

# Адресний простір. Механізм адресації. Керування пам'яттю та зовнішніми пристроями. Побудова модулів пам'яті

[1 Адресний простір](#)

[2 Способи формування адрес](#)

[3 Абсолютна та неабсолютна адресація](#)

[4 Керування пам'яттю та зовнішніми пристроями](#)

[5 Модуль пам'яті](#)

## 1 Адресний простір

При підключенні різних пристроїв до системної шини виникає питання – як розрізнити ці пристрої між собою? З цією метою використовують індивідуальну адресу для кожного пристрою, підключеного до системної шини мікропроцесора. Досить часто пристрої, що підключаються до системної шини, займають цілий діапазон адрес. Оскільки звернення відбувається до кожного елементу пам'яті пристрою індивідуально, то виникає поняття діапазону адрес, що займає кожний пристрій, та розподіл адресного простору мікропроцесорного пристрою в цілому.

Під адресним простором МПС розуміють множину адрес ОЗП та ПЗП, що доступна для програм, що виконується мікропроцесором. Адресний простір мікропроцесорного пристрою представляється графічно прямокутником, одна із сторін якого відповідає розрядам комірки цього мікропроцесора, що адресується, а інша сторона – усьому діапазону доступних адрес для цього ж мікропроцесора. Як елемент адресного простору, доступного для мікропроцесора, що мінімально адресується, вибирається 8-розрядна комірка (байт).

Діапазон доступних адрес мікропроцесора визначається розрядністю шини адреси, яка співпадає з розрядністю лічильника команд і розрядністю вказівника даних цього мікропроцесора. При цьому мінімальний номер елемента пам'яті (адреса) дорівнює нулю, а максимальний – визначається з формули:  $M = 2^N - 1$ , де  $N$  – кількість розрядів шини адреси мікропроцесора.

Якщо адреса в МПП формується у вигляді 16-розрядного слова, то адресний простір дорівнює 64Кбайт, 20-розрядного – 1Мбайт і т.д. Пам'ять складається з осередків, кожному з яких привласнюється своя адреса. Адреси пам'яті можуть займати весь адресний простір (АП) або його частину, а сама пам'ять незалежно від її технічної реалізації може бути умовно представлена набором регістрів (осередків), число яких  $M$ , а розрядність -  $N$  (рисунок 1).

Розподілом пам'яті мікропроцесорної системи називають розбиття адресного простору мікропроцесора на декілька областей, кожна з яких

виділена для розміщення комірок певного елемента цієї системи: ОЗП, ПЗП або зовнішніх пристроїв. Адресний простір, що відповідає структурній схемі мікропроцесорної системи, наведеної на рисунку 2, зображений на рисунку 3.

RG0
RG1
RG2
·
·
RG m-2
RG m-1

Рисунок 1 – Умовне представлення пам'яті.

