



Комп'ютерна схемотехніка та архітектура комп'ютерів

Адресний простір. Механізм адресації.

Керування пам'яттю та зовнішніми пристроями.

Побудова модулів пам'яті

Адресний простір

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).

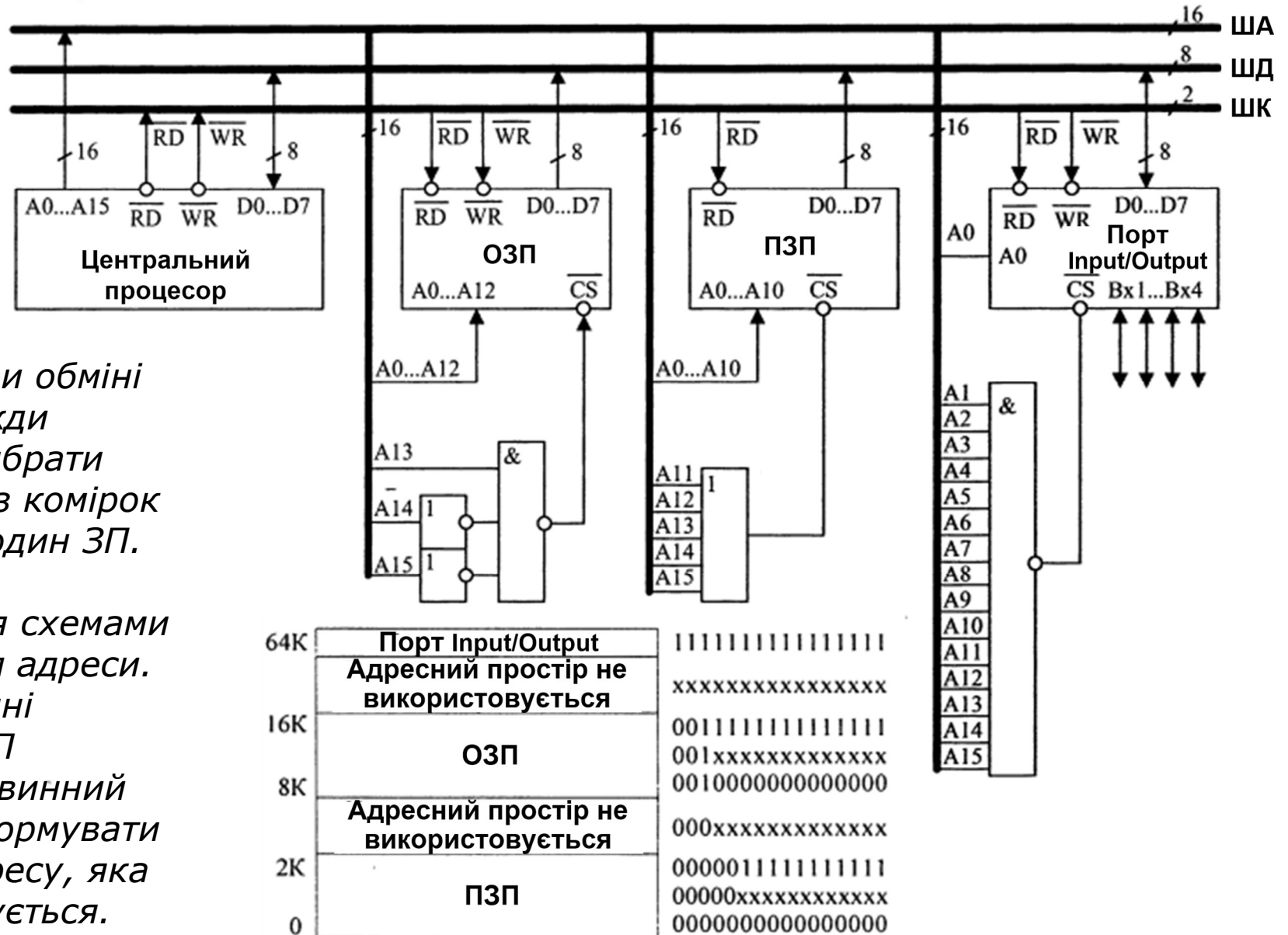
Діапазон доступних адрес МП визначається **розрядністю ША**, яка співпадає з розрядністю лічильника команд і розрядністю вказівника даних цього мікропроцесора. При цьому мінімальний номер елемента пам'яті (адреса) дорівнює нулю, а максимальний – визначається з формули: $M = 2^{N-1}$, де N – кількість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

Якщо адреса в МПС 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

RG0
RG1
RG2
⋮
RG m-2
RG m-1

Адресний простір

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



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 сформулювати потрібну адресу, яка потім декодується.

Способи формування адрес

При **прямій адресації** код адреси утримується в команді, яка підлягає виконанню (LDA 4050H).

При **непрямій адресації** в команді явно або неявно вказується регістр процесора, який містить адресу операнда. Команда зберігає компактність, але для її виконання потрібна попереднє завантаження адреси в регістр (регістр непрямої адреси). (LDAX B – завантажити в A число, адреса якого в реєстровій парі BC).

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 (MVI A, 2DH).

Автоінкрементна (автодекрементна) адресація. При автоінкрементній адресації адреса операнда обчислюється так само, як і при непрямій адресації, а потім здійснюється збільшення вмісту реєстра: на один - для звернення до наступного байта, на два - для звернення до наступного слова (INX SP – збільшити адресу вказівника стеку на 1).

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 реєстра 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 реєстра.

