

Організація багато кристалної пам'яті

[1 Проектування схем підключення мікросхем пам'яті](#)

[2 Способи розширення адресного простору мікропроцесора](#)

[2.1 Метод сторінкового розширення адресного простору](#)

[2.2 Метод сегментного розширення адресного простору](#)

[2.3 Метод розширення адресного простору за допомогою вікон](#)

1 Проектування схем підключення мікросхем пам'яті

При проектуванні схем підключення мікросхем пам'яті до мікропроцесора розв'язуються такі задачі:

- розробка схем адресації пам'яті і формування сигналів керування на функціонально-логічному рівні;
- аналіз навантажувальних умов в отриманій схемі, забезпечення робочих режимів для виходів з відкритим колектором (стоком) і застосування при необхідності буферних елементів для усунення перевантажень джерел сигналів;
- узгодження часових діаграм мікропроцесора і мікросхем пам'яті.

При адресації пам'яті розміщують адреси постійних і оперативних запам'ятовувальних пристроїв в заданих областях адресного простору.

Приклад 1

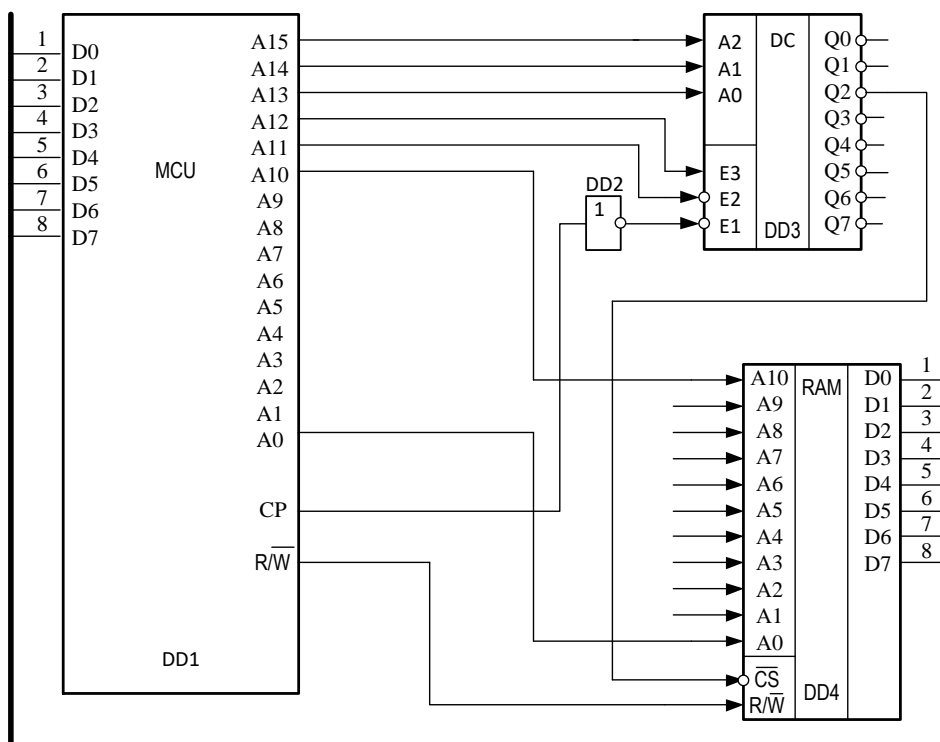


Рисунок 1 – Схема підключення модуля пам'яті

На рисунку 2 показаний один з способів реалізації логіки дешифрації адреси. Сигнал вибору мікросхеми \overline{CS} , 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 \overline{CS} 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

Приклад 3

Типовим елементом схем адресації є дешифратор, у якому використовуються як інформаційні, так і входи дозволу. На рисунку 3 наведена схема адресації ПЗП, яка складається з трьох субмодулей з організацією 4Kx8. Адреси займають 12К у верхній частині АП, тобто зону від 0000H до 2FFFH.