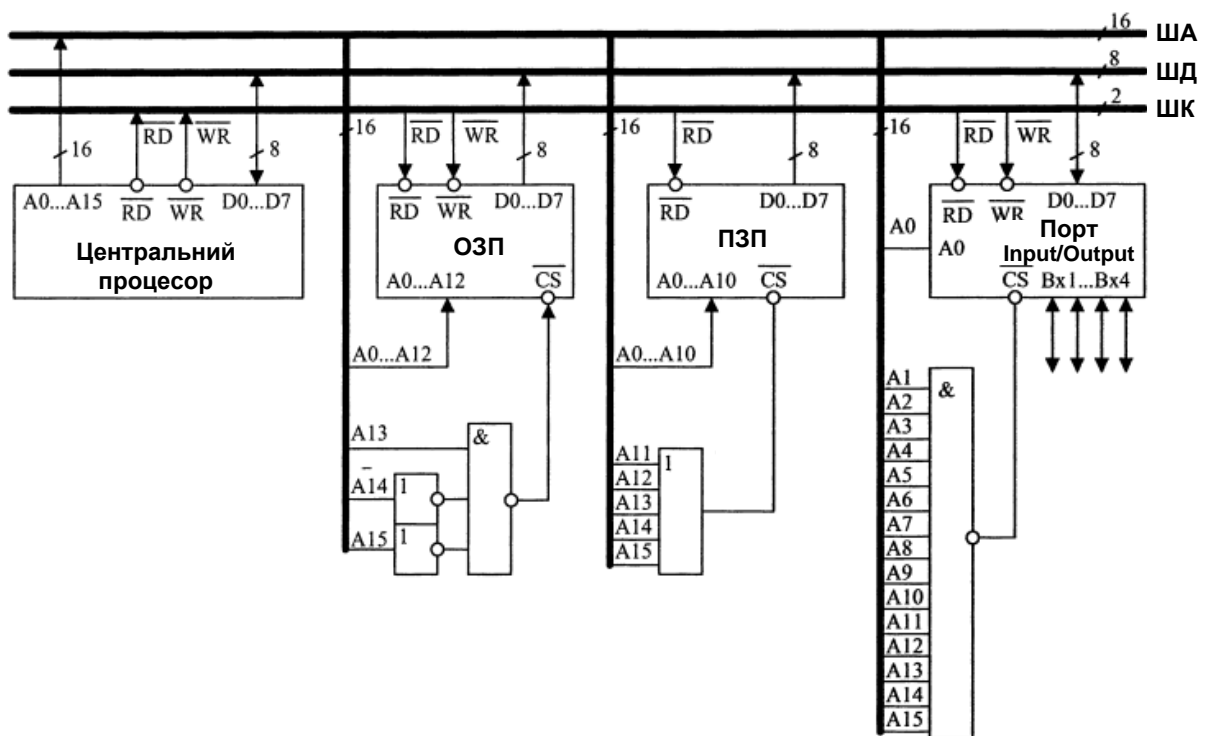


Рисунок 2 – Структурна схема мікропроцесорної системи

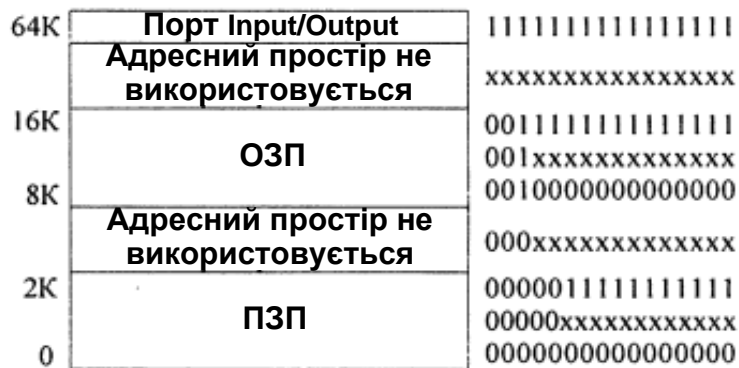


Рисунок 3 – Розподіл адресного простору 16 розрядної мікропроцесорної системи

Свої адреси мають і зовнішні пристрої (ЗП). Процесор при обміні

даними завжди повинний вибрати тільки одну з комірок пам'яті або один ЗП. Такий вибір здійснюється схемами декодування адреси. При керуванні пам'яттю і ЗП процесор повинний спочатку сформувати потрібну адресу, яка потім декодується. Адресний простір розподіляють одночасно з проектуванням структурної схеми пристрою та створення дешифраторів адреси для кожного пристрою, що підключається до системної шини.

## 2 Способи формування адрес

У МПС застосовують кілька способів формування адрес.

При *прямій адресації* код адреси утримується в команді, яка підлягає виконанню. Як приклад розглянемо команду МП i8080 прямого завантаження акумулятора вмістом комірки пам'яті, розміщеної за адресою 0012H. Формат та схему виконання цієї команди показано на рис. 2. У байті 1 команди (рисунок 4, а) міститься код операції пересилання даних в акумулятор із комірки пам'яті, а в байтах 2 і 3 - адреса комірки пам'яті. У байті 2 розташований молодший (12H), а в байті 3 - старший (00H) байт адреси.

