

Міністерство освіти і науки України  
Запорізький державний технічний університет

## **КОНСПЕКТ ЛЕКЦІЙ**

### **З КУРСУ**

## **“ЕОМ ТА МІКРОПРОЦЕСОРНІ СИСТЕМИ”**

**для студентів професійного спрямування**

**6.0804 “Комп’ютерні науки”**

**спеціальності 8.080403**

**“Програмне забезпечення автоматизованих систем  
усіх форм навчання**

Конспект лекцій з дисципліни “ЕОМ та мікропроцесорні системи” для студентів професійного спрямування 6.0804 “Комп’ютерні науки” спеціальності 8.080403 “Програмне забезпечення автоматизованих систем” усіх форм навчання / Укладач С.К.Корнієнко. - Запоріжжя: ЗДТУ, 2001. - 66 с.

Укладач:

Корнієнко С.К., к.т.н

Рецензент:

Арсеньєва С.І., к.т.н.

Відповідальний  
за випуск:

Крищук В.М., к.т.н.

Затверджено  
на засіданні кафедри КВР  
Протокол № 5 від 28.01.2001р.

Затверджено  
на засіданні Ради РПФ  
Протокол № 7 від 24.05.2001р.

## ЗМІСТ

|                                                                  |           |
|------------------------------------------------------------------|-----------|
| <b>ВСТУП</b> .....                                               | <b>5</b>  |
| <b>1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА МІКРОПРОЦЕСОРНИХ СИСТЕМ</b> .....   | <b>6</b>  |
| 1.1 Принципи побудови мікропроцесорних систем.....               | 6         |
| 1.2 Класифікація мікропроцесорних засобів.....                   | 7         |
| 1.3 Поняття архітектури МПС.....                                 | 8         |
| <b>2 ОРГАНІЗАЦІЯ 8-РОЗРЯДНИХ МІКРОПРОЦЕСОРІВ</b> .....           | <b>9</b>  |
| 2.1 Архітектура мікропроцесора 18080.....                        | 9         |
| 2.1.1 Блок реєстрів.....                                         | 9         |
| 2.1.2 Блок АЛП.....                                              | 14        |
| 2.1.3 Двонапрямлена магістраль даних.....                        | 16        |
| 2.1.4 Блок управління та синхронізації.....                      | 16        |
| 2.2 Виконання команд в мікропроцесорі 8080.....                  | 19        |
| 2.2.1 Машинні цикли та такти.....                                | 19        |
| 2.2.2 Слово стану.....                                           | 20        |
| 2.2.3 Системний контролер.....                                   | 21        |
| 2.2.4. Виконання команд.....                                     | 23        |
| 2.3 Мікропроцесор 8085А.....                                     | 25        |
| <b>3 ЗАГАЛЬНІ ПИТАННЯ ПОБУДОВИ МІКРОПРОЦЕСОРНИХ СИСТЕМ</b> ..... | <b>27</b> |
| 3.1 СИСТЕМА ПАМ'ЯТІ МПС.....                                     | 27        |
| 3.1.1 Класифікація запам'ятовуючих пристроїв.....                | 27        |
| 3.1.2 Основні характеристики запам'ятовуючих пристроїв.....      | 28        |
| 3.1.3 Оперативні запам'ятовуючі пристрої (ОЗП).....              | 29        |
| 3.1.4 Постійні запам'ятовуючі пристрої (ПЗП).....                | 29        |
| 3.1.5 Нарощування пам'яті мікро-ЕОМ.....                         | 30        |
| 3.1.6 Підключення ПЗП.....                                       | 31        |
| 3.1.7 Розшарування звернень до пам'яті.....                      | 33        |
| 3.1.8 Кеш-пам'ять.....                                           | 34        |
| 3.1.9 Фізична та логічна структура диска.....                    | 36        |
| 3.2 РЕЖИМИ ОБМІНУ ІНФОРМАЦІЄЮ.....                               | 39        |
| 3.2.1 Протоколи обміну. Арбітраж.....                            | 39        |
| 3.2.2 Програмний обмін інформацією.....                          | 41        |
| 3.2.3 Обмін інформацією в режимі перервань.....                  | 42        |
| 3.2.4 Прямий доступ до пам'яті.....                              | 43        |
| 3.3 ІНТЕРФЕЙСИ МІКРОПРОЦЕСОРНИХ СИСТЕМ.....                      | 44        |

|                                                               |           |
|---------------------------------------------------------------|-----------|
| 3.3.1 Призначення інтерфейсів .....                           | 44        |
| 3.3.2 Класифікація інтерфейсів .....                          | 45        |
| 3.3.3 Інтерфейс із ізольованою шиною .....                    | 46        |
| 3.3.4 Інтерфейс із загальною шиною .....                      | 47        |
| 3.3.5 Апаратний склад інтерфейсу .....                        | 49        |
| <b>4 ОРГАНІЗАЦІЯ 16-РОЗРЯДНОГО МІКРОПРОЦЕСОРА 18086 .....</b> | <b>53</b> |
| 4.1 ОСНОВНІ ХАРАКТЕРИСТИКИ МП .....                           | 53        |
| 4.2 АРХІТЕКТУРА 18086 .....                                   | 53        |
| 4.2.1 Пристрої спряження з каналом .....                      | 53        |
| 4.2.2 Операційний пристрій .....                              | 56        |
| 4.2.3 Регістри МП .....                                       | 56        |
| 4.3 АДРЕСАЦІЯ ПАМ'ЯТІ .....                                   | 58        |
| 4.3.1 Сегментація .....                                       | 58        |
| 4.3.2 Фізичні та логічні адреси .....                         | 59        |
| 4.3.3 Розподіл пам'яті .....                                  | 60        |
| 4.4 РЕЖИМИ РОБОТИ .....                                       | 61        |
| 4.6 СИСТЕМА ТА ФОРМАТ КОМАНД 18086 .....                      | 61        |
| <b>5 RISC-АРХІТЕКТУРА .....</b>                               | <b>64</b> |
| 5.1 ЗАГАЛЬНІ ПРИНЦИПИ РЕАЛІЗАЦІЇ .....                        | 64        |
| 5.2 ТРАНСП'ЮТЕРИ .....                                        | 65        |
| <b>ПЕРЕЛІК ПОСИЛАНЬ .....</b>                                 | <b>66</b> |

## ВСТУП

При створенні автоматизованих систем різного призначення використовують два класи засобів цифрової техніки:

- пристрої з жорсткою структурою, або з "*жорсткою*" логікою;

- програмуємі пристрої, або пристрої з "*програмуємою*" логікою.

Пристрої з "жорсткою" логікою реалізуються на основі комбінаційних логічних схем. Закон функціонування таких пристроїв визначається способом з'єднання логічних елементів. Зміна закону функціонування потребує внесення змін у схему та, відповідно до конструкції пристрою.

Ускладнення електронних схем, реалізація їх на базі *високорівневих інтегрованих схем (ВІС)* призводить до все більшої їх спеціалізації. Звідси зужується область їх застосування, зменшується об'єм їх випуску та, отже, різко зростає їх вартість.

Ця обставина призвела до появи нового підходу до проектування цифрових схем, заснованого на використанні програмуємих *ВІС*. Структура таких *ВІС* достатньо універсальна, а функціональні особливості визначаються не самою схемою, а спеціальною програмою.

Такі універсальні програмно-управляємі *ВІС* отримали назву *мікропроцесорів (МП)*. По своїм можливостям *МП* представляють спрощений варіант процесорного пристрою універсальної *ЕОМ*.

До складу мікропроцесора зазвичай входять *аріфметико-логічний пристрій* та *пристрій управління*. Таким чином, *МП* - це лише частина обчислювального пристрою, для нормальної організації якого потрібні *пам'ять* та *пристрої вводу-виводу* інформації.

*Мікропроцесорна система* - це програмно-апаратний комплекс, призначений для реалізації функцій обробки даних, контролю або управління та побудований на одному або декількох мікропроцесорах.

# 1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА МІКРОПРОЦЕСОРНИХ СИСТЕМ

## 1.1 Принципи побудови мікропроцесорних систем

### Модульна організація систем

**МПС** будуються на основі конструктивно, функціонально та електрично закінчених обчислювальних пристроїв, які називаються *модулями*.

Модульний підхід при проектуванні **МПС** сприяє стандартизації елементів та скороченню витрат на проектування, а також спрощує нарощування потужності та реконфігурацію систем, відсовує час морального старіння технічних засобів.

### Магістральний спосіб обміну інформацією

Обмін інформацією між модулями в системі здійснюється за допомогою шин.

**Шина** - це сукупність ліній передачі інформації, які об'єднані за функціональним призначенням.

**Магістраль** - це сукупність усіх шин **МПС**. Часто терміни "*шина*" та "*магістраль*" використовуються як синоніми.

При цьому всі модулі системи постійно підключені до магістралі, але в кожний момент часу обмін інформацією може відбуватися тільки між двома модулями за рахунок використання спеціальних буферних пристроїв з трьома стійкими станами.

Шини **МПС** підрозділяються на *шини даних (DB -Data Bus)*, *адреси (AD - Address Bus )* та *управління (Control Bus)*.

### Мікропрограмна організація управління

Пристрої з програмною логікою функціонують під управлінням програми, яка зберігається в пам'яті. Виконання кожної команди розкладається на сукупність елементарних дій - *мікрооперацій* (зсув регістра, установка в нуль тощо). Кожній мікрооперації відповідає певна мікрокоманда, сукупність яких має назву *мікропрограма*. Мікропрограми зберігаються в пристрої керування МП.

## 1.2 Класифікація мікропроцесорних засобів

*МПС* поділяються на такі класи:

- система на базі секційованих мікропроцесорних *ВІС* з мікропрограмним керуванням;
- системи на основі однокришталевих мікропроцесорів з програмним керуванням;
- системи зі скороченим набором команд.

В *багатокришталевих мікропроцесорах* використовується сукупність *ВІС*, які реалізують окремі вузли *МП*. Зазвичай *АЛП* в таких *МП* виконуються у вигляді окремих секційних *ВІС* з фіксованою розрядністю. Збільшення розрядності досягається простим з'єднання *ВІС*. В *секційних МП* використовується мікропрограмний спосіб управління, при якому команди реалізуються як послідовність мікрокоманд, записаних у спеціальному *ПЗП*. Зміст такого *ПЗП* можна змінювати за бажанням користувача, при цьому змінюється й система команд *МП*. *МПС* на основі секційних *МП* мають більшу гнучкість, більш високу швидкість, але проектування таких систем потребує високої кваліфікації розробника.

*Однокришталеві мікропроцесори* відрізняються фіксованою структурною розрядністю та системою команд, які користувач не може змінити на свій погляд. До складу такої *ВІС* входять *АЛП*, *ПУ* та схеми погодження.

*Однокришталеві мікро-ЕОМ (ОМЕОМ)* представляють собою функціонально закінчений обчислювальний пристрій, реалізований на одній *ВІС*.

*ОМЕОМ*, які вбудовуються в апаратуру користувача та призначені для управління нею в реальному масштабі часу, звуться програмованими мікроконтролерами.

Системи зі скороченим набором команд (*RISC - Reduced Instruction Set Computer*) відрізняються підвищеною швидкодією за рахунок скорочення кількості команд та апаратної реалізації багатьох команд. *RISC* – буде розглянута у розділі 5.

### 1.3 Поняття архітектури МПС

Архітектура *МПС* - це функціональні можливості апаратних засобів *МПС*, які використовуються для представлення програм та даних, а також для керування процесом обчислювання. Іншими словами, це представлення *МПС* з точки зору користувача.

Існує два основних типи архітектур *МПС*:

- *фоннейманівська*, в якій для зберігання програм і даних використовується єдиний простір пам'яті, і нема ніяких ознак, які вказують на тип інформації, яка зберігається в чарунці пам'яті. Зміст чарунки інтерпретується при цьому самим *МП*;

- *гарвардська*, в якій пам'ять програм та пам'ять даних поділені і мають свої особисті адресні простори та засоби доступу до них.

## 2 ОРГАНІЗАЦІЯ 8-РОЗРЯДНИХ МІКРОПРОЦЕСОРІВ

### 2.1 Архітектура мікропроцесора I8080

Мікропроцесор реалізований на основі загальної внутрішньої шини даних і містить такі функціональні вузли (рис. 2.1):

- блок регістрів з адресною логікою;
- блок АЛП;
- двонапрямлена буферизована внутрішня магістраль даних;
- блок керування та синхронізації.

#### 2.1.1 Блок регістрів

Блок регістрів містить шість 16-розрядних регістрів. Три з них можуть використовуватися як шість окремих 8-розрядних програмно-доступних регістрів загального призначення (**РЗП**) **B, C, D, E, H, L** для зберігання операндів або як три 16-розрядні програмно-доступні регістрові пари **BC, DE, HL** для зберігання адрес або двобайтових операндів (рис. 2.2).

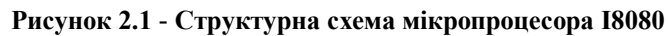
При виконанні арифметичних і логічних операцій з регістровою адресацією в **РЗП** зберігаються 8-розрядні операнди, які передаються в блок **АЛП** для участі в операції. Другий операнд і результат операції зберігаються в блоці **АЛП**.

Пари регістрів **BC, DE** і **HL** зазвичай використовують в якості регістрів-вказівників опосередкованої адресації. При цьому основним регістром-вказівником є пара **HL**.

Крім того, при складанні двох 16-розрядних чисел в регістрових парах можуть зберігатися 16-розрядні операнди. В цьому випадку зміст будь-якої регістрової пари складається в блоці **АЛП** зі змістом пари **HL**, і результат записується в регістрову пару **HL**.

Зміст кожного з регістрів можливо переслати в блок **АЛП** або пам'ять через 8-розрядний мультиплексор і внутрішню магістраль даних. Аналогічно відбувається завантаження регістрів.





**Рисунок 2.1 - Структурна схема мікропроцесора І8080**



Рисунок 2.2 - Блок регістрів мікропроцесора

Крім того, зміст кожного регістра (пари регістрів) можливо програмно збільшити або зменшити на одиницю. Ця зміна здійснюється за допомогою адресної логіки.

***PC (Program Counter)*** - **програмний лічильник (лічильник команд)** використовується для зберігання поточної адреси наступного байту команди.

Оскільки програма - це послідовність команд, які зберігаються в пам'яті **мікро-ЕОМ**, для її коректного виконання необхідно, щоб команди знаходились у суворо презначеній послідовності. При цьому саме на лічильник команд покладається функція стеження за тим, яка команда виконується, а яка підлягає виконанню наступною.