Відносна адресація. При відносній адресації адреса операнда визначається додаванням вмісту програмного лічильника або іншого реєстра із зазначеним у команді числом.

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.

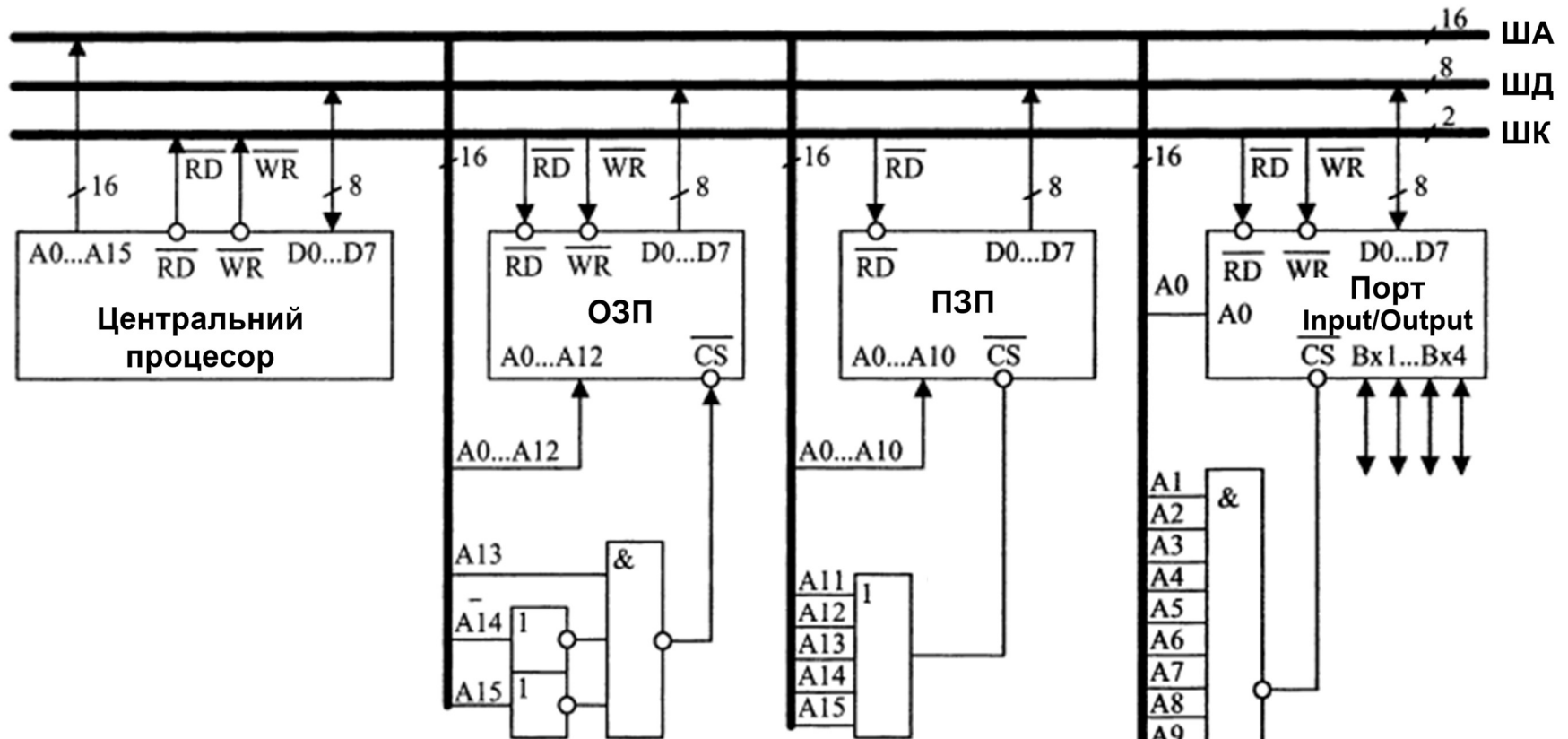
У простих МПС часто адресний код складається з двох частин. Одна частина **вказує на сторінку**, у якій розташований потрібний об'єкт адресації, інша є **адресою об'єкта на даній сторінці**.

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

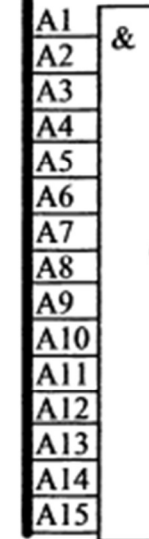
У рамках І-ї концепції для адрес пам'яті і ЗП виділяються частини загального АП. До ЗП звернення відбувається так само, як і до комірок пам'яті, тобто за допомогою тих же команд і тієї ж шини. Недоліком цієї концепції є звуження АП для пам'яті, оскільки частина АП займається ЗП. Перевага полягає в тому,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

У концепції "з окремою шиною" комірки пам'яті та ЗП мають свій АП. При цьому потрібна наявність сигналів керування, що визначають, з яким типом об'єктів ведеться обмін. Сигнал **ІО/М** вказує, адресується пам'ять або ЗП. При цьому пам'ять може використовувати весь АП. Для обміну з ЗП маються тільки операції *IN port* та *OUT port*, й губиться можливість застосовувати до ЗП широкий набір команд, які маються для роботи з даними.