На рисунку 4, б комірка пам'яті з адресою 0012H має вміст 11010111b. Вміст акумулятора до операції становить 00000000b. Після виконання команди значення вмісту комірки пам'яті копіюється в акумулятор.

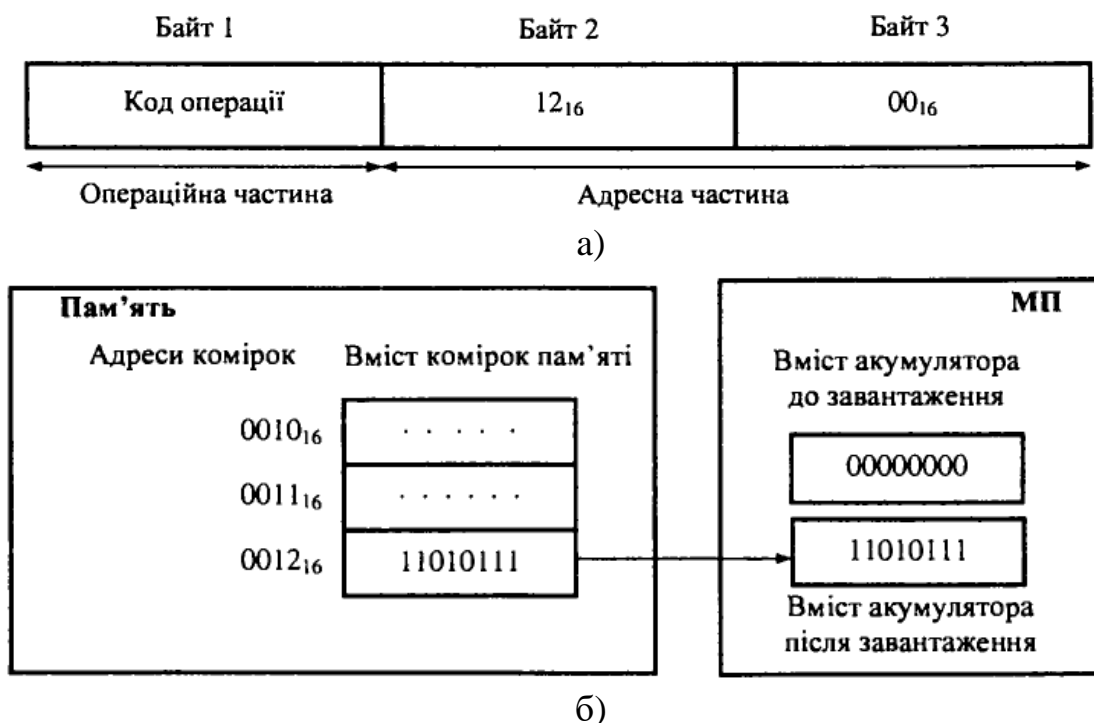


Рисунок 4 – Команда прямого завантаження в акумулятор вмісту комірки пам'яті: а) формат команди; б) схема виконання

Пряма адресація зручна, але подовжує команди (збільшує їх

розрядність), тому що при значних ємкостях пам'яті розрядності адрес досить великі. У випадку прямої регістрової адресації, коли операнд знаходиться в одному з внутрішніх регістрів процесора, адреса є мало розрядною, оскільки число таких регістрів мало. У цьому випадку пряма адресація виявляє усі свої переваги.