Перед виконанням програми в лічильник команд завантажується адрес першої команди програми. Зміст **PC** після вибірки кожного байту команди автоматично збільшується схемою адресної логіки.

Зміст **PC** може бути змінено під впливом самої програми. В результаті відбувається примусова передача керування тій області пам'яті, в якій міститься наступна ділянка програми. Завантаження та видача змісту відбуваються через мультиплексори та внутрішню магістраль даних. Після включення живлення або виконання команди **RESET** виконується примусове обнулення **PC**.

**SP (Stack Pointer)** - **покажчик стека** - використовується при запису в стек або читання з нього.

**Стек** - пам'ять зі спрощеною формою адресації. Це оперативна пам'ять магазинного типу, в якій запис або вибірка слова виконується за принципом: останній записаний елемент вибирається з пам'яті першим (**LIFO - last in, first out**).

В I8080 стек організується наступним чином (рис. 2.3). В **ОЗП** команди розміщуються в чарунки з молодшими, послідовно зростаючими адресами. Стек використовує чарунки зі старшими адресами. По мірі заповнення стека займаються чарунки з адресами, які послідовно убувають. Таким чином, адреси цих двох областей пам'яті змінюються назустріч один одному.

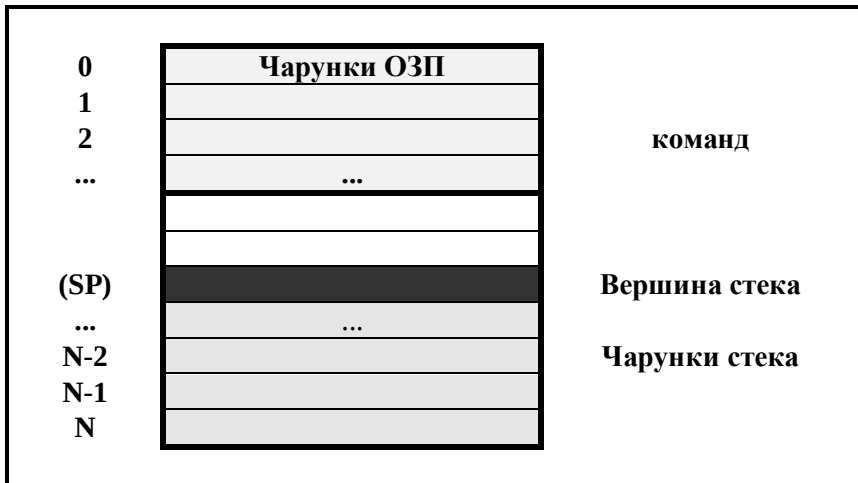


Рисунок 2.3 - Організація стека

Показчик стека містить так звану адресу входу до стека (*вершина стека*), тобто адресу чарунки області пам'яті, до якої було зроблене останнє звернення. Зміст **SP** автоматично за допомогою адресної логіки зменшується на 1 перед кожним занесенням слова до стека та збільшується на 1 після кожного витягання слова зі стека.

Показчик стека спрощує адресацію пам'яті, але виключає можливість звернення до будь якої чарунки пам'яті. Крім цього, з огляду відсутності у коді команди запису до стеку (**PUSH**) та читання із стеку (**POP**) адресного поля, зменшується розрядність цих команд та час їх виконання.

Показчик стека встановлюється програмним шляхом (по команді **LXI SP** в старший байт **SP** заноситься третій байт команди, в молодший байт **SP** - другий байт команди).

**Пара реєстрів WZ** використовується для тимчасового зберігання другого та третього байтів команди, а також для обміну інформацією між реєстровими парами (команда **XCHG**). Реєстри програмно недоступні.

**Адресна логіка** призначена для зберігання та видачі на шину адрес даних та команд. Вона містить:

- буферний реєстр адреси (**РА**);
- логічну схему інкремента-декремента (**СІД**);
- адресний буфер (**БА**).

**Реєстр адреси** приймає та зберігає адресу з будь якого 16-бітового реєстра. Його вихід зв'язаний зі входами **СІД** та **БА**.

**Схема інкремента-декремента** представляє собою схему швидкого переносу (займу), за допомогою якої є можливість змінювати на 1 зміст будь якої реєстрової пари.

**Адресний буфер** представляє собою 16 вихідних формувачів з трьома станами та призначений для видачі адреси на вихід адресної магістралі **A15-A0**. Наявність третього (високоомного) стану в **БА** дозволяє безпосередньо підключати **МП** до системної адресної магістралі **мікро-ЕОМ**.

### 2.1.2 Блок АЛП

Блок АЛП призначений для виконання найпростіших арифметичних та логічних операцій над 8-розрядними двійковими числами (складання, віднімання, зсув, порівняння, логічне множення тощо). Більш складні операції, наприклад, множення, ділення тощо виконуються програмно.

Крім того, *АЛП* може працювати і з 16-розрядними двійковими числами. При цьому операції виконуються в два етапи: спочатку виконуються операції над молодшими, а потім над старшими байтами. На другому етапі виконання команди в операції бере участь перенос (позика), отриманий в операції з молодшими байтами та запам'ятований у тригері переносу *С*.

Інформація оброблюється в *АЛП* з використанням акумулятора (*А*), регістра ознак (*F*), регістра тимчасового зберігання *БР2* та регістра акумулятора *БР1*. При виконанні операції один з операндів пересилається з акумулятора до регістра *БР1*; інший - надходить з пам'яті або *РЗП* через внутрішню магістраль даних до регістру *БР2*, звідки надходить до *АЛП* через кодоперетворювач у прямому або додатковому коді в залежності від операції, яка виконується.

Регістр тимчасового зберігання *БР1* дозволяє запобігти виникненню "гонок", коли будь-який з *РЗП* використовується в одній операції в якості регістра-операнда та в якості регістра-результата.

Результат операції передається через внутрішню магістраль даних до акумулятора, а ознака результату записується до регістра ознак *F*.

Регістр ознак (*F*) (або прапорців) (мал. 2.4) призначений для зберігання двійкових ознак, або прапорців.



**Рисунок 2.4 - Формат регістра ознак.**

Двійкові ознаки мають такий сенс:

- C (Carry)*** - ознака переносу, встановлюється в "1" у випадку переносу 1 зі старшого розряду регістра *A* при виконанні деяких операцій над даними. При відсутності переносу ознака *C* скидується (фактично 1 переносу зі старшого розряду акумулятора запам'ятовується в тригері переносу)
- P (parity)*** - прапорець парності, який приймає значення 1, якщо сумарне число одиничних розрядів у регістрі *A* являється парним
- AC (Auxiliary Carry)*** - ознака додаткового переносу з молодшого напівбайта числа до старшого. Використовується в командах десятикової арифметики
- Z (zero)*** - ознака нульового результату
- S (sign)*** - ознака знака, яка приймає значення одиниці, як що зміст *A* стає негативним у результаті виконання деякої операції

**Слово стану програми *PSW* (Program Status Word)** - це 16-розрядне слово, старший байт якого дорівнює змісту акумулятора, а молодший - змісту регістра ознак *F*:

$$PSW = [A] + [F].$$

За допомогою регістра *F* здійснюється перехід в програмах (порушення природної послідовності виконання команд).

Схема десяткової корекції (*СДК*) призначена для перетворення двійково-кодованого результату в двійково-десятковий код при обробці двійково-десяткових чисел.

### ***2.1.3 Двонапрявлена магістраль даних***

Призначена для організації зв'язку МП з іншими мікросхемами, які входять до складу мікро-ЕОМ. Вона включає в себе внутрішню магістраль даних, буфер даних (*БД*) і з'єднана з виводами магістралі даних *D7-D0* МП.

8-розрядний двонапрявлений буфер даних з трьома станами призначений для розв'язки внутрішніх та зовнішніх магістралей даних. Він складається з буферного регістра даних та формувачів.

В режимі виведення інформація з внутрішньої магістралі завантажується в буферний регістр, а потім передається на зовнішню магістраль даних через формувачі. При вводі інформація з зовнішньої магістралі через формувачі безпосередньо передається на внутрішню магістраль. Буферний регістр даних при цьому вимикається. Він вимикається також при виконанні операцій, не пов'язаних з передачею інформації процесора.

### ***2.1.4 Блок управління та синхронізації***

БУС призначений для приймання команд, синхронізуючих та управляючих зовнішніх сигналів, а також для формування внутрішніх сигналів мікрооперацій.

Він містить регістр команд (*РК*), дешифратор команд (*ДшК*) та пристрій управління (*ПУ*).

*РК* призначений для приймання першого байта команди в якому міститься код операції. Код операції після дешифрації використовується для формування керуючих сигналів, під впливом яких виконуються мікрооперації в окремих вузлах МП.

*ПУ* містить виконану на програмованій логічній матриці управляючу пам'ять, в якій зберігаються мікропрограми окремих операцій. Користувач не може змінити зміст управляючої пам'яті, а значить, і склад команд. Приймання та видача синхронізуючих та керуючих сигналів здійснюється через виводи МП:

### **Виводи синхронізації:**

- CI,C2*** - входи двох неперекривних послідовностей синхроімпульсів зовнішнього тактового генератора
- SYN*** - вихід сигналу синхронізації, який визначає початок кожного машинного циклу. Використовується для ідентифікації дій, які виконуються в МП, та синхронізації поциклового управління процесором
- RESET*** - вхід сигналу початкової установки процесора. Після закінчення сигналу (його тривалість повинна бути не менше трьох періодів синхросигналів) програмний лічильник встановлюється в нульовий стан, та процесор починає виконувати програму з нульової адреси

### **Виводи керування чекання:**

- READ*** - вхід сигналу готовності, який вказує готове або ні зовнішній пристрій або пам'ять до обміну. Дозволяє синхронізувати обмін інформацією з зовнішніми пристроями або з пам'яттю більш низької швидкодії.
- WAIT*** - вихід сигналу, який вказує на те, що процесор знаходиться у стані **ЧЕКАННЯ**

### **Виводи керування записом та читанням із пам'яті або зовнішніх пристроїв:**

- WR*** - вихід сигналу управління записом в пам'ять або до зовнішнього пристрою

**DBIN** - вихід сигналу, який вказує на те, що магістраль даних знаходиться в режимі приймання. Використовується для управління читанням даних з пам'яті або з зовнішніх пристроїв

**Виводи керування перериваннями:**

**INT** - вхід сигналу запиту переривання, який надходить від зовнішнього пристрою та запитує обмін з МП у режимі переривання

**INTE** - вихід сигналу дозволу переривання, який вказує на те, що процесор готовий до обміну в режимі переривання (має змогу прийняти запит на переривання). Після переходу до обслуговування переривання на виході **INTE** встановлюється сигнал низького рівня, та запити на переривання не сприймаються. Здібність сприйняття процесором послідовних запитів встановлюється програмним шляхом.

**Виводи керування захопленням магістралей в режимі ПДП:**

**HOLD** - вхід сигналу запиту захоплення магістралей **D7-D0** та **A15-A0** зовнішніми пристроями. Призначається для управління в режимі ПДП. Під дією сигналу захоплення процесор переходить в стан **ЗАХОПЛЕННЯ**, в якому магістралі даних та адрес МП знаходяться в третьому стані, завдяки чому системна магістраль може використовуватися зовнішніми пристроями

**HLDA** - вихід сигналу підтвердження захоплення, вказує на те, що МП знаходиться в стані **ЗАХОПЛЕННЯ**. Магістралі даних та адреси при цьому відключені від виводів МП.

## 2.2 Виконання команд в мікропроцесорі BM80

### 2.2.1 Машинні цикли та такти

В МП I8080 операнди, коди виконуємих команд та результати обчислювань передаються по одній і тій же магістралі **D7-D0**. Щоб розрізняти призначення інформації, яка передається по шині даних, використовується суворо визначена послідовність виконання команд, яка задається синхроімпульсами **C1** та **C2** зовнішнього ГТІ.

Всі дії МП синхронізовані вкладеними циклами трьох рівнів: командними, машинними та мікротактами.

**Командний цикл** складається з декількох (від 1 до 5) машинних циклів, які позначаються **M1 - M5** та реалізують фазу вибірки та фазу виконання команди.

**Машинний цикл** - це час, потрібний МП для одного звернення до пам'яті або ПВВ для вибірки або запису одного байта інформації. Виключенням є команда **DAD** (складення змісту регістрової пари зі змістом **HL**). Вона містить три машинних цикли, але тільки в першому циклі **M1** відбувається звернення до пам'яті, а два інших відводяться для виконання команди без звернення до пам'яті або **PBB**.

Цикл **M1** - єдиний цикл, який реалізує фазу вибірки коду операції. Реалізація машинного циклу покладається на пристрій управління.

Машинний цикл складається з **машинних тактів** (мікротактів). **MT** - це інтервал часу, який дорівнює періоду слідування синхроімпульсів фази **C1**. При частоті імпульсів **C1**, яка дорівнює 2 МГц, тривалість **MT** дорівнює 0,5 мс. Один машинний цикл включає 3-5 машинних тактів **T1-T5**. Протягом одного **MT** виконується один мікронаказ **ПУ**.

Зазвичай цикл **M1** триває 4-5 тактів, цикли **M2-M5** містять 3 такти. Сигнал **SYN** видається в першому такті кожного машинного циклу.

Існує три виключення з цього положення:

- стан **ЧЕКАННЯ** ( $T_w$ ), в якому **МП** знаходиться в очікуванні операнда при відсутності готовності **ПВВ** або пам'яті до обміну;
- стан **ПДП** ( $T_{wh}$ ), в який **МП** переходить під впливом зовнішніх сигналів управління **МПС**;
- стан **ЗУПИН**, в який **МП** може бути введений за допомогою команди зупину **HLT**.

Ці три стани не зв'язані з тактовою частотою сигналів **C1** та **C2**, їх тривалість невизначена, тобто. залежить від зовнішніх, по відношенню до **МП**, подій. Ці стани тривають цілу кількість тактів.

### 2.2.2 Слово стану

Для нормального функціонування **МПС** недостатньо управляючих сигналів. Так, в даному наборі сигналів не має можливості відрізнити цикли звернення до пам'яті від циклів вводу-виводу. Розширення кількості управляючих сигналів зроблено за допомогою спеціального 8-розрядного слова стану **SW (Status Word)**, яке видається **ПУ МП** на шину даних в такті **T1** кожного машинного циклу.