Адресний простір



64K	Порт Input/Output	1111111111111111
	Адресний простір не використовується	xxxxxxxxxxxxxxxxxx
16K	ОЗП	0011111111111111
		001xxxxxxxxxxxxxxxxx
8K	Адресний простір не використовується	0010000000000000
		000xxxxxxxxxxxxxxxxx
2K	ПЗП	0000011111111111
		00000xxxxxxxxxxxxxxxxx
0		0000000000000000

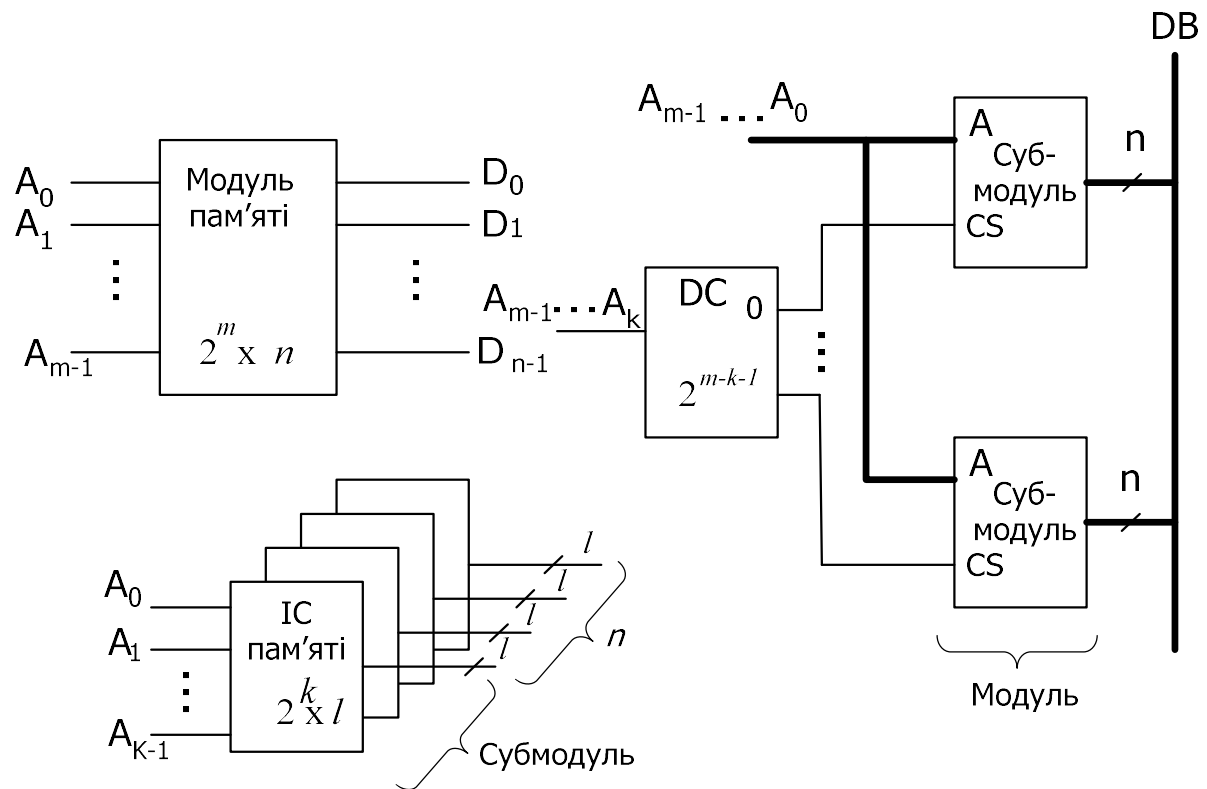


Модуль пам'яті

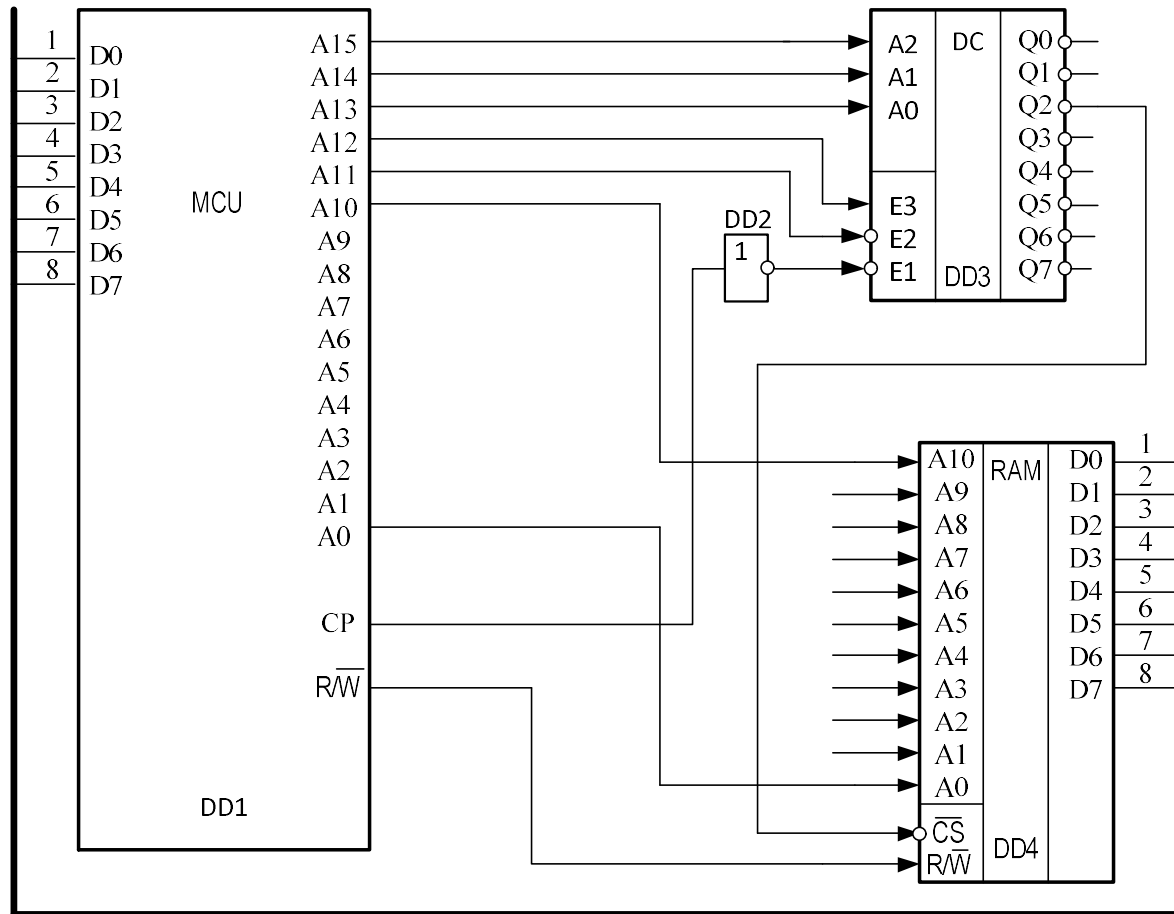
Модуль пам'яті складається не з однієї мікросхеми, а з декількох. Для мікросхем пам'яті типова організація $2^k \times l$, де k - парне число; 2^k - число збережених слів; l - розрядність слів. Якщо потрібен модуль пам'яті з організацією $2^m \times n$, а маються мікросхеми з організацією $2^k \times l$, де $k < m$ та $l < n$, 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