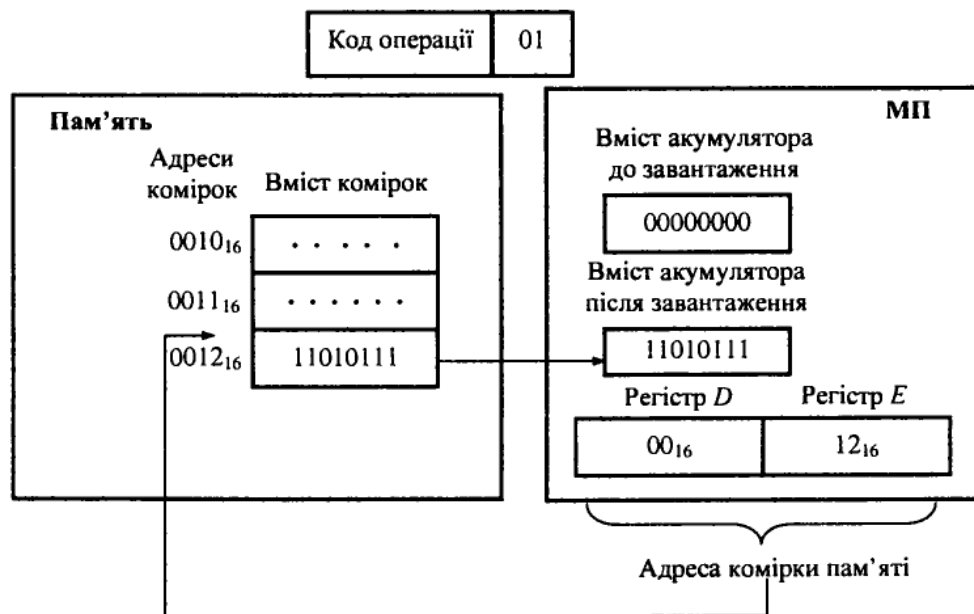


Рисунок 5 – Команда непрямого завантаження акумулятора

При непрямої адресації в команді явно або неявно вказується регістр процесора, який містить адресу операнда. Команда зберігає компактність, але для її виконання потрібна попереднє настроювання - завантаження адреси в регістр (регістр непрямої адреси). Непряма адресація зручна при обробці списків, коли настроювання виконується однократно, а чергова адреса виходить модифікацією попередніх (змінюю його на одиницю). Як приклад виконання команди непрямого завантаження в акумулятор вмісту комірки пам'яті з адресою 0012H, яка зберігається в регістровій парі DE, показано на рисунок 5.

При безпосередній адресації в команді утримується сам операнд. У першому байті команди з безпосередньою адресацією розміщується код операції. Значення операндів заносяться в команду під час програмування і знаходяться у другому або другому і третьому байтах.

Цими значеннями є деякі константи, заздалегідь відомі програмісту. У процесі виконання програми значення операндів залишаються незмінними, оскільки вони разом із командою розміщуються в ПЗП. Використання такого способу не потребує адреси операндів. Як приклад на рисунку 6 зображено формат і схему виконання команди безпосереднього завантаження акумулятора значенням 11010111b, яке зберігається у другому байті команди.

Після виконання команди це число копіюється в акумулятор. При кожному черговому зверненні до цієї команди в акумулятор записується таке саме число.