Сигнали слова стану **SW** кодують тип машинного циклу, тобто. визначають вид інформації та напрямок її руху через шину даних (іншими словами, визначають, що **МП** буде робити в даному машинному циклі).

Всього **МП ВМ80** має 10 типів машинних циклів та, відповідно, 10 кодів слова стану, які ідентифікують ці цикли: вибірка коду операції (**MI**), читання та запис даних у пам'ять, читання та запис даних у стек, ввід і вивід даних у зовнішні пристрої, цикл обслуговування переривання, зупин, обслуговування переривання при роботі **МП** в режимі зупину.

Таким чином, управління **МПС** здійснюється генерацією управляючих впливів на двох рівнях:

- на рівні управляючих сигналів (мікронаказів), які вироблюються **ПУ МП** у кожному машинному такті;
- на рівні мінінаказів шляхом генерації слова стану у кожному машинному циклі.

### 2.2.3 Системний контролер

Оскільки слово стану видається на мультиплексовану шину стану/даних у першому такті, а повинно використовуватися протягом всього машинного циклу, його необхідно запам'ятати в спеціальному зовнішньому, по відношенню до **МП**, регістрі слова стану (**PrCC**).

Запис слова стану до **PzCC** здійснюється у другому такті по стробуючому імпульсу, який виробляється в момент збігу сигналів **SYN** та імпульсу **C1** (рис.2.5).

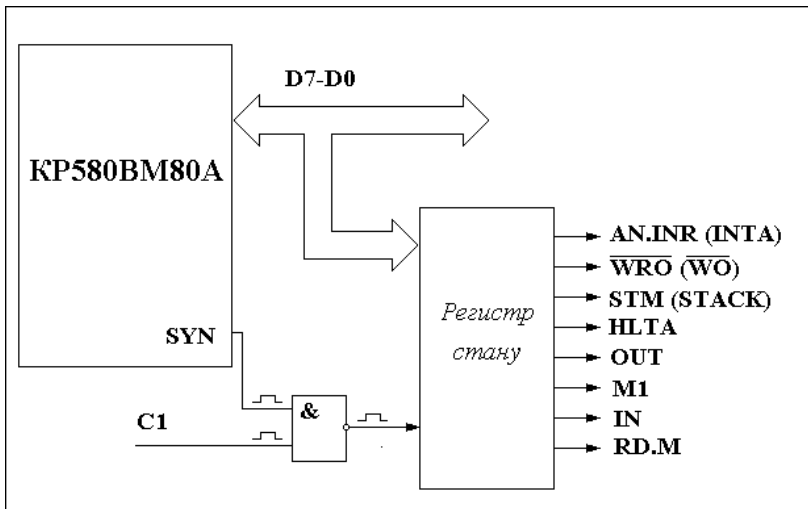


Рисунок 2.5 – Занесення слова стану до **PzCC**

На основі одержаної на виході *P2CC* інформації та управляючих сигналів **ПРИЙНЯТТЯ** (*RC.D*) та **ЗАПИС** (*WR*), які видаються МП, формуються взаємно виключаючі управляючі сигнали **ЧТЗП**, **ЗПЗП**, **ЧТВВ**, **ЗПВВ** та сигнал підтвердження переривання **ППР**. Формування цих сигналів виробляється за допомогою комбінаційної схеми.

Для спрощення апаратної реалізації шини управління зручно використовувати спеціалізовану **ВІС КР580ВК28** (рис.2.6) - системний контролер та шинний формувач (**СКФ**).

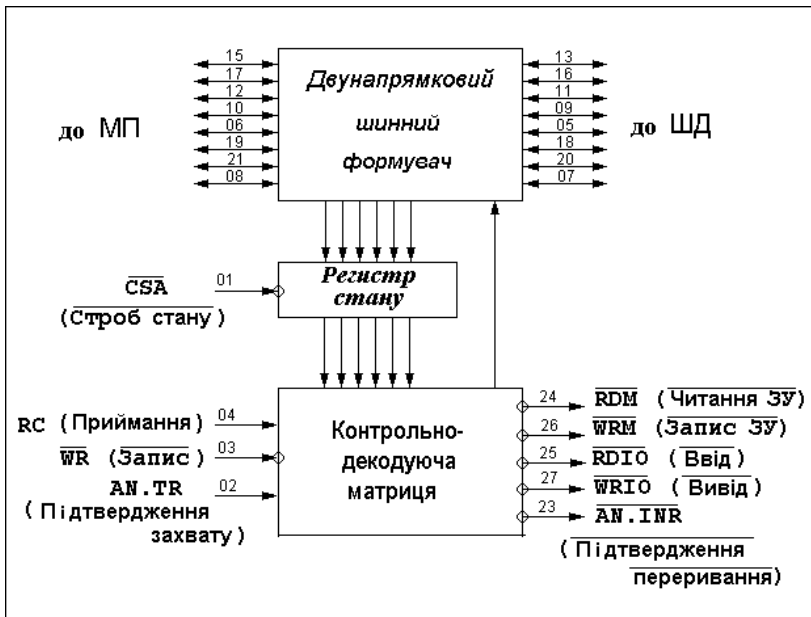


Рисунок 2.6 – Системний контролер

**Системний контролер** призначений для фіксації слова стану та виробки управляючих сигналів.

**Шинний формувач** призначений для буферизації шини даних та управління напрямком передачі даних.

### 2.2.4. Виконання команд

*Виконання команди здійснюється в такій послідовності:*

- 1) Зміст чарунки пам'яті, яка вказана програмним лічильником, читається з пам'яті до регістру команд МП.
- 2) Зміст програмного лічильника збільшується на одиницю
- 3) Код операції команди пересилається до дешифратора команд.
- 4) Після дешифрації коду операції та виробки управляючих сигналів здійснюється виконання команди. Для одноциклових команд виконання здійснюється в останніх двох тактах циклу вибірки (**MI**), для багатоциклових команд - в наступних циклах.

Вказана послідовність дій по тактам реалізується таким чином (рис. 2.7).

У такті **TI** за імпульсом **C2 МП** виробляє сигнал керування **SYN**, який, по-перше, ідентифікує інформацію на шині даних як слово стану процесора, а по-друге, свідчить про початок машинного циклу. Одночасно на шину адреси з програмного лічильника виставляється адреса чергової команди (байту команд).

У другому такті за імпульсом **C1** та сигналом **SYN** слово стану записується до системного контролера; за імпульсом **C2** сигнал **SYN** скидається, на виході **МП RC.D(DBIN)** з'являється сигнал високого рівня, на шині даних скидається слово стану, і вона переводиться до **режиму вводу**.

Одночасно **МП** перевіряє сигнали на входах **READY, HOLD** та тригері **HLTA**. В залежності від значення сигналів на цих виводах **МП** переходить в різні стани: чекання, захоплення або зупину. Якщо на виводі **READY, HOLD** надійшли сигнали **1,0**, а тригер **HLTA** встановлений в "0", мікропроцесор переходить до виконання такту **T3**. При відсутності готовності мікропроцесор переходить у стан чекання, в якому він виконує холості такти **Tw**.

У такті **T3** виконується читання інформації або запис, після чого шина даних переходить в високоімпедансний стан.

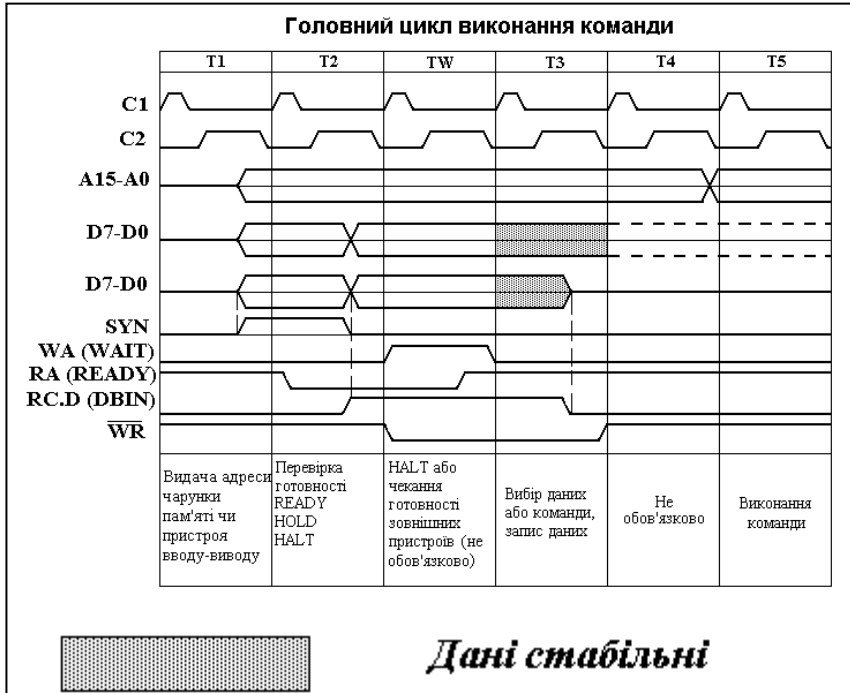


Рисунок 2.7 - Головний цикл виконання команди

Зазвичай до четвертого такту вже зміниться значення лічильника команд, і, таким чином, в ньому знаходиться адреса нового байту, яка надходить на шину адреси в наступному машинному циклі.

В такті **T4** код команди, який надійшов в регістр команд, дешифрується - визначається, скільки циклів та тактів треба для виконання команди, яка потім відпрацьовується **МП** під час даного або наступного такту **T5**.

Якщо команда двох- або трьохбайтна, то її виконання здійснюється під час наступних машинних циклів після вибірки останніх байтів команди.

## 2.3 Мікропроцесор 18085A

Однокришталний **МП 18085A** є удосконаленим 8-розрядним **МП**, який має повну архітектурну сумісність з **МП 18080**. В **МП 18085A** на одному кристалі розміщається пристрій, еквівалентний трьом **BIC**: мікропроцесору **18080**, генератору тактових імпульсів **KP580ГФ24** та системному контролеру **KP580BK28**.

В **18085A** використовується такий самий набір програмно-доступних регістрів. Збережена логічна організація пам'яті та простори **BB**. Система команд **18085A** включає весь набір команд **18080** в їх старому кодуванні. Разом з цим до складу системи команд **18085A** введені дві нові, що пов'язано з розширенням засобів обробки переривань та послідовним **BB**:

**RIM** - читання маски переривань;

**SIM** - встановлення маски переривань.

Методологія побудови системи переривань **18085A** підпорядкована архітектурі **18080**, але кількість можливих джерел переривань на апаратному рівні збільшено з одного до п'яти. Поряд із типовим векторним запитом **INTR** (**INT** для **18080**) введені ще чотири, які мають фіксовані вектори переривань: **TRAP**, **RST 7.5**, **RST 6.5**, **RST 5.5**.

Запити на переривання щільно упорядковані. Вищим пріоритетом володіє немаскуємий запит **TRAP**, нижчий наданий векторному перериванню **INTR**.

Ще одна важлива відзнака архітектури **18085A** від **18080** складається в підтримці послідовної лінії **BB**. В **18085A** є можливість програмного доступу до ізольованого однобітового простору **BB**, який реалізується за допомогою команд **RIM** та **SIM**. При виконанні даних команд разом з читанням або установкою регістра масок переривань виконується ввід даних з лінії в старший розряд акумулятора або їх вивід у зворотному напрямку:

**RIM; A7 <--- SID**

**SIM; SOD <--- A7,**

де **A7** - старший розряд акумулятора;

**SID** та **SOD** - однорозрядні ізольовані порти вводу та виводу, відповідно.

На відміну від **І8080 МП І8085А** уявляє собою закінчений однокристальний центральний процесор. Для його запуску необхідно тільки кварцовий резонатор та схема скидання на вході **RESIN**.

Тактова частота **І8085А** дорівнює **3 МГц**. Маються мікропроцесори з більш високою тактовою частотою, наприклад, **І8085-2  $f = 5$  МГц**.

Застосування **МП І8085А** порівнянно з **І8080** вигідніше за рахунок збільшення кількості входних ліній для одержання запитів на переривання, використання каналу послідовного **ВВ**, переходу до єдиного джерела живлення **+5 В**, а також покращання швидкісних якостей системного каналу центрального процесора.

## 3 ЗАГАЛЬНІ ПИТАННЯ ПОБУДОВИ МІКРОПРОЦЕСОРНИХ СИСТЕМ

### 3.1 Система пам'яті МПС

#### 3.1.1 Класифікація запам'ятовуючих пристроїв

За функціональним призначенням всі **ЗП**, які використовуються в **МПС**, підрозділяються на такі групи:

- **понадоперативні ЗП**, які уявляють собою набір регістрів МП, зміст яких безпосередньо використовується при обробці інформації в **МП**;
- **оперативні ЗП**, які зберігають оперативну інформацію (операнди, частини програми), потрібну в процесі роботи;
- **постійні ЗП**, призначені для тривалого зберігання незмінної в процесі роботи **мікро-ЕОМ** інформації (програми, константи);
- **зовнішні ЗП**, призначені для зберігання великих обсягів інформації з невеликою питомою вартістю біта зберігаємої інформації;
- **буферні ЗП** призначені для узгодження різноманітних рівнів системи пам'яті **мікро-ЕОМ** між собою та зовнішніх пристроїв з системою пам'яті.

**ОЗП** і **ПЗП** утворюють **основну пам'ять МПС**, яка реалізується на основі напівпровідникових запам'ятовуючих елементів, або чарунок, які зберігають двійкову інформацію.

Основна пам'ять відноситься до класу **ЗП** з довільною вибіркою (**ЗПДВ**). На відміну від пам'яті з послідовним доступом у **ЗПДВ** до інформації можна звертатися в довільному порядку, і час звернення до чарунки пам'яті не залежить від її адреси. В послідовній пам'яті дані можливо зчитувати тільки у тому порядку, в якому вони записувалися.