Проектування схем підключення мікросхем пам'яті

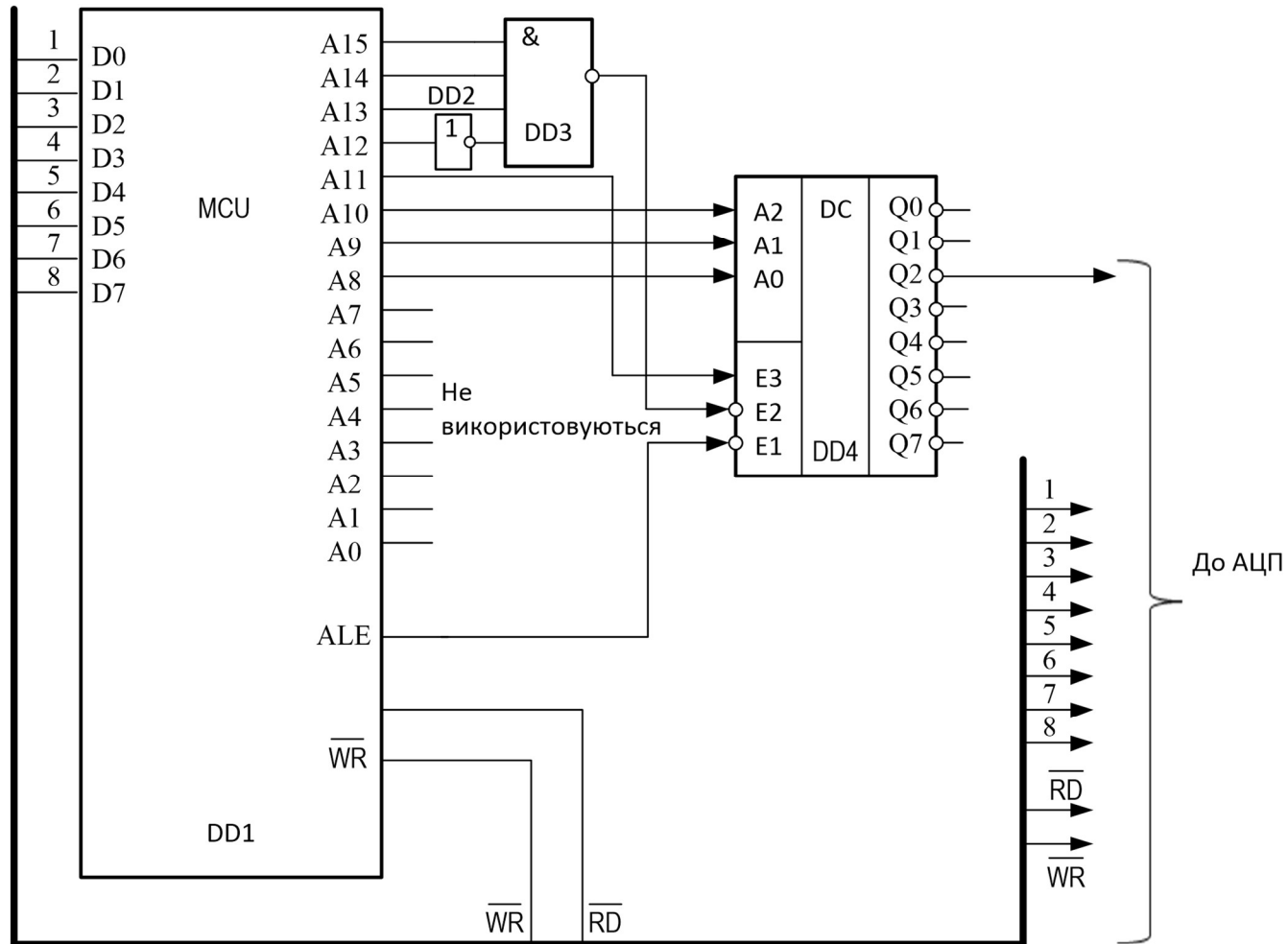


1. МП поміщає адресу пам'яті на виводи адреси A15–A0.
2. МП формує сигнал R/W, який інформує модуль пам'яті о режимі роботи (1 – READ, 0 – WRITE).
3. П'ять старших біт адреси керують входом дозволу роботи модуля пам'яті (CS).
4. Решта 11 біт адреси поступають на модуль пам'яті. Модуль DD4 використовує їх для вибірки конкретної комірки всередині мікросхеми, доступ до якої запитує МП