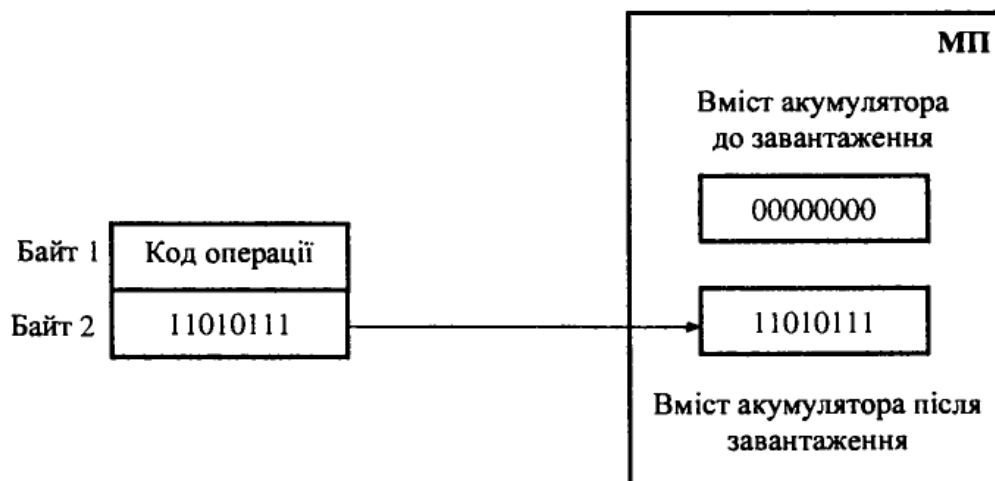


Рисунок 6 – Команда безпосереднього завантаження акумулятора

*Автоінкрементна (автодекрементна) адресація.* При автоінкрементній адресації адреса операнда обчислюється так само, як і при непрямій адресації, а потім здійснюється збільшення вмісту регістра: на один - для звернення до наступного байта, на два - для звернення до наступного слова. Розмір операнда визначається кодом операції.

*Сторінкова адресація.* Під час використання сторінкової адресації пам'ять поділяється на ряд сторінок однакової довжини. Адресація сторінок здійснюється або з програмного лічильника, або з окремого регістра сторінок. Адресація пам'яті всередині сторінок здійснюється адресою, що міститься в команді.

*Індексна адресація.* Для утворення адреси операнда до значення адресного поля команди додається значення вмісту індексного регістра, яке називається індексом.

*Відносна адресація.* При відносній адресації адреса операнда визначається додаванням вмісту програмного лічильника або іншого регістра із зазначеним у команді числом. Вміст програмного лічильника або іншого регістра називається *базовою адресою*. Для збереження базових адрес у МП можуть бути передбачені базові регістри або спеціально виділені комірки пам'яті. Тоді в адресному полі команди вказується номер базового регістра.

### 3 Абсолютна та неабсолютна адресація

Адресація може бути абсолютною або неабсолютною. При абсолютній адресації звернутися до комірки пам'яті або ЗП можна тільки по одній єдиній

адресі. При неабсолютній адресації до комірки пам'яті або ЗП можна звернутись через деяку зону адрес. Число таких зон менше ніж число окремих адрес, тому для вказівки зони буде потрібна менша розрядність адреси. Іншими словами, абсолютна адресація вимагає повного декодування адреси, а неабсолютна – часткового, що спрощує схеми декодування. Можливість використання неабсолютної адресації зв'язана з наявністю в АП "зайвого" простору. Часткою неабсолютної адресації ЗП є так звана лінійна селекція (лінійний вибір).

У простих МПС часто адресний код розглядається як би він складається з двох частин. Одна частина вказує на сторінку, у якій розташований потрібний об'єкт адресації, інша є адресою цього об'єкта на даній сторінці. Сторінкою є та або інша частина АП (яка саме - залежить від організації мікросхем, з яких будується модуль пам'яті).

З погляду використання АП пам'яттю і ЗП розрізняють концепції інтерфейсу з загальною шиною і окремою шиною.

У рамках першої концепції для адрес пам'яті і ЗП виділяються частини загального АП. До ЗП звернення відбувається так само, як і до комірок пам'яті, тобто за допомогою тих же команд і тієї ж шини. Недоліком цієї концепції є звуження АП для пам'яті, оскільки частина АП займається зовнішніми пристроями. Перевага полягає в тому, що над даними, що одержуються від ЗП, можна робити тіж операції, що мають в системі команд процесора для даних, які знаходяться в комірках пам'яті. Таких операцій багато і це сприяє поліпшенню параметрів програм і спрощенню програмування. Концепцію "із загальною шиною" називають також введенням/виведенням, що відображений на пам'ять.

У концепції "з окремою шиною" комірки пам'яті і ЗП мають свій АП. При цьому потрібна наявність сигналів керування, що визначають, з яким типом об'єктів ведеться обмін. Наприклад, вводиться сигнал ІО/М, який вказує, адресується пам'ять або ЗП. При цьому пам'ять може використовувати весь АП. Для обміну з ЗП мають тільки операції введення ІN port і виведення ОUТ port, і губиться можливість застосовувати до даних від ЗП широкий набір команд, які мають для роботи з даними, збереженими в пам'яті.