### 3.1.2 Основні характеристики запам'ятовуючих пристроїв

Запам'ятовуючі пристрої характеризуються сукупністю якісних показників:

- 1 **Ємність ЗП** визначається максимально можливою кількістю бітів зберігаємої інформації.
- 2 **Питома ємність** - кількістю чарунок пам'яті (або бітів) на кристал. Важливою характеристикою **ЗП** є інформаційна організація кристала пам'яті  $M \times N$ , де **M** - кількість *слів*, **N** - *розрядність слова*. Наприклад, кристал ємністю **16Кбіт** може мати різноманітну організацію: **16Кx1, 4Кx4, 2Кx8**.
- 3 **Ширина вибірки** пов'язана з інформаційною організацією та визначається кількістю інформації, яка записується в **ЗП** або зчитується з нього за одне звернення. При однаковому часі звернення **ЗП** з більшою шириною вибірки має більшу інформаційну ефективність.
- 4 **Швидкодія ЗП** характеризується двома параметрами: **часом вибірки  $T_{чв}$** , який представляє собою інтервал часу між моментом подачі сигналу вибірки та появою зчитаних даних на виході, та **циклом запису  $T_{цз}$** , який визначається мінімально допустимим часом між моментом подачі сигналу вибірки при запису та моментом, коли припустиме наступне звернення до пам'яті.
- 5 **Швидкість** обміну інформацією між **ЗП** та іншими пристроями, яка визначається кількістю біт (байт), які передаються в одиницю часу.
- 6 **Питома вартість**, яка визначається співвідношенням вартості **ЗП** до інформаційної ємності, тобто. вартість біта зберігаємої інформації.
- 7 **Надійність** та споживаєма потужність.
- 8 **Здібність зберігання інформації** при відключенні джерела живлення. Розрізняють **енергонезалежну пам'ять**, в якій при збоях у роботі системи живлення інформація не руйнується, та **енергозалежну**, в якій руйнується.

### 3.1.3 Оперативні запам'ятовуючі пристрої (ОЗП)

**ОЗП** призначені для зберігання оперативної інформації: виконуваних програм, проміжних, вихідних і остаточних результатів.

За принципом зберігання інформації напівпровідникові **ОЗП** підрозділяються на два основні класи:

- *статичні (SRAM-Static Random-Access Memory);*
- *динамічні (DRAM-Dynamic RAM).*

В *статичних ОЗП* кожний біт пам'яті уявляє собою тригер, стан якого визначає стан відповідного біта пам'яті.

В *динамічних ОЗП* стан біта пам'яті запам'ятовується у вигляді наявності або відсутності заряду на входній ємності **MOS-транзистора**.

Через наявність струмів відтоку заряд конденсатора необхідно з визначеною періодичністю регенувати (оновлювати). Під час регенерації запис нової інформації повинен бути заборонений.

Для регенерації пам'яті потрібен спеціальний блок, який автоматично виконує періодичний перезапис змісту усіх чарунок пам'яті (приблизно кожну мілісекунду).

Динамічна пам'ять у зрівнянні зі статичною має більшу питому ємність, меншу вартість, але більше енергоспоживання та меншу швидкодію.

Вважаючи характеристики типів **ОЗП**, у якості **ОЗП ПЕОМ** у дійсний час використовується **DRAM**. Статична пам'ять використовується в основному для аобудови кеш-пам'яті, яка усовує різницю по швидкодії між різноманітними пристроями **ПЕОМ**.

### 3.1.4 Постійні запам'ятовуючі пристрої (ПЗП)

**ПЗП** призначені для зберігання програм, спеціального математичного забезпечення, констант та іншої незмінної інформації. **ПЗП** є енергонезалежними.

**ПЗП** підрозділяються на такі типи:

- програмуємі на заводі-виробнику, або маскові;
- програмуємі користувачем
- репрограмуємі.

Перші два типи допускають тільки однократне програмування, третій тип **ПЗП** дозволяє змінювати інформацію, яка в ньому зберігається, багаторазово.

**Маскові ПЗП** програмуються їх виробником, який за підготовленою користувачем інформацією робить фотошаблони, за допомогою яких заносить цю інформацію в процесі виготовлення на кристал **ПЗП**. **Маскові ПЗП** характеризуються високою надійністю, але через велику вартість їх використання доцільне тільки при великих серіях випуску, тому що вартість ВІС визначається вартістю дуже коштовного технологічного обладнання.

Програмуємі користувачем **ПЗП (ППЗП)** є більш універсальними. Вони являють собою матрицю, в якій у якості елементів пам'яті використовуються чарунки, обладнані легкоплавкими перемичками. Програмування здійснюється перепалюванням перемичок в потрібних розрядах, після чого матриця стає **ПЗП**.

Репрограмуємі **ПЗП (РПЗП)** - це **ПЗП**, які допускають багаторазовий перезапис інформації. Запис інформації в подібні **ПЗП** здійснюється користувачем за допомогою спеціальних пристроїв **програмакторів**, а стирання - електрично або за допомогою опромінювання ультрафіолетовим світлом.

Останній тип **РПЗП** є більш поширеним в наслідок достатньо малого часу вибірки та енергоспоживання, великої ємності.

### **3.1.5 Нарощування пам'яті мікро-ЕОМ**

Сукупність чарунок пам'яті, до яких може звертатися **мікро-ЕОМ**, називається адресним простором або полем пам'яті, та визначається розрядністю шини адреси:

$$(V=2^m, \text{ де } m - \text{розрядність шини адреси}).$$

Якщо ємності однієї **BIC** пам'яті недостатньо для організації всієї пам'яті **мікро-ЕОМ**, то необхідна ємність може набиратися з декількох **BIC**.

Як вже відмічалось, **BIC** пам'яті можуть мати при однаковій ємності різну організацію **MxN**. При цьому потрібний обсяг пам'яті **мікро-ЕОМ** та розрядність слів досягаються нарощуванням її як по горизонталі, так і по вертикалі.

**Горизонтальне нарощування** дозволяє отримувати потрібну розрядність пам'яті при фіксованій кількості слів. Наприклад, пам'ять обсягом **1Kx8** можливо отримати з **8 BIC**: з організацією **1Kx1**.

**Вертикальне нарощування** використовується для отримання потрібного обсягу пам'яті при фіксованій розрядності слів. Наприклад, пам'ять обсягом **1Kx8** можливо отримати з **4 BIC** з організацією **256x8**.

У цілях отримання потрібної організації пам'яті зазвичай використовуються одночасно обидва способи нарощування. Для **мікро-ЕОМ** найбільш зручні **BIC 3П**, які мають розрядність, порівняну з розрядністю шини даних.

### 3.1.6 Підключення ПЗП

Роздивимося організацію **ПЗП** ємністю **4К** байт, який складається з **BIC** ємністю **1К** байт. Число ліній шини адреса, необхідних для підключення пам'яті заданого обсягу, визначається з виразу

$$Na = \text{Log}_2 V,$$

де **V** - ємність ПЗП,

**Na** - число розрядів ША.

В нашому випадку **Na = 12**.

Адреса будь якої чарунки буде складатися з номера **BIC** і номера чарунки **BIC**:

|              |                       |
|--------------|-----------------------|
| 2 розряди    | 10 розрядів           |
| <b>N BIC</b> | <b>Адреса чарунки</b> |

При цьому адресні входи кожної **BIC** пам'яті підключаються до молодших розрядів **ША**, а старші розряди цієї шини використовуються для вибірки (селекції) необхідної **BIC**: (рис.3.1).

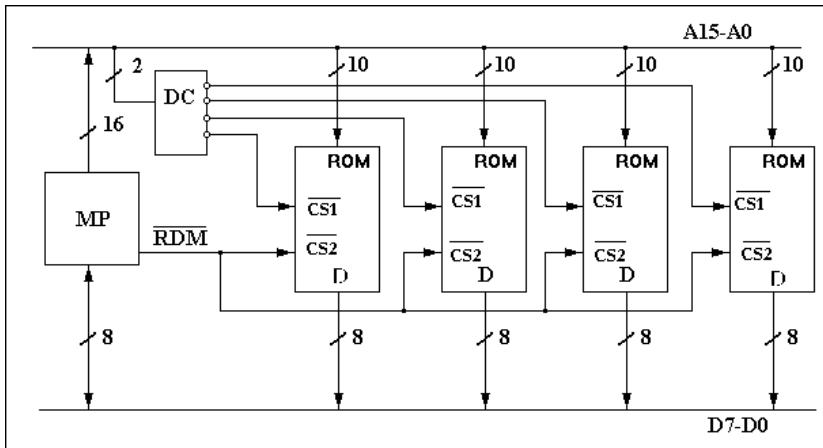


Рисунок 3.1 - Схема підключення ПЗП

У функціональному відношенні схема селекції **BIC** пам'яті представляє собою дешифратор, на входи якого подаються сигнали з ліній **ША**, а на виході утворюються сигнали вибору.

Одноіменні лінії даних всіх кристалів системи пам'яті об'єднуються паралельно та підключаються до системної **ШД**. Але при зверненні до пам'яті з **ШД** електрично пов'язаний тільки один обраний кристал, інші відключені від **ШД**.

По сигналу на вході вибору кристала **CS1** дозволяється робота конкретної **BIC**, по сигналу на вході **CS2** видається зміст визначеної чарунки пам'яті з вибраної **BIC**.

### 3.1.7 Розширення звернень до пам'яті

Зростання швидкодії процесорів за рахунок використання нових технологічних та архітектурних рішень значно випереджає зростання швидкодії оперативної пам'яті. Без узгодження пропускних здібностей процесора та пам'яті неможливо в комп'ютері реалізувати продуктивність, відповідну продуктивності процесора. З цією метою використовуються спеціальні структурні рішення:

- **конвеєризація процедур** циклу виконання команди. В простішому випадку - це виконання паралельно в часі операцій в *АЛП* з вибіркою з пам'яті наступної команди;
- **розширення ОП** шляхом багатомодульної побудови з "віяльною" ("чергуючою") адресацією, при якій суміжні адреси інформаційних одиниць належать різним блокам;
- **буферизація** - використання вбудованої між процесором та *ОП* суттєво більш швидкодіючої, ніж *ОП*, буферної пам'яті порівняно невеликої ємності.

В конвеєрних процесорах адреса чергового звернення до пам'яті генерується одночасно з вибіркою попереднього слова даних. При цьому черговий цикл звернення до пам'яті повинен починатися до закінчення попереднього.

Це досягається шляхом двоспрямованого розширення пам'яті, яке заключається у використанні двох плат пам'яті, звернення до однієї з яких відбувається за парними адресами, а до іншої – за непарними.

Доти на одній з плат завершується виконання попереднього циклу, на другій може початися черговий цикл звернення. При цьому збільшується пропускна здібність *ОП* за рахунок перекриття в часі звернень до різних модулів пам'яті. В системах із багатоспрямованим розширенням контролер пам'яті забезпечує розподіл послідовно виробляємих адрес між кількома платами пам'яті.

### 3.1.8 Кеш-пам'ять

При організації швидкодіючої буферної пам'яті часто використовують так звані *асоціативні ЗП*, які передбачають пошук інформації за деякими визначеними ознаками, які називаються *ознаками запиту*. Простий пошук передбачає знаходження слова, асоціативна ознака якого співпадає з ознакою запиту. При складному пошуку виконується, наприклад, знаходження всіх чисел, порівняних або менш заданого тощо.

**Кеш-пам'ять** є понадоперативною пам'яттю, яка має значно менший цикл звернення.

Основна пам'ять  $M$  (рис.3.2) поділена на  $N$  сторінок (звичай обсягом 1024 байт). Кеш містить  $n$  сторінок такого ж обсягу. При цьому  $N \gg n$ . Будь яка сторінка кешу може заповнюватися інформацією з  $M$  всього за кілька тактів. В кожній сторінці кешу розташовані данні зі суміжними адресами, різні ж сторінки кешу можуть не стикуватися за адресами.

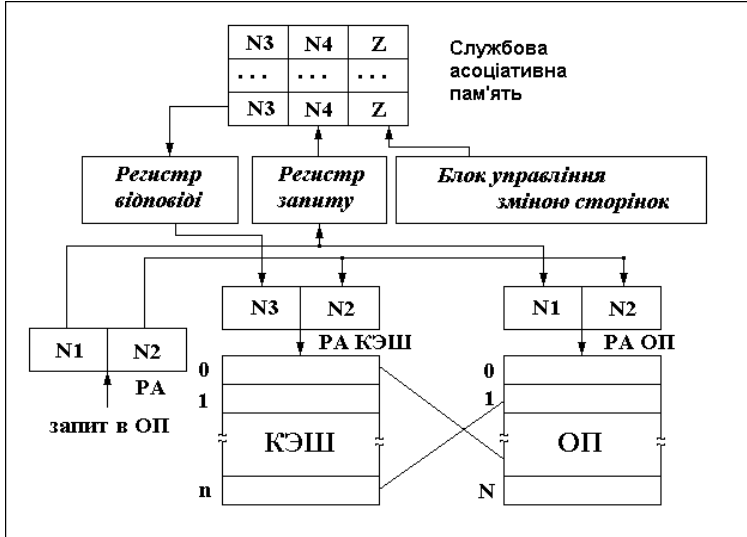


Рисунок 3.2 - Структура кеш-пам'яті

При зверненні до ***M*** з пристрою вибірки команд або пристрою вибірки операндів за допомогою асоціативної пам'яті виконується перевірка наявності потрібної інформації в кеші. Якщо інформація мається, робиться звернення до кешу.

Якщо потрібна сторінка в кеші відсутня, то з ***M*** до кешу робиться запис нової сторінки.

На рисунку 3.2 використовуються такі умовні позначення:

**N1** і **N2** - номери сторінки та слова, вказані в запиті до ОП;

**N3** - номер сторінки кешу;

**N4** - номер сторінки ОП, розміщеної в кеші.

При всякому зверненні до ***M*** за допомогою асоціативної пам'яті робиться перевірка наявності даної сторінки у кеші. Якщо вона є, то адреса в кеші буде:

$$A = N3 \text{ (із асоціативної пам'яті)} + N2 \text{ (із команди звернення)}.$$

До поля ***Z*** періодично заноситься деяке число. При кожному зверненні до сторінки кешу із цього числа відраховується одиниця. Коли в кеші не виявляється необхідної інформації, потребується замінити сторінку. Для цього проглядаються поля ***Z***. Там, де кількість найбільша, сторінка вважається рідко використаною. Замість неї з ***M*** вводиться нова сторінка, й асоціативна пам'ять відповідно коректується.

Ефективність кешу залежить також від характеру розподілу даних в оперативній пам'яті. Нераціональний розподіл даних призводить до частої зміни сторінок.

Виділяються типи кеш-пам'яті:

- із запам'ятовуванням нової інформації одночасно в кеші та в **ОП** ("*крізне запам'ятовування*" - ***store-through***), при цьому в **ОП** завжди є остання копія зберігаємої в кеші інформації. Але в цьому випадку тривалий цикл **ОП** знижує продуктивність процесора;
- із запам'ятовуванням нової інформації тільки в кеші та копіювання її в **ОП** тільки при передачі в інші пристрої або при витисненні з кешу (***store-in-cashe***).