Визначте організацію та об'єм ОЗП

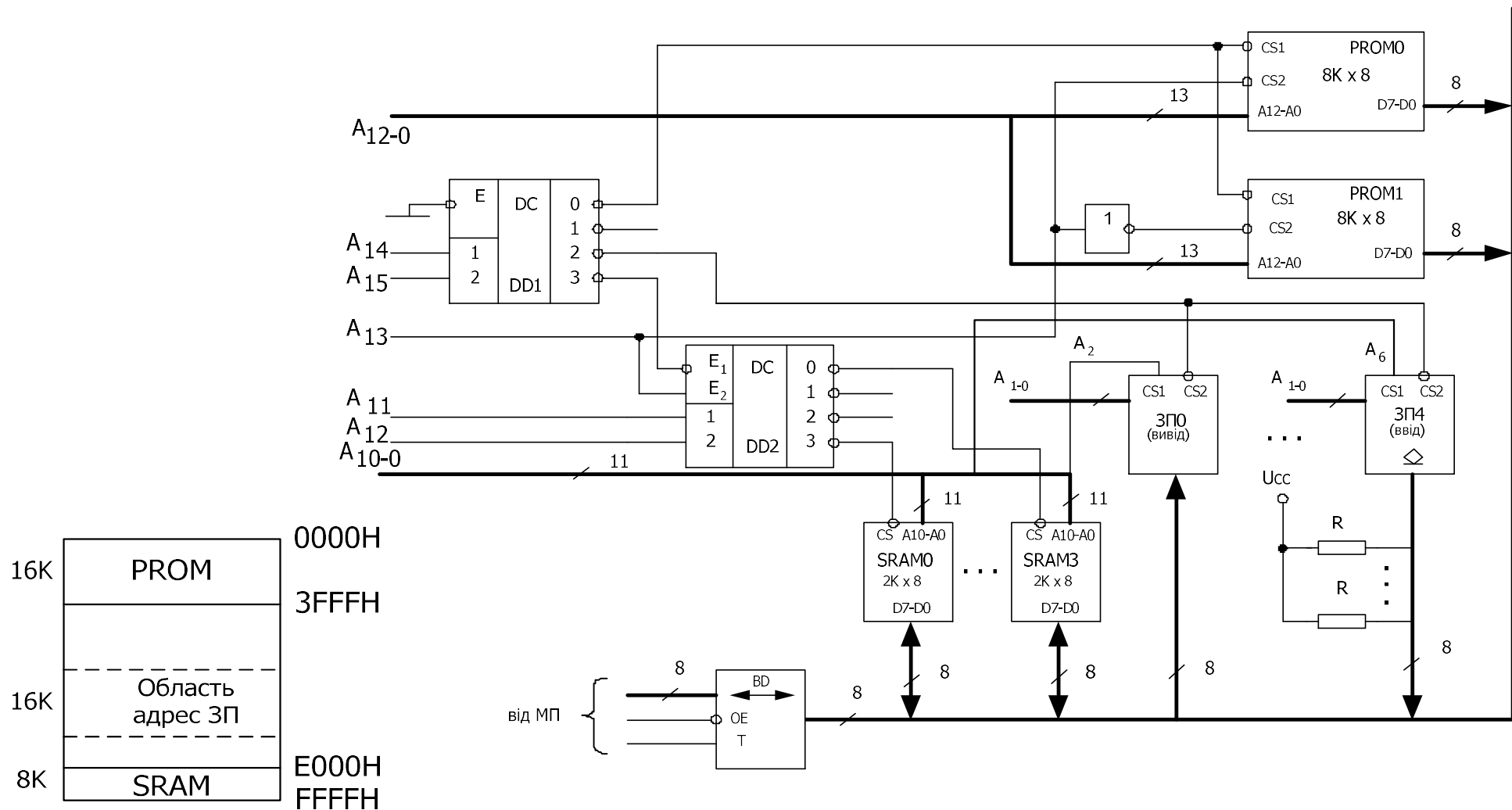
1. Намалюйте адресний простір МПС та показати де розташований модуль ОЗП.
2. Визначте, який діапазон шістнадцятирічних адрес може активізувати модуль ОЗП.
3. Визначити чи можуть шістнадцятирічні адреси 607F, 57FA, 5F00, 15D0, AFF3 активізувати модуль пам'яті, і якщо так, то які з них.
4. Наведіть схему керування модулями ОЗП, якщо початковий об'єм ОЗП збільшений у два рази.