АП використовується блоками ОЗП, ПЗП і ЗП, до яких звертається процесор. Розподіл АП виконується проектувальником системи. Для стислості записів адреси в АП застосовується шістнадцятирична система числення, для оцінки ємностей АП використовується часто одиниця виміру  $Kіло = 2^{10} = 1024$  або  $Мега = 2^{20} = 1048576$ .

#### 4 Керування пам'яттю та зовнішніми пристроями

Адресація – тільки частина процесу керування пам'яттю і ЗП. Крім адрес потрібні строби читання і запису ( $\overline{RD}$  і  $\overline{WR}$ ), які задають напрямок обміну, сигнали дозволу роботи ( $\overline{CS}$ , EN), ознака звертання до ЗП або пам'яті (IO/M). Процесор виробляє мінімальну групу сигналів, тоді як у системному інтерфейсі може бути передбачена трохи інша група. Зокрема, мікропроцесор дає три сигнали: сигнал читання ( $\overline{RD}$ ), запису ( $\overline{WR}$ ) і сигнал IO/M, тобто звертання до ЗП при високому рівні і до пам'яті - при низькому. У системному ж інтерфейсі використовується система з чотирьох сигналів: сигналу читання з пам'яті  $\overline{MEMR}$ , запису до пам'яті  $\overline{MEMW}$ , читання з ЗП  $\overline{IOR}$  і запису до ЗП  $\overline{IOW}$ .

До четвірки сигналів легко перейти за такими співвідношеннями:

$$\overline{MEMR} = \overline{RD} \cdot \overline{IO/M} = \overline{RD} \vee \overline{IO/M}$$

$$\overline{MEMW} = \overline{WR} \cdot \overline{IO/M} = \overline{WR} \vee \overline{IO/M}$$

$$\overline{IOR} = \overline{RD} \cdot \overline{IO/M} = \overline{RD} \vee \overline{IO/M}$$

$$\overline{IOW} = \overline{WR} \cdot \overline{IO/M} = \overline{WR} \vee \overline{IO/M}$$

Статичні ОЗП можуть бути асинхронними або синхронними. Для синхронних ОЗП потрібний імпульсний характер сигналу керування (сигналу  $\overline{CS}$ ). У цьому випадку для повторного дозволу роботи пам'яті потрібно попередньо повернути сигнал у пасивний стан. Для додання сигналові імпульсного характеру можна застосувати співвідношення  $\overline{CS} = \overline{MEMR} \cdot \overline{MEMW}$ . При цьому забезпечується пасивний стан сигналу  $\overline{CS}$  на інтервалах, на яких не діють ні сигнал читання, ні сигнал запису ( $\overline{MEMR} = \overline{MEMW} = 1$ ).

Іноді умовою обміну є готовність до нього пам'яті або ЗП. Для виявлення готовності застосовують такий метод: поява адреси повільного пристрою веде до запуску генератора одиночного імпульсу необхідної тривалості, на час існування якого сигнал готовності RDY знімається. Тривалість інтервалу неготовності розраховується відповідно до вимог повільного пристрою. Процесор чекає появи сигналу готовності і тільки після його появи виконує операцію обміну. Щоб уникнути втрат часу, бажано генерувати інтервал неготовності з прив'язкою його до синхроімпульсів МПС.

Розпочнемо розподіл адресного простору мікропроцесорної системи з виділення діапазону адрес для мікросхеми ПЗП.

Мікропроцесори після включення живлення і виконання процедури скидання завжди розпочинають виконання програми з певної адреси, частіше

усієї нульової. Проте є і виключення. Наприклад, процесори, на основі яких будуються універсальні комп'ютери IBM PC або Macintosh, стартують не з нульової адреси. Виконувана програма або її завантажувач повинні зберігатися в пам'яті мікропроцесорної системи, яка не стирається при виключенні живлення, тобто у ПЗП. Таким чином, адреса, що записана в лічильник команд процесора після виконання його скидання, обов'язково повинна потрапляти в діапазон адрес, виділених для розміщення ПЗП.