### ***3.1.9 Фізична та логічна структура диска***

Під форматом носія інформації розуміється структура інформації на носії та способи адресації елементів цієї структури. Розрізняють фізичний та логічний формати дисків.

#### **Фізичний формат диска**

Інформація розташовується на концентричних **доріжках** поверхні диска в секторах фіксованої довжини. Одноіменні доріжки декількох поверхней утворюють **циліндр**. Кількість робочих поверхней та циліндрів є характеристикою дисководу. Кількість секторів на доріжці задається програмно (драйвером пристрою).

Кожний сектор складається з поля даних та поля службової інформації. **DOS** встановлює розмір сектора (ємність поля даних) - **512 байт**.

Кількість циліндрів, секторів на доріжці та розмір сектора встановлюється при форматуванні (розмітці) диска.

Фізична адреса сектора визначається тріадою **[t-h-s]**,

де **t** - номер циліндра (доріжки);

**h** - номер поверхні диска (магнітної головки);

**s** - номер сектора на доріжці;

Наприклад, тріада [1-0-2] адресує сектор 2 доріжки (циліндра) 1 нульової поверхні.

Для жорсткого диска нумерація секторів на доріжці не відповідає послідовності їх фізичного розташування. Висока швидкість обертання диска не дозволяє передати до комп'ютеру зміст зчитаного сектора до підводу до головок фізично наступного сектора.

Ось чому сектор з номером, на одиницю більшим попереднього, розташовується за ним через певну кількість секторів (коефіцієнт чергування **n**). При **n=3** нумерується кожний третій сектор, два пропускаються. При цьому уся доріжка зчитується за **n** оборотів замість **S** (кількості секторів).

Така нумерація встановлюється за допомогою процедури, яка має назву **форматування нижнього рівня**.

Обмін інформацією між **ОЗП** та дисками здійснюється тільки секторами.

### Логічний формат диска

Кожна дискета розглядається як єдиний логічний диск. Жорсткий диск розбивається на декілька **розділів**, які використовуються різними **ОС** (максимальна кількість розділів - 4):

**I - первинний розділ DOS** (системний логічний диск C);

**II - розширений розділ DOS** (логічні диски);

**III, IV - розділи не-DOS** (наприклад, **Unix, Xenix** і т.д.). Ці розділи **DOS** не доступні.

Логічний дисковий простір будь якого логічного диска поділяється на дві області: **системну** та **даних**.

**Системна область** створюється та ініціалізується при форматуванні, а в подальшому оновлюється при маніпулюванні файловою структурою.

Область даних містить файли та каталоги, які підпорядковані кореневому. На відміну від системної доступна через інтерфейс користувача. Системна область логічного диску складається з наступних компонент:

- завантажувального запису ( **BR - Boot Record**);
- таблиці розміщення файлів ( **FAT - File Allocation Table** );
- кореневого каталогу ( **RDir - Root Directory**).

**Boot Record** містить:

- блок параметрів диску ( **DPB - Disk Parameter Block** );
- системний завантажувач ( **SB - System Bootstrap** ).

**SB** завантажує систему, якщо диск є системним.

**DPB** містить інформацію о кількості та розмірі секторів, розмірі кластера, розмірі **FAT** тощо, тобто служить для ідентифікації фізичних та логічних форматів логічного диску.

Сектори диску об'єднуються в більш крупні структури - **кластери**.

**Кластер** - це група суміжних секторів, яка є мінімальною одиницею дискової пам'яті, котра виділяється файлу (або некореневому каталогу).

При занесенні файла на диск він розбивається на групу кластерів, які можуть розташовуватися, де завгодно на диску. Послідовність при організації файла не потребується.

Кожний з файлів займає цілу кількість кластерів. При цьому останній кластер може бути задіяний не повністю. На дискетах кластери займають один або два сектори, на жорстких дисках - звичайно 4 або 8.

**File Allocation Table** - представляє собою карту (образ) області даних, в якій описується стан кожного кластера та зв'язуються у ланцюжок кластери, які належать одному файлу (некореновому каталогу).

**FAT** показує, на яких кластерах яких доріжок зберігаються файли. Не має значення, як розкидані окремі кластера, виділені під файл - програма бачить тільки весь файл цілком.

Логічне розбиття області даних на кластери замість використання одиничних секторів має наступний сенс:

- зменшується можлива фрагментація файлів;
- зменшується розмір **FAT**, а значить, і обсяг системної області логічного диска;
- прискорюється доступ до файлу, тобто в декілька разів скорочується довжина ланцюжків фрагментів дискового простору.

Але дуже великий розмір кластера веде до неефективного використання області даних, особливо у випадку великої кількості маленьких файлів.

З метою безпеки **FAT** зберігається у двох ідентичних примірниках. Оновлюються копії одночасно. Використовується тільки перший примірник. У випадку його пошкодження звернення йде до іншого примірника.

**Root Directory** - кореневий каталог - містить інформацію про файл, в том числі номер першого кластера кожного файла. **RD** розташовується на диску зразу після **FAT**. Розмір каталога залежить від формату диска.

Підкаталоги розташовуються у просторі даних диска так само, як і звичайні файли. Розмір підкаталогу, як і звичайного файла не обмежений.

**Master Boot Record (MBR)**- головний завантажний запис (на жорсткому диску). Має адресу [0-0-1].

Містить внесистемний завантажник (***NSB - NonSystem Bootstrap***) та таблицю розділів ( ***Partition Table*** ). Таким чином, в стартовому секторі фізичного жорсткого диска знаходиться не ***BR***, а ***MBR***.

***NSB*** здійснює завантажування в ***ОЗП SB (System Bootstrap)*** з ***BR*** логічного диска в активному розділі та передачу йому керування.

***PT*** описує розташування та характеристики розділів, які є на вінчестері.

### 3.2 Режими обміну інформацією

Між МП та периферійними пристроями (***ПП***) відбувається обмін інформацією двох типів: службовою та власно даними. Службова інформація від МП ініціює дії, пов'язані з обміном даних, та передається за допомогою управляючих слів ***CW (Control Word)***.

Службові повідомлення від ***ПП***, які інформують систему про поточний стан ***ПП***, мають назву ***слів стану SW (Status Word)***.

Дані передаються за допомогою ***слів даних DW (Data Word)***.

Обсяг службової інформації, якою обмінюються ***ПП*** та ***МПС***, а також її інтерпретація залежать від типу ***ПП***. Для найбільш простих пристроїв (наприклад, світлодіодних матриць) службова інформація не потрібна. Для інших пристроїв, наприклад НГМД, керуюча інформація та дані про стан ***ПП*** можуть мати значний обсяг.

#### 3.2.1 Протоколи обміну. Арбітраж

Під протоколом обміну розуміється сукупність правил, які встановлюють єдині принципи взаємодії підсистем.

Основними протоколами обміну є:

- ***синхронний*** (тактуємий) обмін, при якому передача інформації по магістралям ***МПС*** здійснюється за один суворо заданий по тривалості проміжок часу (такт);

- *асинхронний* - нетактуємий обмін;
- *напівсинхронний*, при якому сам процес обміну тактується в часі, але займає декілька тактів.

**Задавач** - пристрій, який керує роботою магістралей при роботі з іншим пристроєм - виконавцем.

**Виконавець** - пристрій, який керується задавачем при обміні інформацією.

### **Синхронний обмін**

Обмін інформацією між задавачем та виконавцем відбувається на інтервалі дії імпульсу синхронізації.

Перевага синхронного обміну міститься в простоті його реалізації та максимальній швидкодії. Але у випадку наявності в МПС виконавців з різною швидкістю, мінімальну тактову частоту синхроімпульсів необхідно знижувати, сходячи з надійного забезпечення обміну інформацією з самим повільним виконавцем.

### **Асинхронний обмін**

При наявності в МПС виконавців з різною швидкістю оптимальним є режим, при якому обмін з кожним виконавцем здійснюється з тією швидкістю, на яку він розрахований. В цьому випадку виконавці отримують магістралі МПС на необхідний їм час.

### **Арбітраж**

**Арбітр** - схема, яка керує у відповідності з деяким пріоритетом черговістю захоплення задавачами управління магістралями.

**Арбітраж** - процедура розгляду запитів задавачів на управління магістралями.

Основна функція арбітражу міститься у розподілі часу користування магістралями МПС між задавачами з метою усунення конфлікту, пов'язаного з участю у обміні декількох задатчиків та виконавців. Якщо в системі є один задавач, необхідність в арбітражі відпадає.

### 3.2.2 Програмний обмін інформацією

Даний режим характеризується тим, що всі дії по обміну даними реалізуються командами прикладної програми. Найбільш простими ці дії виявляються для "завжди готових" периферійних пристроїв, наприклад, індикаторів на світлодіодах.

При необхідності обміну у відповідному місті програми використовуються команди **IN** або **OUT**. Така передача даних називається **синхронним**, прямим або безумовним обміном (рисунок 3.3,а).

Для більшості периферійних пристроїв до виконання операції необхідно переконатися в їх готовності до цих операцій, тобто обмін стає **асинхронним**.

Загальний стан периферійних пристроїв характеризуються прапорцем готовності **READY**, який входить до слова стану цього пристрою. МП перевіряє прапорець готовності, та якщо він встановлений, ініціює власно ввід або вивід даних.

Існує два типи умовного обміну: з заняттям циклу (рисунок 3.3,б) та сполучений (рисунок 3.3,в). У першому випадку МП зависає на циклі чекання готовності, витрачаючи на це весь машинний час. В другому випадку, якщо ПП не готовий до обміну, МП повертається до основної задачі без виконання операції ВВ.

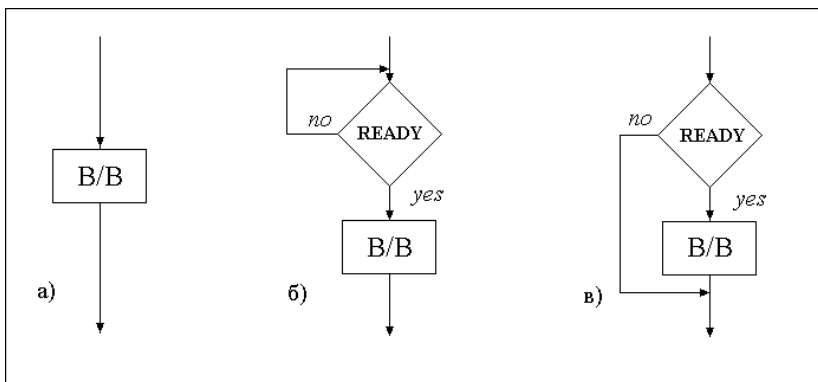


Рисунок 3.3 - Програмний ввід/вивід

Після завершення операції обміну сигнал готовності **READY** повинен бути знятий та виставлений знову тільки при новій готовності до обміну. З цією метою периферійний пристрій слід проінформувати про закінчення операції, для чого використовується включений до одного з управляючих слів **CW** сигнал підтвердження **ACK(Acknowledgment)**.

Протокол обміну службовою інформацією такого типу має назву **квитування**. Він забезпечує надійну асинхронну передачу даних зі швидкостями, які визначає ПП.

Основний недолік програмного обміну пов'язаний з непродуктивними витратами часу процесора в циклах чекання, в яких він не виконує ніякої корисної праці.

Перевагою програмного ВВ є простота його реалізації, яка не потребує додаткових апаратних витрат.

### 3.2.3 Обмін інформацією в режимі перервань

Більш ефективним є обмін даними з перериванням програми. Цей вид обміну ініціюється зовнішнім пристроєм (ЗП), який надсилає до МП спеціальний сигнал **INT** - "Запит переривання".

Цей сигнал з'являється в довільні моменти часу, асинхронно до дій МП, тому керувати його появою програма не може. Тому, реагуючи на сигнал **INT**, МП після виконання поточної команди повинен припинити виконання основної програми і виставити сигнал **INTE** ("Дозвіл переривання"). При цьому зміст програмного лічильника, **PSW**, а також усіх або частини **P3H** записуються до **P3П**. Після цього виконується підпрограма обміну (цей режим називають також апаратним викликом підпрограм).

У випадку одночасної подачі запитів на переривання від кількох зовнішніх пристроїв вступає в силу система пріоритетів, яка дозволяє встановити черговість їх обслуговування. Найбільш часто це вирішується двома способами: послідовним опитом(система **полінга**) або перериванням за вектором (**система пріоритетних переривань**).

У першому методі за сигналом запиту на переривання МП виробляє сигнал дозволу переривання, який надходить через системний контролер послідовно до всіх зовнішніх пристроїв. Цей сигнал відсилається спочатку до пристрою з найвищим пріоритетом. Якщо цей пристрій не запитував переривання, то сигнал передається наступному пристрою (з більш низьким пріоритетом) і т.і. Якщо ж пристрій запитував переривання, то він перериває подальше поширення сигналу. Подібний ланцюжок отримав назву "*полінга*".

При використанні другого способу перехід на обслуговування переривання йде безпосередньо по запиту конкретного зовнішнього пристрою, який передає МП також інформацію про свою адресу (адресу відповідної підпрограми), в наслідок чого відпадає необхідність опиту інших ЗП.

Система пріоритетних переривань дозволяє ідентифікувати перериваючий пристрій з допомогою сукупності апаратних засобів, які при одночасному надходженні кількох запитів надають обслуговування тільки одному пристрою, який має старший пріоритет. Завжди самий вищий пріоритет присвоюється найменш швидкодіючому пристрою.

У випадку, якщо деяка програма не може бути перервана з будь-яких міркувань, використовується особливий стан системи, який має назву *маскування*. Запити на переривання, ініційовані замаскованим пристроєм, не приймаються системою. Замаскувати можливо всі без виключення запити на переривання або тільки запити, які мають пріоритет не вище завданого. Зокрема при виконанні програми, початої на основі запиту на переривання з деяким пріоритетом, усі запити на переривання, які мають більш низький пріоритет, маскуються автоматично.

### ***3.2.4 Прямий доступ до пам'яті***

У разі необхідності передачі великих масивів інформації доцільно робити обмін даними між зовнішніми ЗП та ОЗП безпосередньо, без використання робочих регістрів МП, зокрема акумулятора.