Проектування схем підключення мікросхем пам'яті



На рис. показаний один з способів реалізації логіки дешифрації адреси. Сигнал вибору мікросхеми, який активізує АЦП, знімається з 8 старших ліній шини адреси мікропроцесора. Якщо МП потрібно обмінятися інформацією з АЦП, то він видає на шину адресу, а логіка дешифрації адреса переводить сигнал в стан з низьким рівнем напруги. Необхідно: **а) визначити адресу АЦП;** **б) змінити схему таким чином, щоб АЦП мав адресу E8XXh,** **в) змінити схему таким чином, щоб АЦП мав адресу FFXXh**

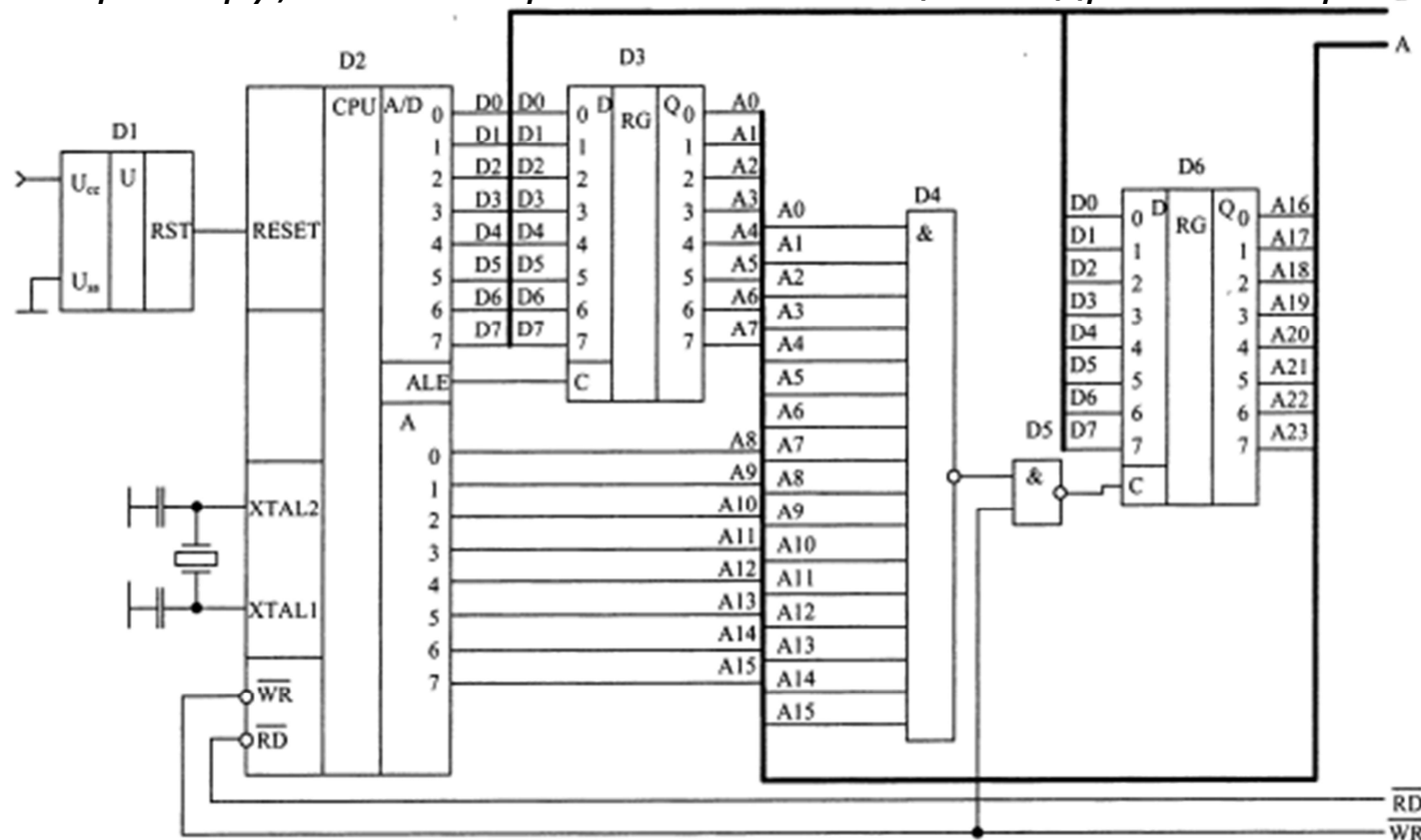
Проектування схем підключення мікросхем пам'яті