Виберемо для побудови мікропроцесорної системи мікросхему ПЗП об'ємом 2 Кбайт, як це показано на рисунку 2 і 3. Мікропроцесор після зняття сигналу скидання RESET розпочинає роботу з нульової адреси, тому для ПЗП в адресному просторі виділимо номери комірок, починаючи з нульової адреси.

Якщо просто підключити адресні входи ПЗП до відповідних виводів адресної шини, то залишиться ще п'ять старших розрядів цієї шини. Чи можна залишити ці розряди незадіяними? Ні в якому разі! Якщо не використати ці розряди, то комірки ПЗП займуть увесь адресний простір мікропроцесора. Зверніть увагу (рисунок 3) на двійкове представлення адреси елементів пам'яті. Молодші одинадцять розрядів адреси в адресному просторі повторюються 32 рази! Тому для того, щоб ПЗП реагував тільки на свої адреси, потрібно було ввести додатковий пристрій – дешифратор адреси.

Дешифратор адреси є комбінаційним пристроєм, що дозволяє виділити певну комбінацію двійкових символів і подати сигнал дозволу на вивід вибору кристала ПЗП. Для того, щоб нульова комірка ПЗП була розташована за нульовою адресою адресного простору мікропроцесора, старші розряди шини адреси (5 старших розрядів, починаючи з розряду A11) мають дорівнювати 0. Саме на таку комбінацію слід побудувати дешифратор. Це виконується за допомогою зовнішнього дешифратора адреси, яка в даному випадку вироджується в 5-входову схему "АБО".

При використанні дешифратора адреси звернення мікропроцесора за межі нижньої області 2 Кбайт не приведе до читання комірок ПЗП тому, що в цьому випадку на вході вибору кристала CS рівень напруги залишиться високим (неактивним).

Підключимо до системної шини мікропроцесорної системи ОЗП. Для прикладу виберемо мікросхему об'ємом 8 Кбайт. Вибір будь-якого з елементів пам'яті цієї мікросхеми можливий за допомогою 13-розрядної адреси, тому необхідно додатково декодувати сигнали трьох старших ліній 16-розрядної шини адреси. Оскільки початкові елементи пам'яті адресного простору вже зайняті ПЗП, то їх використовувати не можна. Їм відповідає значення старших розрядів адресної шини 000. Виберемо для адресації ОЗП



комбінацію сигналів 001. Дешифратор адреси вироджується в даному випадку в 3-входову схему "I-HE" з двома інверторами на вході. Принципова схема дешифратора адреси ОЗП наведена на рисунку 2. Цей дешифратор адреси забезпечує нульовий рівень сигналу на вході CS мікросхеми ОЗП тільки при комбінації старших біт адреси A15 ... A13, які рівні 001.

*Зверніть увагу:* так як об'єм мікросхеми ПЗП менше об'єму мікросхеми ОЗП, то між областями адрес мікросхем ПЗП і ОЗП утворився порожній простір невживаних адрес пам'яті мікропроцесорної системи. Його можна було б уникнути при ускладненні схеми дешифратора адреси ОЗП або застосування додаткового двійкового суматора, проте чи варто це робити? Це, окрім ускладнення схеми пристрою, призводить до зниження швидкодії мікропроцесорної системи в цілому. Якщо мікропроцесорна система не ускладнюватиметься, то навіщо ускладнювати схему дешифратора адреси, а якщо буде, то, швидше за все, буде збільшений об'єм ПЗП і тоді цей резерв адрес буде просто корисний.

Вважатимемо, що порт введення-виведення відображається в адресний простір мікропроцесорного пристрою як поодинокий елемент пам'яті, тому можна вибрати практично будь-яку вільну адресу.