Такий режим роботи отримав назву **прямого доступу до пам'яті (ПДП)**. ПДП реалізується за допомогою спеціальних апаратних засобів, які мають назву контролера ПДП.

Для прямого доступу до пам'яті управління шинами МПС передається ЗП, який і виконує захоплення шин, призупиняючи при цьому виконання програми МП. Після отримання сигналу "Захоплення шин" МП, закінчує виконання поточного машинного циклу, припиняє виконання програми, переводить шини адреси і даних у високоомний стан і встановлює сигнал логічної одиниці на лінії "Підтвердження захоплення".

Отримавши цей сигнал, ЗП здійснює обмін з пам'яттю. По завершенні обміну ЗП встановлює на лінії "Захоплення шин" логічний нуль, і управління шинами передається МП.

При виконанні ПДП зміст внутрішніх регістрів МП не модифікується, тому його не треба запам'ятовувати в пам'яті; а потім поновлювати, як при обробці переривань. Зазвичай передача даних у режимі ПДП має пріоритет перед іншими видами обміну.

Інтерфейс каналу ПДП по складності перевищує інші типи інтерфейсів. Хоча його структура може бути спрощена за рахунок програмної реалізації частини функцій, це не може бути доцільним, тому що основна ціль використання режиму ПДП - забезпечення вищої швидкості передачі даних.

### 3.3 Інтерфейси мікропроцесорних систем

#### 3.3.1 Призначення інтерфейсів

Під **стандартним інтерфейсом** розуміється сукупність уніфікованих апаратних, програмних і конструктивних засобів, необхідних для реалізації взаємодії окремих функціональних елементів (ФЕ) в автоматичних системах збору та обробки інформації. Термін "інтерфейс" зазвичай трактується як синонім слова "сполучення".

Основним призначенням інтерфейсу є уніфікація внутрисистемних та міжсистемних зв'язків і пристроїв спряження. Основні функції інтерфейсу містяться в забезпеченні інформаційної, електричної та конструктивної сумісності між ФЕ системи.

**Інформаційна сумісність** - це узгодження взаємодій ФЕ системи у відповідності зі сукупністю логічних умов.

Логічні умови визначають: структуру та склад шин; засіб кодування і формати інформації; часові співвідношення між керуючими сигналами. Логічні умови визначають функціональну та структурну організацію інтерфейсу.

**Електрична сумісність** - це узгодження параметрів електричних сигналів, тобто рівні напруги (струму), тривалість імпульсів.

**Конструктивна сумісність** - це узгодження конструктивних елементів інтерфейсу: типи та геометрія роз'ємних з'єднань, кількість і розташування контактів тощо.

### 3.3.2 Класифікація інтерфейсів

Основними класифікаційними ознаками є:

- **спосіб з'єднання** компонентів системи (магістральний, радіальний, ланцюжковий або комбінований);
- **спосіб передачі** інформації (послідовний, паралельний, паралельно-послідовний);
- **принцип обміну інформацією** (синхронний, асинхронний);
- **режим передачі інформації** (однобічна, двобічна одночасна та двобічна почергова передача).

За **функціональним призначенням** інтерфейси підрозділяються на наступні основні класи:

- системні (машинні або вводу-виводу) інтерфейси ЕОМ;
- периферійного обладнання;
- мультимікропроцесорних систем;
- розподілених обчислювальних систем;
- програмнокеруємих пристроїв.

### 3.3.3 Інтерфейс із ізольованою шиною

Характерною особливістю даного типу організації інтерфейсу (рисунок 3.4) є роздільна адресація пам'яті та периферійних пристроїв при обміні інформацією. Це здійснюється шляхом використання окремих груп команд для обміну інформацією з пам'яттю та зовнішніми пристроями.

В МП I8080 для обміну інформацією з ПБВ використовуються команди **IN PORT** та **OUT PORT**. Дані команди дозволяють МП адресувати до 256 портів вводу та 256 портів виводу. Максимальна ємність адресуємої пам'яті при цьому - 64 Кбайт.

Обмін здійснюється по керуючим сигналам вводу та виводу, які формуються системним контролером при виконанні відповідних команд. Передача даних здійснюється між акумулятором та буферним регістром відповідного порту.

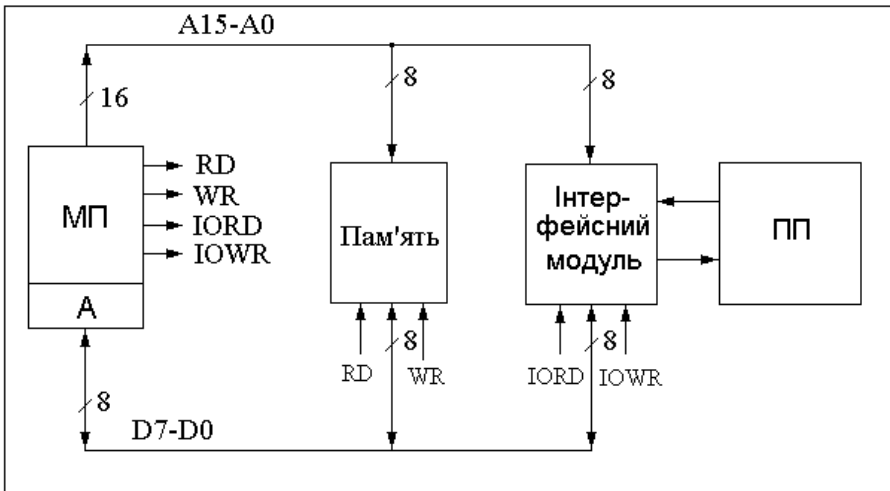


Рисунок 3.4 - Інтерфейс із ізольованою шиною

**Перевага:** відносна простота реалізації.

**Недоліки:**

1 Обмін виконується тільки через акумулятор МП, та для передачі даних до РЗН, якщо *A* зайнятий, потрібне виконання чотирьох команд (додатково до команд читання в *A* і пересилки з *A* в **РЗП** додаються команди запам'ятовування змісту *A* в якому-небудь **РЗП** та наступної відбудови змісту *A*).

2 Кількість підключаємих периферійних пристроїв вводу або виводу не більш 256.

### **3.3.4 Інтерфейс із загальною шиною**

При такій організації частина загального адресного простору відводиться для периферійних пристроїв, які адресуються, як і чарунки пам'яті. Звернення до периферійних пристроїв здійснюється безпосередньо набором команд, які використовуються для передачі даних між МП та пам'яттю. При цьому команди вводу-виводу не використовуються.

Такий принцип розподілу адресного простору реалізується шляхом виділення, наприклад, старшого розряду **A15** шини адреси (рисунок 3.5) в якості ознаки звернення до пам'яті (**0-32K**) або **ПВВ** (**32K-64K**):

0 – звернення до пам'яті

**A15=**

1 – звернення до ПВВ

Під периферійні пристрої можливо виділити будь який адресний простір, якщо використовувати в якості ознаки звернення замість **A15** кількох старших розрядів адреси.

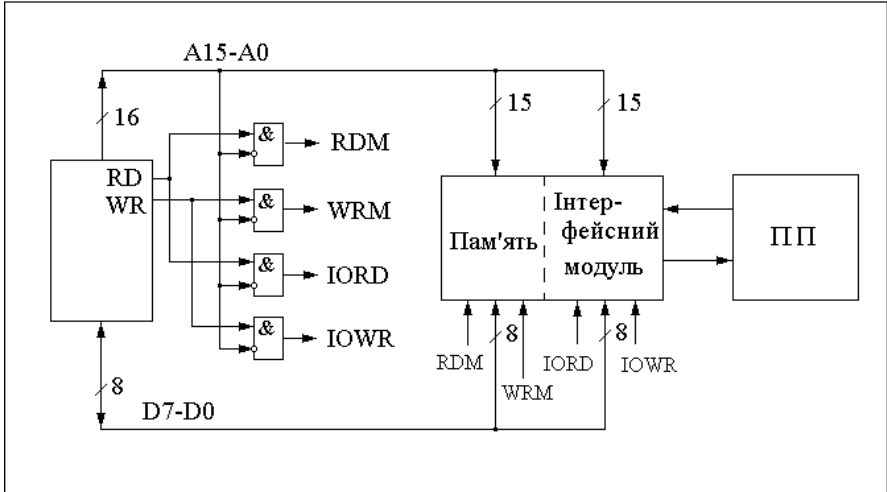


Рисунок 3.5 - Інтерфейс з загальною шиною

**Переваги:**

- Розширення набору команд для звернення до ПБВ, що дозволяє скоротити обсяг програм та підвищити швидкодію. Наприклад: **MOV r,M**, **MOV M,r**; **ADD M**, тощо.
- Значне збільшення ПБВ, які підключаються до МП.
- Можливість обміну інформацією з будь яким регістром МП.

**Недолік:** скорочення області пам'яті та ускладнення дешифруючих схем.

Розподіл адресного простору показано на рисунку 3.6.

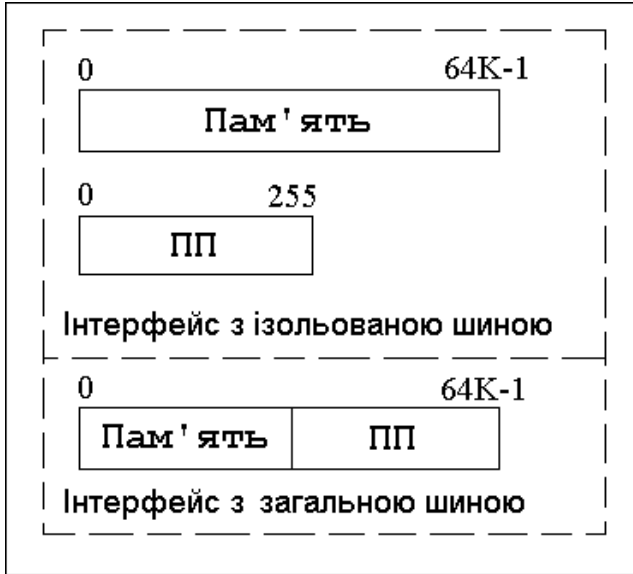


Рисунок 3.6 - Розподіл адресного простору

### 3.3.5 Апаратний склад інтерфейсу

#### Шинні формувачі

Як відомо, навантажна здібність виходів МП обмежена. Тому щоб уникнути струмового перевантаження мікропроцесора проводять буферизацію шин, що дозволяє суттєво збільшити їх навантажну здібність. Особливо часто така ситуація виникає, якщо розробляється система має великі розміри та утворює значне ємкісне навантаження із-за кабелів великої довжини.

Для підвищення навантажної здібності виводів МП використовуються спеціальні шинні посилювачі (шинні формувачі), які представляють собою 8-розрядні паралельні приймально-передавачі з трьома стійкими станами, реалізовані, наприклад, у вигляді ІС КР580ВА86(87) (рисунок 3.7).

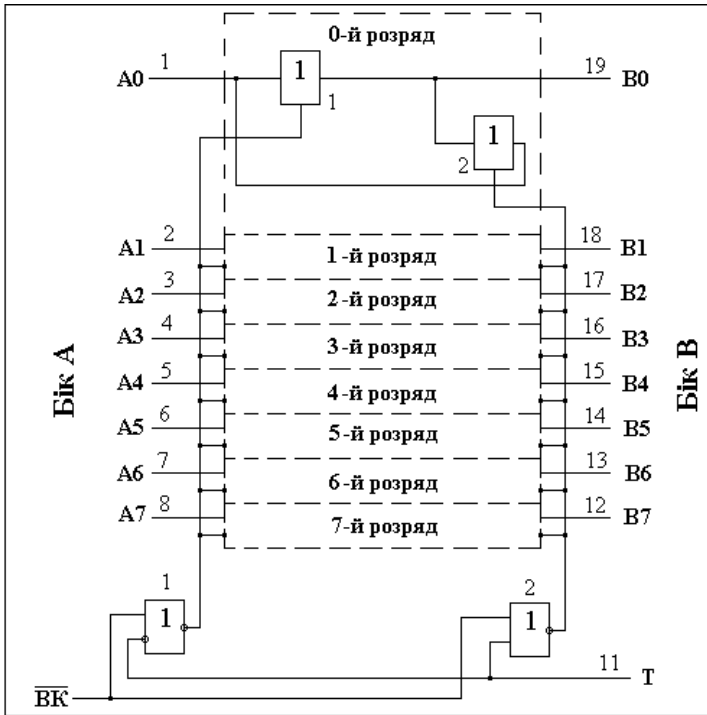


Рисунок 3.7 - Шинний формувач

Кожний приймальнопередавач складається з двох повторювачів, які включені назустріч один одному та мають три стійких стану. При цьому якщо один з повторювачів знаходиться в увімкненому стані, другий буде знаходитися у вимкненому (третьому) стані. Таким чином, передача інформації може здійснюватися тільки в одному напрямку: або від виводів **A** до **B**, або від **B** до **A**.

Вхід **T** - вхід управління напрямком передачі, **БК** - вхід дозволу на передачу, виводи **A** та **B** - інформаційні. До виводів **A** підключаються виходи МП, а до виводів **B** - системні шини адреси або даних.

Якщо на вхід **ВК** подається сигнал високого рівня (наприклад сигнал "**Підтвердження Захоплення**"), то всі інформаційні входи та виходи мікросхеми переходять в високоомний стан, та МП відключається від системної шини. При подачі на цей вхід сигналу низького рівня відбувається передача даних у напрямку, який визначається сигналом на вході **T** (наприклад, "**Запис**"): при **0** - від **B** до **A**, **1** - від **A** до **B**.

### **Порти вводу-виводу**

Портом вводу виводу називається комплекс засобів, призначений для підключення периферійних пристроїв до мікро-ЕОМ. Порти вводу-виводу виконують наступні функції:

- вибір потрібного модуля системи, тобто селекцію необхідного зовнішнього пристрою;
- підключення обраного модуля до шини даних по відповідному сигналу або відключення від неї (роль буфера);
- зберігання рівнів бінарних сигналів під час їх читання або запису незалежно від тривалості цих операцій, тобто роль фіксатора рівнів сигналів.

У найпростішому випадку порт вводу-виводу є комбінацією шинного формувача та дешифратора (рисунок 3.8), у якій дешифратор служить для селекції необхідного зовнішнього пристрою. При цьому на виході дешифратора сигнал з'являється тільки при наявності на його вході суворо визначеного коду, відповідного конкретному ПВВ.

Крім розшифрування коду зовнішнього пристрою дешифратор також повинен розшифровувати сигнал обраного ПВВ ("**Читання**" або "**Запис**").