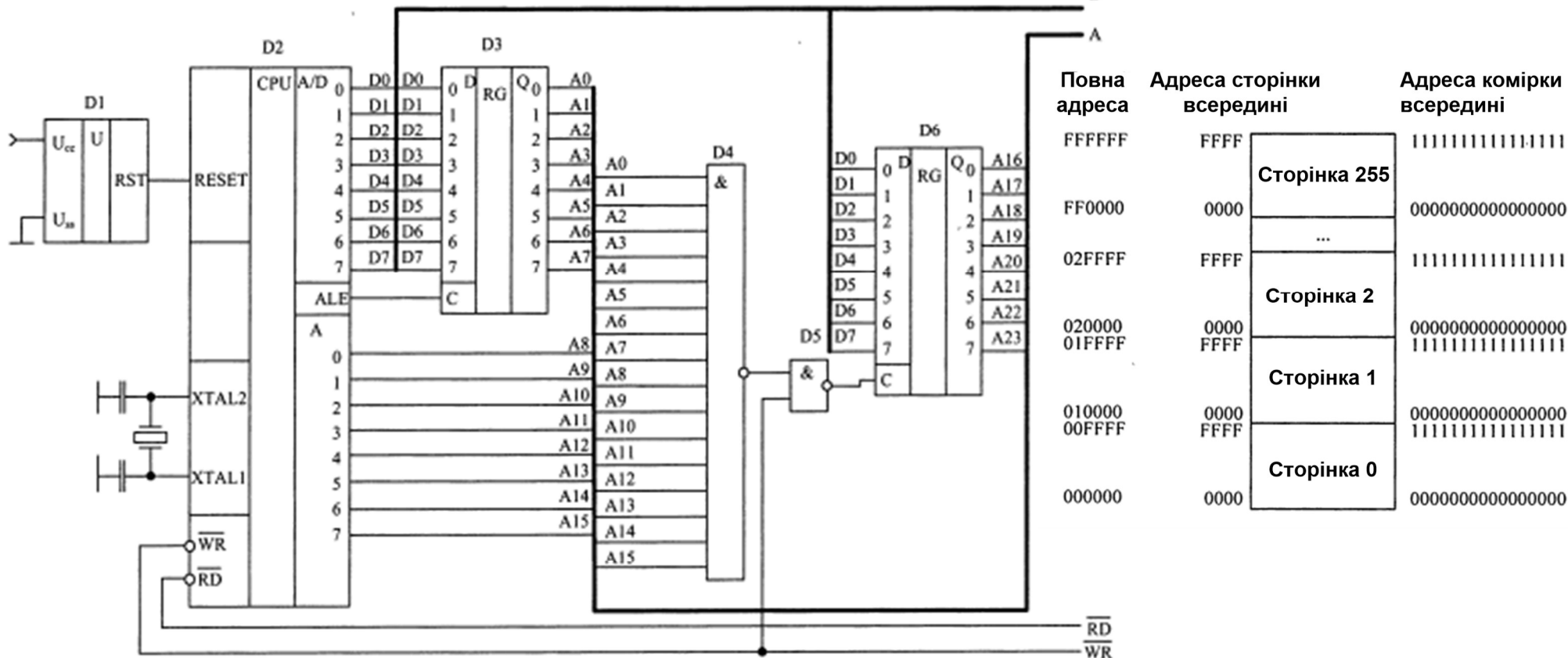
Для пам'яті використовуємо абсолютну адресацію, а для ЗП - лінійну селекцію. Нехай для ПЗП відведено 16К адреси на початку АП, адреси ЗП займають третю чверть АП, а адреси ОЗП займають останні 8К адресного простору. Прийmemo, що в системі мається 5 ЗП, кожний з яких має 4 внутрішніх регістри зі своїми адресами, а як ОЗП використовується тригерне ЗП.

Метод сторінкового розширення адресного простору

Для розширення адресного простору МПС можна скористатися схемою, подібною до паралельного порту. Виходи реєстра використовуватимуть як старші розряди адресної шини. При додаванні до адресної шини одного додаткового розряду подвоюється адресний простір мікропроцесорної системи. Записавши в цей біт нульове значення, отримаємо колишній варіант мікропроцесорної системи, а записавши одиничне значення, звертатимемося до додаткового адресного простору. Додавши до адресної шини вісім додаткових біт, отримаємо 256 адресних просторів, подібних до первинного адресного простору. Записуючи в додаткові розряди адреси число, що відповідає номеру адресного простору, можна перемикатися між цими адресними просторами.



Метод сторінкового розширення адресного простору



Початковий адресний простір еквівалентно сторінці книги, а перемикання між адресними просторами еквівалентно перегортанню сторінок в цій книзі. Тому описаний метод розширення адресного простору дістав назву сторінкового методу адресації

На цій схемі для економії зовнішніх виводів ШД і ША об'єднані і для їх розділення використовується спеціальний сигнал – ALE. Розділення адрес і даних робить паралельний регістр D3, який запам'ятовує молодший байт адреси по сигналу ALE. При застосуванні восьмирозрядного перемикача сторінок в МПС з'явилися додаткові вісім ліній адреси. В результаті адресний простір мікропроцесорної системи збільшився до $2^{24} = 16$ Мбайт.

Метод сегментного розширення адресного простору

Метод сторінкової адресації простий в реалізації, і при формуванні адреси фізичної пам'яті не призводить до додаткових часових затримок. Проте при використанні багатозадачного режиму роботи процесора для кожного активного завдання виділяється ціла сторінка в системній пам'яті мікропроцесора. Якщо програмний код завдання не займає повністю сторінку, то в системній пам'яті процесора залишається багато невживаних областей. Розв'язати виниклу проблему дозволяє [метод сегментної організації пам'яті](#).

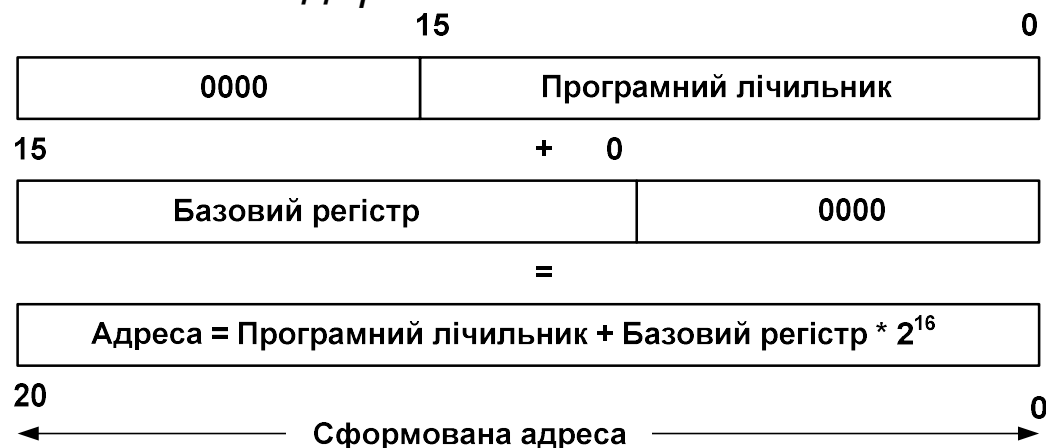
Для розширення адресного простору використовується **базовий реєстр**, відносно якого робиться адресація команд або даних в програмі. **Розрядність базового реєстра вибирають рівній розрядності лічильника команд**. Для звернення до базового реєстра диспетчера пам'яті, як і при сторінковій організації пам'яті, можна використати схему, подібну до схеми паралельного порту. Для відображення 16-розрядного реєстра в адресному просторі 8-розрядного мікропроцесора знадобиться вже два елементи пам'яті

