту адреси масиву, значення молодшого та старшого байту адреси лічильника циклів.

3. Після завантаження керуючих слів КІДП почне свою роботу, в додатковому вікні (рис.83) «Окно просмотра передачи данных по DMA» відобразиться робота контролера, а у вікні зліва відображаються діаграми роботи контролера. Вихід з режиму виконується за допомогою кнопки зупинки процесу.

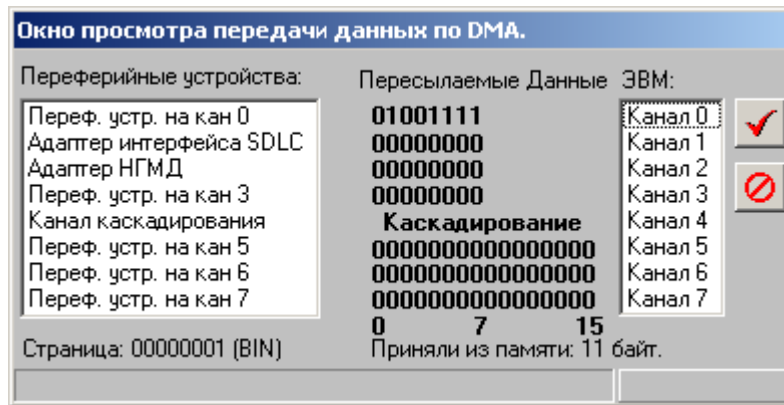


Рисунок 83 – Вікно демонстрації передачі масиву даних

## **Контрольні питання та завдання**

1. У яких режимах працюють КПДП і його функції в системі?
2. Визначте програмно-доступні регістри і їх адресацію.
3. Складіть схему підключення I8257 до шин адреси і даних МПС.
4. Складіть програму ініціалізації контролера для поблокового обміну по одному каналу.
5. Поясніть реалізацію режиму ПДП на I8257.
6. Поясніть адресацію до регістрів КПДП при програмуванні.
7. Як формується 16-бітна адреса КПДП при управлінні обміном?
8. Які пріоритети запитів підтримує I8257 ?
9. В якій послідовності необхідно проводити завантаження регістрів контролера при його програмуванні?
10. На лабораторній роботі використовуючи емулятор КПДП написати програму передачі або прийому даних до пам'яті відповідно варіанту і відлагодити її.

## Література

1. Архітектура комп'ютерних систем: конспект лекцій для студентів усіх форм навчання з курсу «Архітектура комп'ютерних систем» / Укладачі: Голотенко О.С. – Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2016 – 120 с.
2. Тарарака В.Д. Т19 Архітектура комп'ютерних систем: навчальний посібник. – Житомир : ЖДТУ, 2018. – 383 с
3. Сучасні напрямки комп'ютерної та мікропроцесорної техніки Розділ 1. Основні тенденції розвитку комп'ютерної і мікропроцесорної техніки. Розділ 2 Характеристики ARM і Cortex процесорів: конспект лекцій. [Електронний ресурс]: для студ. спеціальності 171 Електроніка, спеціалізації «Електронні компоненти та системи» /Т. О. Терещенко, Ю.С. Ямненко; КПІ ім. Ігоря Сікорського; уклад,– Електронні текстові данні 1 файл: 5,248 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2020. – 68 с.
4. Архітектура комп'ютерів. Особливості використання комп'ютерів в ІС : навчальний посібник / С. В. Кавун, І. В. Сорбат. – Харків : Вид. ХНЕУ, 2010. – 256 с.
5. Мартін Р. Чиста архітектура. – Фабула, 2019. – 368 с
6. Тарарака В.Д. Архітектура комп'ютерних систем: навчальний посібник. – Житомир: ЖДТУ, 2018. –383с
7. Матвієнко М.П. Архітектура комп'ютерів: Навчальний посібник / Матвієнко М.П., Розен В.П., Закладний О.М. – К.: Ліра-К, 2019. – 264 с.
8. Архітектура комп'ютерів: методичні рекомендації для студентів денної та заочної форми навчання за спеціальністю 123 “Комп'ютерна інженерія”/ уклад.Сидоренко В.В., Минайленко Р.М., Михайлов С.В. — Кропивницький: ЦНТУ, 2019. — 63 с.
9. Рикалюк Р.Є. Лабораторний (симуляційний) практикум з курсу «Архітектура комп'ютерних систем». Ч. 2 / Р. Є. Рикалюк, Л. Б. Галамага, Р. Г. Селіверстов. – Львів: Видавн. центр Львів. ун-ту, 2017. — 33 с
10. Комп'ютерна схемотехніка та архітектура комп'ютерів. Лабораторний практикум/ Укладачі В.М. Єфимець, Є.В. Красовська та ін. – К: НАУ, 2013. - 64с.
11. Bartlett J. Programming from the Ground Up. — [http://www.freebookcentre.net/ComputerScience-BooksDownload/Programming -fromthe-Ground-Up-\(J.-Bartlett\).html](http://www.freebookcentre.net/ComputerScience-BooksDownload/Programming-fromthe-Ground-Up-(J.-Bartlett).html)
12. Світ електронних схем. [Електронний ресурс] – Режим доступу: [ttp://ua.nauchebe.net](http://ua.nauchebe.net)
13. Дистанційна освіта ЦНТУ. – URL: <http://moodle.kntu.kr.ua/my/>