У відзнаку від пам'яті зовнішні пристрої не мають безпосереднього контакту з системними шинами адреси та даних. Зв'язок **ПВВ** з шинами відбувається через порти вводу-виводу.

Код вибору зовнішнього пристрою міститься в команді вводу-виводу та по ША передається на усі контролери периферійних пристроїв. Кожний з них має свій власний "замок", який називається селектором пристрою. Код вибірки представляє собою своєрідний ключ до конкретного пристрою.

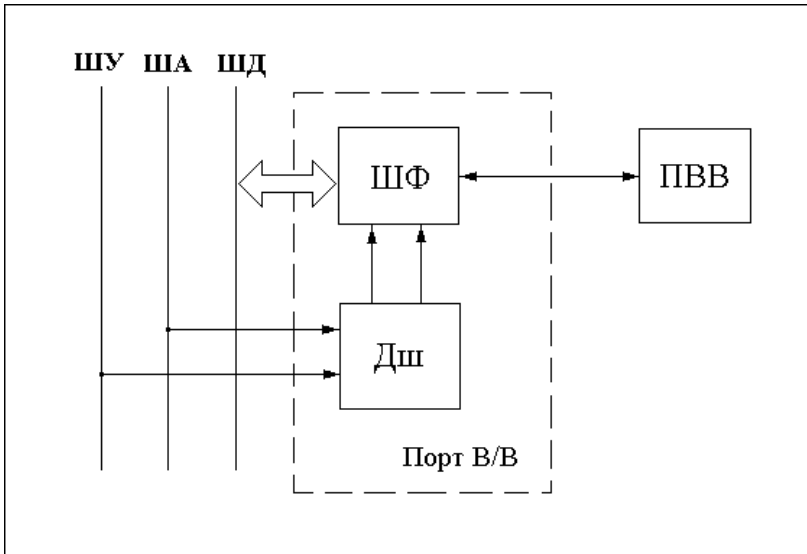


Рисунок 3.8 - Порт вводу-виводу

З контролера вводу-виводу сигнал передається на вхід "Вибір модулю" (**ВМ**) відповідного порту вводу або виводу. При наявності такого сигналу до **МП** у процесі роботи системи може підключатися тільки один периферійний пристрій. Іншими словами, у кожний момент часу у системі є тільки один модуль, який бере участь у обміні даними. Входи та виходи усіх інших периферійних пристроїв при цьому лишаються у високоімпедансному стані, тобто практично відключеними від системних шин.

## 4 ОРГАНІЗАЦІЯ 16-РОЗРЯДНОГО МІКРОПРОЦЕСОРА I8086

### 4.1 Основні характеристики МП

**МП I8086** представляє собою однокришталний 16-розрядний **МП**, виконаний по *n-MOS* технології. **МП** має 20-розрядну мультиплексну шину та розрахований на роботу як в одно-, так і в багатопроцесорних системах. **МП** може працювати з пам'яттю обсягом до 1Мбайт, обмінюватися інформацією з 64Кбайт зовнішніх пристроїв, має до 256 різних видів перервань.

**I8080** на мові асемблера сумісний з **I8086**. Регістри та систему команд **I8080** можна розглядати як підсистему регістрів та команд **I8086**.

За реальною продуктивністю **I8086** в 8-12 разів перевищує **I8080**, що досягається за рахунок підвищення швидкодії схемних елементів та ускладнення центрального процесора, в якому використані нові архітектурні рішення.

### 4.2 Архітектура I8086

#### 4.2.1 Пристрої спряження з каналом

Зазвичай процес виконання команд в МП складається з наступних етапів:

- вибірка коду команди з пам'яті;
- виконання команди;
- запис результату (якщо потрібно).

Як правило, ці етапи виконуються послідовно, що призводить до недовикористання по часовому завантаженню магістралей мікро-ЕОМ.

В **18086** процес виконання команд складається з тих же етапів, але проводиться в двох окремих блоках: **пристрої спряження з каналом (ПСК)** та **операційному пристрої (ОП)** (рисунок 4.1). Блоки можуть працювати незалежно один від одного, отже, процеси переутворення та передачі інформації в них можуть йти паралельно.

**ПСК** формує 20-розрядну адресу пам'яті, виконує вибірку команд та операндів з пам'яті, формує чергу команд, запам'ятовує результат виконання команд у пам'яті. В той час, як **ОП** зайнятий виконанням команди, **ПСК** вибирає наступні команди програми з пам'яті та зберігає їх в реєстрі черги команд ( **РЧК** ). В **РЧК** може бути записано до шести кодів (байт) команд.

З черги коди команд передаються до **ОП**. Після вибірки чергових двох байтів **ПСК** автоматично звертається до пам'яті за наступною командою (конвеєрний принцип обробки команд). За один цикл отримання коду команди в **ПСК** записується з пам'яті 2 байта команди.

В більшості випадків у **ПСК** знаходиться хоча б одна команда, тому **ОП** не чекає, поки чергова команда буде витягнута з пам'яті. Коди команд надходять до **ОП** послідовно так, як вони записані у програмі. Черга (конвеєр) команд функціонує як буфер з дисципліною **FIFO** ("першим прийшов - першим обслуговуваний").

У випадку, якщо **РЧК** заповнений, а **ОП** не оброблює інформацію, то з каналу інформація не поступає, й канал може бути зайнятий іншим мікропроцесором.

Якщо **ОП** виконує команду передачі управління в інше місце програми, то **ПСК** очищує конвеєр команд, отримує код команди з нової адреси (адреси переходу), передає його в **ОП** і починає заповнювати чергу знову. Якщо **ОП** потребує звернення до пам'яті або зовнішнього пристрою, то **ПСК** припиняє процес отримання команд в чергу та організує необхідний цикл обміну даними.

До складу **ПСК** входять:

- шість 8-розрядних реєстрів конвеєра команд;
- чотири 16-розрядних сегментних реєстри **CS, DS, SS, ES**;
- 16-розрядний реєстр адреси (показчик команди) **IP**;
- два буферних 16-розрядних реєстра зв'язку для зберігання даних, які пересилаються, та адрес;
- 20-розрядний суматор адреси.

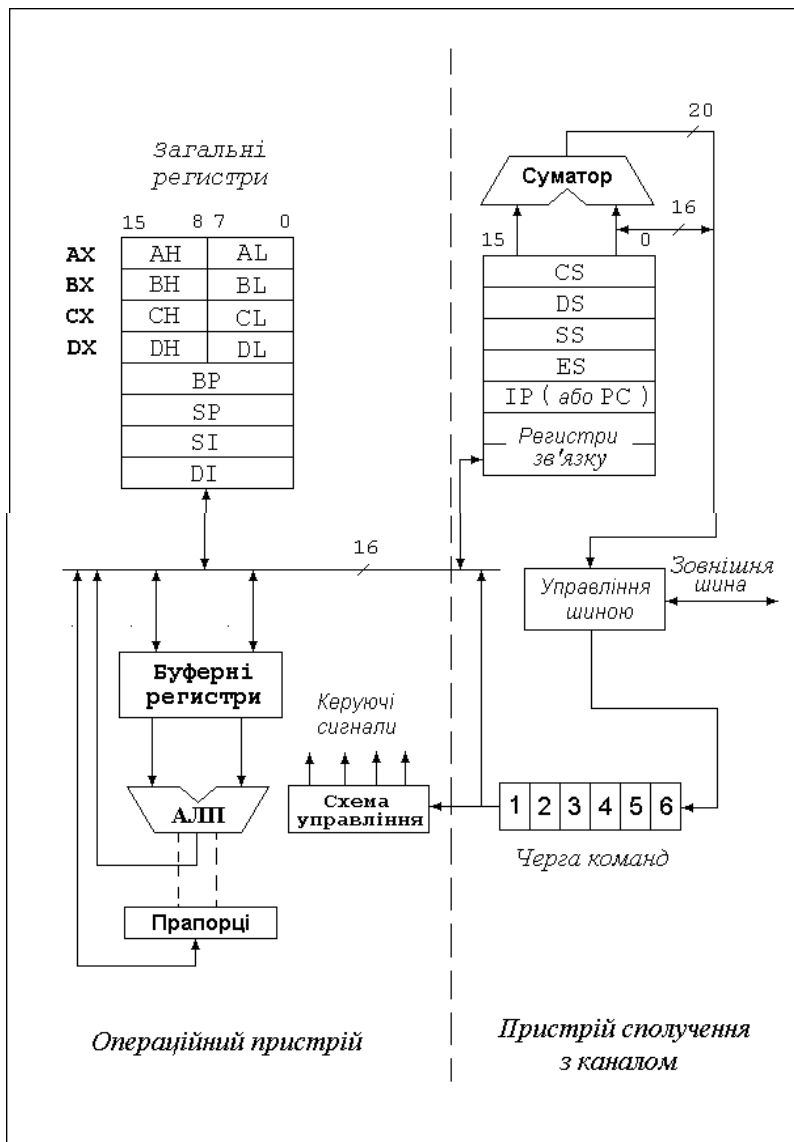


Рисунок 4.1 - Структурна схема I8086

#### 4.2.2 Операційний пристрій

**ОП** призначений для виконання операцій по обробці даних. До складу **ОП** входять:

- 16-розрядний **АЛП**, який виконує арифметичні та логічні операції з 8 та 16-розрядними операндами;
- мікропрограмний пристрій управління;
- реєстри загального призначення.

Команди, вибрані з пам'яті та записані в реєстри черги команд **ПСК**, по запитам від **ОП** надходять через 8-розрядну магістраль команд на мікропрограмний пристрій управління, який декодує команди та виробляє відповідну послідовність мікрокоманд, яка керує процесом виконання поточної операції.

**ОП** не має безпосереднього зв'язку з зовнішньою магістраллю системи та обмінюється даними з **ПСК** через реєстр обміну (зв'язку). Результат обробки фіксується в одному з реєстрів або передається в **ПСК** по двонаправленій 16-розрядній магістралі.

#### 4.2.3 Реєстри МП

**МП** має наступні групи 16-розрядних реєстрів (рисунки 4.2):

- реєстри загального призначення ;
- сегментні реєстри;
- реєстр-показчик команд **IP (Instruction Pointer)** та реєстр ознак **F**.

–

**РЗП** використовуються для зберігання даних і адрес та розділені на дві групи по чотири реєстра. Група реєстрів даних (**H-L**) може адресуватись як вісім 8-розрядних реєстрів **AL-DL** або як чотири 16-розрядні реєстри **AX-DX**. При виконанні деяких команд ці реєстри мають спеціальне призначення: **AX** - акумулятор, **BX** - база, **CX** - лічильник, **DX** - дані.

**Адресні реєстри** - це два реєстра-показчика адреси та два індексних реєстра:

- реєстр-показчик стека **SP** зберігає адресу вершини стеку відносно його початку;
- реєстр **BP** - базові адреси;
- реєстри **SI, DI** - індекси при адресації масивів даних .

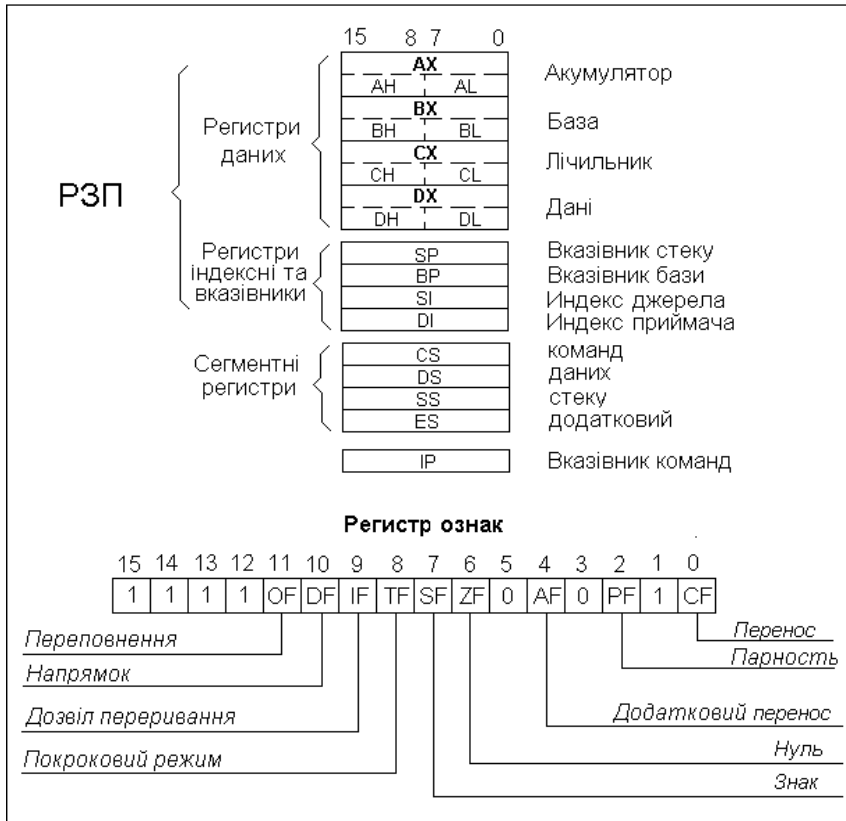


Рисунок 4.2 - Регістри 18086

Сегментні реєстри призначені для зберігання початкових адрес сегментів пам'яті, які використовуються в поточній програмі для зберігання команд або даних. Регістр **CS** вказує сегмент команд, реєстр **SS** - сегмент стека, реєстри **DS, ES** - сегменти даних.

Регістр-показчик команд **IP** виконує функції програмного лічильника. Він вказує зміщення поточної адреси команди в сегменті команд відносно базової адреси сегмента, яка зберігається в реєстрі **CS**.

### 4.3 Адресація пам'яті

Незважаючи на те, що всі адреси, з якими оперують **ОП**, 16-розрядні, **ПСК** робить необхідні перетворення адрес так, щоб **ОП** мав можливість звертатися до всього можливого адресного простору (1 Мбайт) МПС.

#### 4.3.1 Сегментація

Простір пам'яті при цьому умовно розбивається на групу сегментів. Сегмент уявляє собою логічну одиницю пам'яті розміром 64 Кбайт. Він складається із суміжних чарунок пам'яті та уявляє собою незалежну та окремо адресуєму одиницю пам'яті.