Приклад розподілу адресного простору при використанні сегментного методу адресації і різному розмірі програмних сегментів наведений на рис. Як видно з цього рисунку, розміри окремих сегментів пам'яті на відміну від розмірів сторінок можуть бути різними. Це пов'язано з тим, що вікна (розмір вікна залишається тим самим – 64 Кбайт) при сегментній організації пам'яті можуть перекриватися, і якщо частина пам'яті у вікні, виділеному для попередньої програми, не використовується, то наступне вікно можна розмістити, починаючи з першого вільного елемента пам'яті.

Повна адреса	Адреса сегменту всередині	
FFFFFF		Не використаний простір
17030 1702F	7FFF	Сегмент 2
0F030	0000	...
0F028	0018	Сегмент 1
0F010	0000	...
0F005	F005	Сегмент 0
00000	0000	

Метод сегментного розширення адресного простору

Для формування фізичної адреси, при сегментному виді розширення адресного простору МПС, використовується паралельний двійковий суматор. На входи цього суматора подається вміст базового реєстра і лічильника команд. **Додавання робиться зі зміщенням коду базового реєстра ліворуч на декілька біт відносно утримуваного лічильника команд.** У результаті максимальний розмір сегменту визначається розрядністю програмного лічильника, а максимальна невикористана область пам'яті – зміщенням базового реєстра відносно програмного лічильника. Враховуючи, що базовий реєстр зміщений відносно програмного лічильника на чотири розряди, мінімальний крок при розміщенні вікон буде $2^4 = 16$ байт, тобто в цьому випадку максимальна область невикористаної пам'яті між програмними сегментами дорівнюватиме 15 байтам.

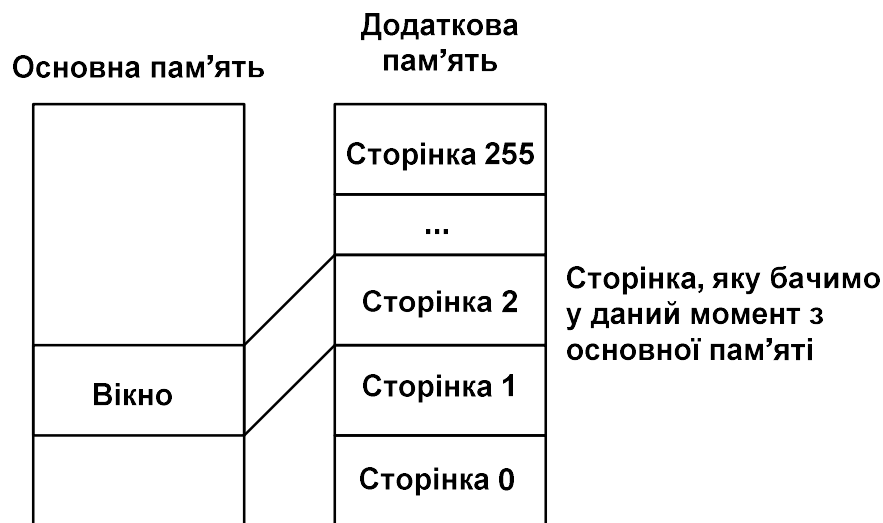


У ПК є чотири базові реєстри, що визначають сегмент даних – DS, сегмент програми – PS, сегмент стека – SS і додатковий сегмент – ES. Інформацію у базові реєстри заносить операційна система при запуску програми і перемиканні між активними програмами

Зазвичай у диспетчері пам'яті використовується не один, а декілька базових реєстрів. Це дозволяє мікропроцесору працювати з декількома сегментами одночасно. Кількість базових реєстрів визначає максимальний об'єм даних, доступних для програми.

Розширення адресного простору за допомогою вікон

При використанні вікон робиться розширення не усього адресного простору, а тільки його частини. **Усередині адресного простору виділяється деяка область, яка називається вікном. У це вікно може відображатися частина іншого адресного простору. При використанні вікон може бути застосований як сторінковий, так і сегментний метод відображення сусіднього адресного простору у вікно.** При цьому розмір сторінки, що відображається у вікно, не може перевищувати розмір самого вікна. При використанні сторінкового методу відображення, конкретна сторінка додаткового адресного простору, що відображається у вікно основної пам'яті, визначається перемикачем сторінок. Перемикач сторінок будується за принципом, розглянутим раніше



При використанні сегментного методу конкретна область адресного простору, яка відображається у вікно, визначається вмістом базового реєстра. Якщо розрядність адреси допоміжного адресного простору, що відображається у вікно основної пам'яті, співпадає з розрядністю базового реєстра, то будь-яка область допоміжної пам'яті може бути відображена в основну пам'ять з точністю до байта.

Віконна адресація часто використовується при розвитку МПС, коли розміру областей пам'яті, відведених для конкретних завдань в молодших моделях сімейства, бракує для старших моделей, а при цьому треба підтримувати апаратну сумісність з молодшими моделями сімейства.