Сигнал дозволу роботи дешифратора $\overline{E} = E_1E_2E_3$. Дванадцять молодших розрядів адреси вибирають комірку у субмодулі. Старші розряди адреси декодуються для формування сигналів вибору кристала \overline{CS} . Сигналом \overline{RD} визначається інтервал виконання операції читання.

Однією з умов дозволу роботи дешифратора є низький рівень сигналу ІО/М.

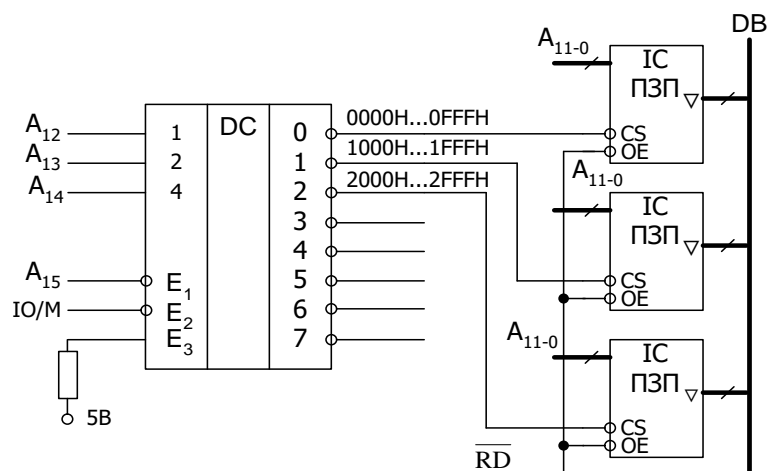


Рисунок 3 – Приклад адресації модуля пам'яті.

Приклад 4

На рисунку 4 для адресації модулів пам'яті, складеного із субмодулей з організацією 2Kx8 при розміщенні адрес у зоні адресного простору (АП), що починається з адреси 8000H, використовується дешифратор. Якщо адреса знаходиться в межах 8000H...BFFFH, то робота дешифратора дозволена, тому що цим межам відповідають умови $A_{15} = 1$ і $A_{14} = 0$.

Область АП, що лежить у зазначених межах, у залежності від значень біт $A_{13}...A_{11}$ поділяється на частини по 2К (0800Н). Кожний з виходів дешифратора сигналом \overline{CS} може вибирати мікросхему пам'яті з числом збережених слів 2К. Лінії адреси A_{10-0} адресують комірки на кристалі.

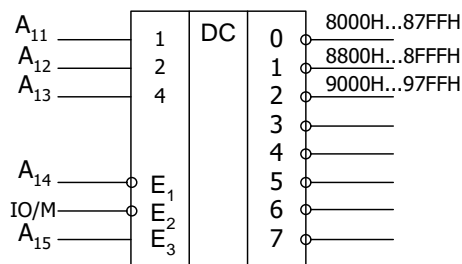


Рисунок 4 – Варіант адресації модуля пам'яті.

Приклад 5

Розглянемо приклад розміщення в АП модулів ПЗП, ОЗП і зовнішніх пристроїв (ЗП). 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. Приймемо, що в системі мається 5 ЗП, кожний з яких має 4 внутрішніх регістри зі своїми адресами, а як ОЗП використовується тригерне ЗП. Розподіл АП показаний на рисунку 5.

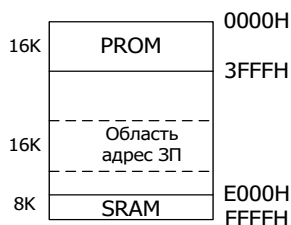


Рисунок 5 – Приклад розподілу адресного простору між модулями пам'яті і зовнішніх пристроїв.