Кожному сегменту програмою призначається базова (початкова) адреса, яка є адресою його першого байта в адресному просторі пам'яті. Всі сегменти починаються на 16-байтних межах пам'яті, які мають назву ***меж параграфів***. Сегменти можуть слідувати один за одним безперервно, з інтервалом або перекриватися.

Сегментація дозволяє організувати захист областей пам'яті від непередбаченого доступу. Інформація може зберігатися в неперекривних сегментах пам'яті в залежності від її виду. При цьому можна виключити такі ситуації, як вибірка з пам'яті даних та інтерпретація їх як команд, самознищення інформації за рахунок непередбаченого перекриття областей стека та програм тощо.

Вибір сегментного реєстра при обчислюванні фізичної адреси робиться МП автоматично в залежності від виду інформації, яка витягається з пам'яті.

Сегментні реєстри вказують чотири сегмента, доступні в поточний момент часу. Для отримання доступу до інших сегментів зміст сегментних реєстрів змінюється програмним шляхом.

#### 4.3.2 Фізичні та логічні адреси

Кожна чарунка пам'яті має дві адреси: фізичну та логічну. Фізична адреса уявляє собою 20-бітне значення в діапазоні від **0** і до **FFFFF**, яке однозначно ідентифікує положення кожного байта в просторі пам'яті 1М байт. Саме фізична адреса видається на шину адреси на початку кожного циклу шини, пов'язаного зі зверненням до пам'яті.

Програми оперують не фізичними, а логічними адресами, що дозволяє розробляти їх без апіорного знання того, як конкретна програма розташована в пам'яті. Логічна адреса складається з двох 16-бітних беззнакових значень: **базової** (початкової) адреси сегмента та внутрісегментної адреси або **зміщення**. Для будь якої чарунки пам'яті база ідентифікує перший байт відповідного сегмента, а зміщення означає відстань у байтах від початку сегмента до цієї чарунки.

При формуванні фізичної адреси зміст кожного сегментного регістра розглядається як 16 старших розрядів **A19-A4** початкової адреси відповідного сегмента. Молодші розряди **A3-A0** цієї адреси завжди покладаються рівні нулю і тому не запам'ятовуються в регістрах, а приписуються праворуч до старших розрядів під час операції обчислювання фізичних адрес.

Ця операція виконується суматором адреси, розташованим в **ПСК**, та полягає в складанні 20-розрядної початкової адреси сегмента з 16-розрядним зміщенням (рис. 4.3).

Суматор адреси здійснює, наприклад, наступні обчислення: **CS+IP** - при виборці чергової команди, **SS+SP** - при зверненні до стеку. Операнди звичайно розташовуються в сегменті даних, який адресується сегментним регістром **DS**. Зміщення в сегменті уявляє собою виконавчу адресу, яка обчислюється в **ОП** згідно вказаному в команді способу адресації.

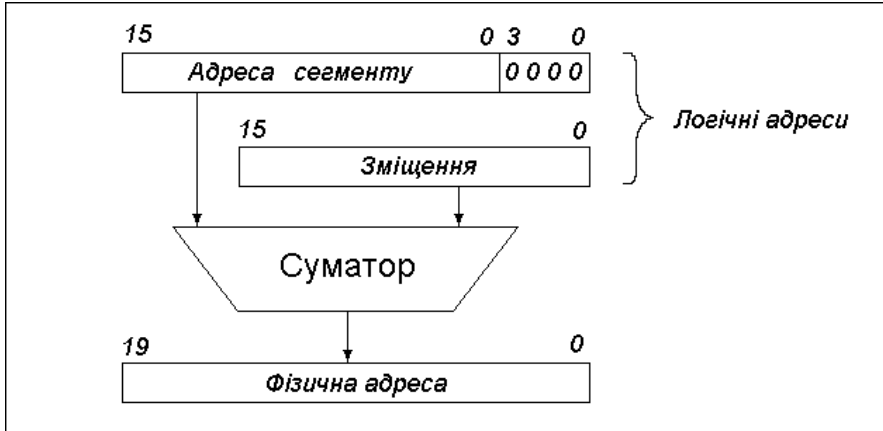


Рисунок 4.3 - Обчислювання фізичної адреси

#### 4.3.3 Розподіл пам'яті

Адресний простір пам'яті являє собою одномірний масив байт, кожний з яких має 20-бітні фізичні адреси. Байт являє мінімальну адресуєму одиницю пам'яті. Будь які два суміжні байта в пам'яті утворюють 16-бітне слово. Молодший байт слова має меншу адресу, а старший - більшу.

Фізично пам'ять **18086** організується у вигляді двох банків по 512 Кбайт. Старший банк зберігає старші байти, молодший - молодші байти слова. Обидва банки адресуються паралельно, а доступ до них здійснюється сигналами вибірки банків, які формуються процесором.

При ***VHE=0*** - вибірка байта по старшій половині шини даних (непарний адресат), при ***A0=0*** - парні адреси (***A0*** - молодший розряд адреси).

Команди, байти та слова даних можна вільно розташовувати за будь якою адресою байта, що економить пам'ять за рахунок щільної упаковки інформації. Але доцільно розташовувати слова в пам'яті по парним адресам, тому що ***МП*** може передавати такі слова за один цикл звернення до пам'яті.

Слова з парною адресою мають назву *вирівняних на межі слова*. Слова з непарними адресами (невирівняні) також припустимі, але для їх передачі потребуються два цикли вибірки, що знижує продуктивність МП. Особливо важливо мати вирівняні слова для операцій зі стеком, тому що в них приймають участь тільки слова.

#### 4.4 Режими роботи

Особливістю **I8086** є можливість програмування функцій групи управляючих виводів в залежності від того, в якій системі буде використовуватись МП. Така настройка здійснюється за допомогою входу **MN/MX** (мінімальний/ максимальний).

В мінімальному режимі (до входу **MN/MX** підключені джерела живлення +5В) МП орієнтований для використання в невеликих однопроцесорних системах, які складаються з обмеженої кількості модулів. В цьому режимі він самостійно виробляє сигнали на всіх управляючих виводах.

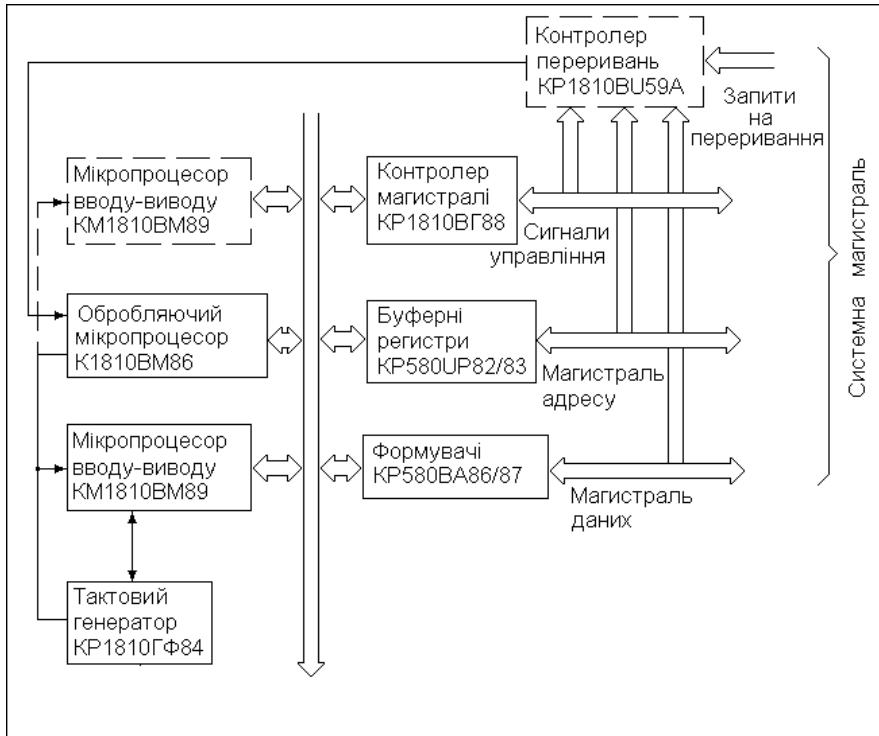
В максимальному режимі (до входу **MN/MX** підключений нульовий потенціал) МП орієнтований на використання в мультипроцесорних системах з організацією системної шини типу **MULTIBUS** (Рисунок 4.4).

В такому режимі МП не виробляє самостійно значну частину управляючих сигналів, а видає на контролер шини код стану, з якого контролер формує сигнали, необхідні для організації циклу обміну з пам'яттю та зовнішніми пристроями.

#### 4.6 Система та формат команд I8086

Система команд **I8086** має практично всі команди, які використовуються в **I8080**, а також виконуючі істотно нові функції:

- команди множення та ділення двійкових та десяткових чисел зі знаком або без знака;
- команди роботи з масивами чисел: переміщення, сканування та порівняння масивів даних довжиною до 64 Кбайт;



**Рисунок 4.4 - Схема реалізації максимального режиму роботи I8086**

- команди тестування змісту окремих чисел;
- команди перетворення кодів з одного типу в інший;
- команди програмного формування переривань;
- група команд, які використовуються для координації роботи мультимікропроцесорних систем.

Майже всі вони можуть працювати як з байтами, так і з двохбайтними словами. Вказівка до праці з тою або іншою довжиною даних міститься в коді команди.

Для реалізації команд використовуюся одно-, двох-, трьох-, чотирьох-, -п'яти- та шостибайтні формати. Перші один або два байта команди або їх частини відведені для кодів операції, а останні байти - для операндів, виконавчих адрес та зміщень. Таким чином, в МП **18086** реалізовані нуль-, одно- та двоадресні команди, загальна кількість яких дорівнює 11986.

Мінімальний цикл шини, тобто час, необхідний для передачі порції інформації між **МП** та **ОП** або периферійним пристроєм, - 4 такта.

## 5 RISC-АРХІТЕКТУРА

### 5.1 Загальні принципи реалізації

Сучасні ЕОМ характеризуються постійно зростаючою універсальністю вживання, що потребує різкого збільшення числа команд і приводить до невиправданого ускладнення апаратури та серйозним труднощам при тестуванні та налагодженні ЕОМ. Крім того, продуктивність ЕОМ тим менше, чим складніше є команди, які реалізуються, та чим частіше здійснюється звернення до пам'яті.

Перші розробки комп'ютерів із **RISC (Reduced Instruction Set Computer)**-архітектурою відносяться до початку 80-років. Скорочення набору команд виробляється за рахунок команд, які статистично рідко використовуються (в універсальних процесорах 80% усіх команд використовуються на протязі 20% машинного часу). При цьому розбивання складних операцій на базові елементарні команди, такі, як "записати", "скласти", "зрівняти", "розгалуження" тощо дозволяє збільшити продуктивність комп'ютера.

Комбінація кількох команд в таких комп'ютерах виконується швидше, ніж одна еквівалентна цій комбінації складна команда у звичайному комп'ютері.

RISC-архітектура передбачає реалізацію в комп'ютері скороченого набору простих, але часто вживаємих команд, які виконуються за один машинний цикл процесора. Виконання більш складних, але рідко використовуваних операцій, забезпечується підпрограмами.

При цьому за один машинний цикл робиться вибірка двох операндів з регістрів, виконання операції в АЛП та запам'ятовування результату у регістрі.

#### **Принципи RISC-архітектури :**

- спрощений фіксований склад команд (30-50 команд, які мають однакову довжину та структуру);
- апаратна реалізація управління замість мікропрограм;
- виконання всіх (або по меншій мірі більшості) команд за один цикл;

– більшість команд є швидкими типа "регістр-регістр" та виконуються без звернення до ОЗП.

Доступ до пам'яті зберігається тільки в командах завантаження з пам'яті та запам'ятовування в ОЗП. Для цього процесор повинен містити достатньо велику кількість загальних реєстрів (до кількох сотень в деяких моделях).

## 5.2 Трансп'ютери

Трансп'ютери представляють собою мікропроцесори, які розраховані на роботу в мультипроцесорних системах із однотипними процесорами та апаратну підтримку обчислювальних процесів. Особливістю трансп'ютерів є наявність швидких каналів зв'язку, кожний з яких може одночасно передавати по одній магістралі дані в процесор, а по іншій - дані з нього.

Продуктивність обчислювальної системи, яка складається з  $M$  трансп'ютерів з швидкодією  $N$  оп/сек кожний, складає  $M \times N$ , що дуже важливо при розробці ЕОМ п'ятого покоління.

Проектування трансп'ютера було тісно пов'язано зі створенням мови **ОККАМ**, яка відноситься до процедурних мов високого рівня. Він дозволяє описувати систему, яка складається з набору трансп'ютерів, як ряд паралельних незалежних процесів.

Всі аспекти мови **ОККАМ** реалізовані в апаратурі трансп'ютера, в результаті чого відпадає необхідність в програмних засобах для планування процесів та організації обміну даними.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Богумирский Б.С. Руководство пользователя ПЭВМ Ч.1.- С.Петербург: ОИЛСО,1992.-480с.
2. Брамм П., Брамм Д. Микропроцессор 80386 и его программирование. -М.: Мир,1990. - 440 с.
3. Джордейн Р. Справочник программиста персональных компьютеров типа IBM PC, XT и AT. - М.: Финансы и статистика, 1992. - 541 с.
4. Майоров В.Г., Гаврилов А.И. Практический курс программирования микропроцессорных систем. - М.: Машиностроение,1989. - 272 с.
5. Микропроцессорный комплект K1810: Структура, программирование, применение. Справочная книга /Под ред. Ю.М.Казаринова. - М.: Высш. шк., 1990. - 269 с.
6. Морс С., Алберт Д. Архитектура микропроцессора 80286.- М.: Радио и связь.1990. - 299 с.
7. Мячев А.А. Интерфейсы средств вычислительной техники. Энциклопедический справочник. - М.: Радио и связь,1993. - 310 с.
8. Проектирование микропроцессорной электронно-вычислительной аппаратуры. Справочник./В.Г.Артюхов, А.А.Будняк, В.Ю.Лапий и др. /- К.: Техніка, 1988. - 260 с.
9. Сташин В.В. Проектирование цифровых устройств на однокристалльных микроконтроллерах. - М.: Энергоатомиздат,1990. - 272 с.
- 10.Шевкоплас Б.В. Микропроцессорные структуры. Инженерные решения. Справочник. - М.: Радио и связь, 1990. - 512 с.
- 11.Щелкунов Н.Н., Дианов А.П. Микропроцессорные средства и системы. - М.: Радио и связь, 1989. - 287 с.