Звернемо увагу на те, що порт є групою адрес, що складається тільки з двох елементів пам'яті, то доводиться декодувати 15-розрядну адресу. При цьому, чим більший номер матиме цю адресу, тим більша кількість одиниць буде присутньою в двійковому представленні цього числа, і тим простіше буде реалізовуватися дешифратор адреси відповідного пристрою.

Найпростіше побудувати дешифратор самої старшої адреси – 65535. Це число відповідає шістнадцятирічному – FFFFh. У цьому випадку дешифратор адреси вироджується в звичайну 15-входову схему "15I-HE", тому і виберемо цю адресу для розміщення порту введення-виведення. Саме такий дешифратор і зображений на схемі, що наведена на рисунку 2.

## **5 Модуль пам'яті**

Модуль пам'яті складається не з однієї мікросхеми, а з декількох. Для мікросхем пам'яті типова організація  $2^k \times l$ , де  $k$  - парне число;  $2^k$  - число збережених слів;  $l$  - розрядність слів. Якщо потрібен модуль пам'яті з організацією  $2m \times n$ , а маються мікросхеми з організацією  $2^k \times l$ , де  $k < m$  і  $l < n$ , то при сторінковій організації модуля його склад і структура визначаються так.

Для нарощування розрядності збережених слів до необхідної включаються паралельно кілька мікросхем ( $n / l$ ). Це утворить субмодуль (сторінку), яка зберігає  $2^k$  слів.

Для збільшення числа збережених слів до  $2^m$  потрібно взяти  $2^{m-k}$  субмодулей. Адреса слова в межах субмодуля вказується  $k$  молодшими розрядами адреси, які надходять безпосередньо на адресні входи мікросхем, а старші розряди адреси використовуються для формування сигналу дозволу роботи того або іншого субмодуля (рисунок 7).

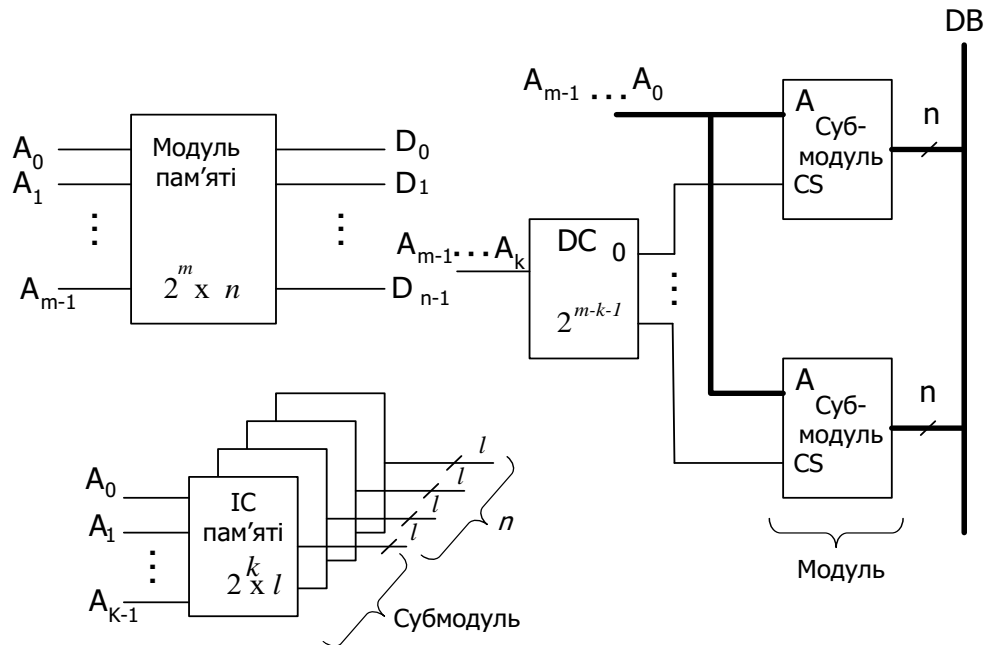


Рисунок 7 – Структура модуля пам'яті.