Нехай ПЗП будується на мікросхемах з організацією 8Кх8, а ОЗП на мікросхемах 2Кх8. Для адресації ЗП використовуємо молодші розряди шини адреси, число яких визначається як $N + 2$, де N - число ЗП, а дві лінії потрібні для адресації їх внутрішніх регістрів. Схема адресації, яка відповідає таблиці адрес (таблиця 1), наведена на рисунку 6. Дешифратор DD1 розподіляє АП на чотири частини, його виходи дозволяють роботу тим об'єктам адресації, що розташовані у відповідній чверті АП. Лінія A_{13} дозволяє роботу мікросхеми PROM0 у першій половині першої чверті АП при нульовому стані і роботу мікросхеми PROM1 - при одиничному. Лінії

$A_2..A_6$ використані для лінійної селекції зовнішніх пристроїв, а лінії A_{12} і A_{11} декодуються дешифратором DC2, для дозволу роботи мікросхемам SRAM3...SRAM0 у їх зонах адрес

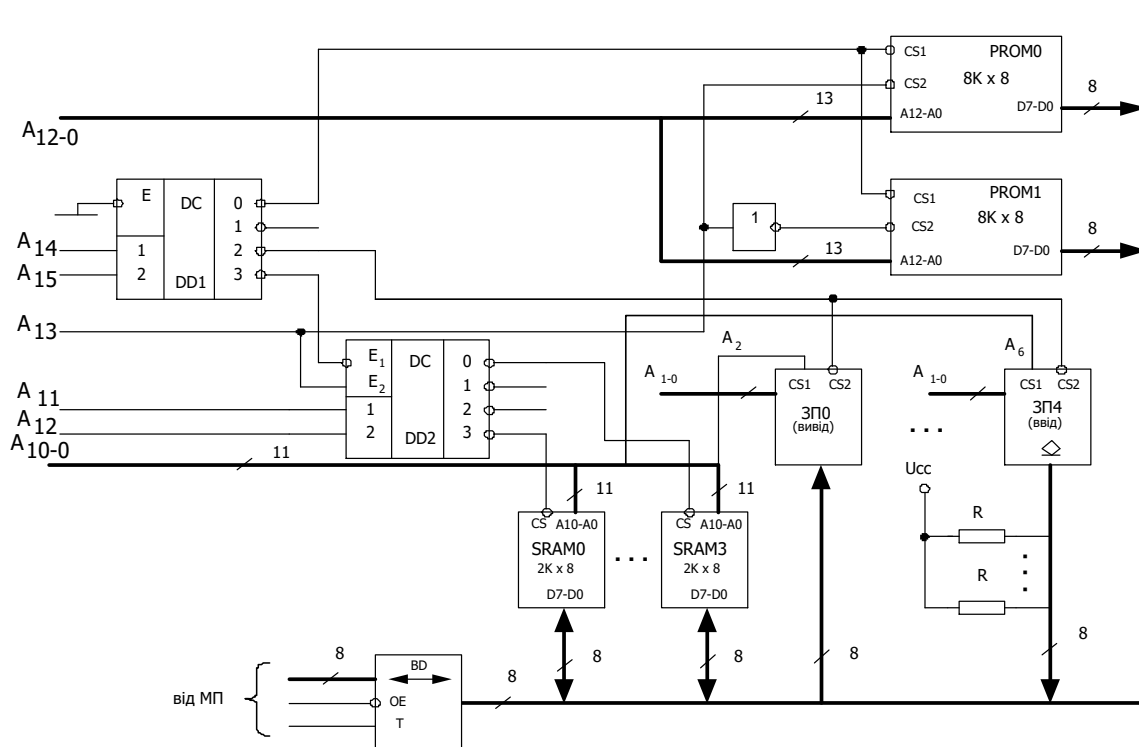


Рисунок 6 – Схема адресації модулів пам'яті і зовнішніх пристроїв.

Таблиця 1 – Схема адресації

Об'єкт	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
PROM0	0	0	0	d	d	d	d	d	d	d	d	d	d	d	d	d
PROM1	0	0	1	d	d	d	d	d	d	d	d	d	d	d	d	d
ЗП 0	1	0	x	x	x	x	x	x	x	0	0	0	0	1	d	d
ЗП 1	1	0	x	x	x	x	x	x	x	0	0	0	1	0	d	d
ЗП 2	1	0	x	x	x	x	x	x	x	0	0	1	0	0	d	d
ЗП 3	1	0	x	x	x	x	x	x	x	0	1	0	0	0	d	d
ЗП 4	1	0	x	x	x	x	x	x	x	1	0	0	0	0	d	d
SRAM0	1	1	1	0	0	d	d	d	d	d	d	d	d	d	d	d
SRAM1	1	1	1	0	1	d	d	d	d	d	d	d	d	d	d	d
SRAM2	1	1	1	1	0	d	d	d	d	d	d	d	d	d	d	d
SRAM3	1	1	1	1	1	d	d	d	d	d	d	d	d	d	d	d

Символом "X" позначені байдужні стани адресних розрядів, буквою d - розряди, що входять до складу адресних входів самих мікросхем пам'яті або адресних ліній внутрішніх регістрів ЗП.

Приклад 6

З метою спрощення схем декодування і при наявності "зайвого"

адресного простору можна застосувати неабсолютну адресацію, при якій кожному об'єкту адресації привласнюється не одна єдина адреса, а група адрес (деяка зона АП). Наприклад, в АП ємністю 64К потрібно розмістити всього два субмодуля пам'яті по 8К адрес у кожному. При абсолютній адресації (рисунок 7, а) на адресні входи самих ІС пам'яті надходять 13 молодших розрядів адреси для адресації 8К комірок субмодуля. Розряди, що залишилися, A_{15-13} надходять на дешифратор, нульовий і одиничний виходи якого дозволяють роботу субмодулей СМ1 і СМ2. Інші виходи дешифратора можуть бути використані для підключення інших об'єктів адресації в зоні АП, що залишилася вільної (48К).

При неабсолютній адресації лінії адреси $A_{12}-A_0$ подаються на адресні входи ІС, а для вибору однієї з них використовується лінія A_{15} . Лінії A_{14} і A_{13} не використовуються взагалі, їхні стани байдужні. Схема адресації (рисунок 7, б) спрощується, замість дешифратора "3 на 8" потрібний тільки інвертор. Платою за це є заняття двома субмодулями по 8К усього АП. Дійсно, всі адреси виду $0XXdd\dots d$ належать субмодулю СМ1, а це відповідає верхнім 32К АП. Всі адреси виду $1XXdd\dots d$ належать субмодулю СМ2 і займають 32К в нижньої частини АП.

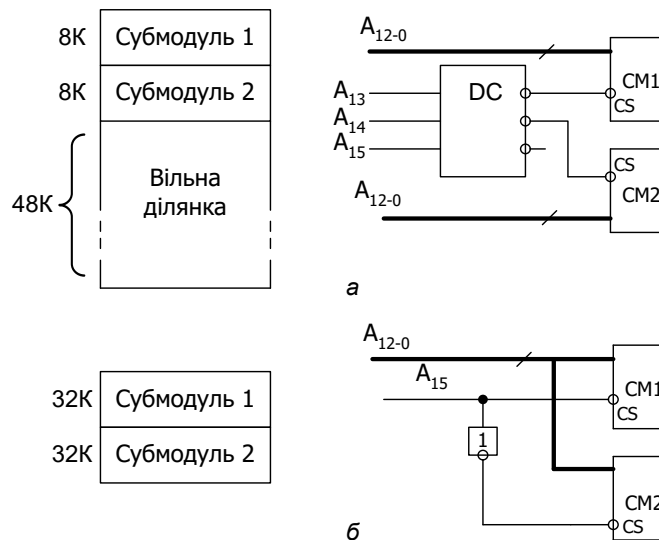


Рисунок 7 – Приклади реалізації абсолютної (а) і неабсолютної (б) адресації субмодулей пам'яті.

Неабсолютна адресація – досить гнучкий підхід для побудови схем декодування адреси. Для адресації об'єкта можна використовувати більш широкую зону АП, вибираючи при необхідності компроміс між крайніми рішеннями, показаними на рисунку 7.

Приклад 7

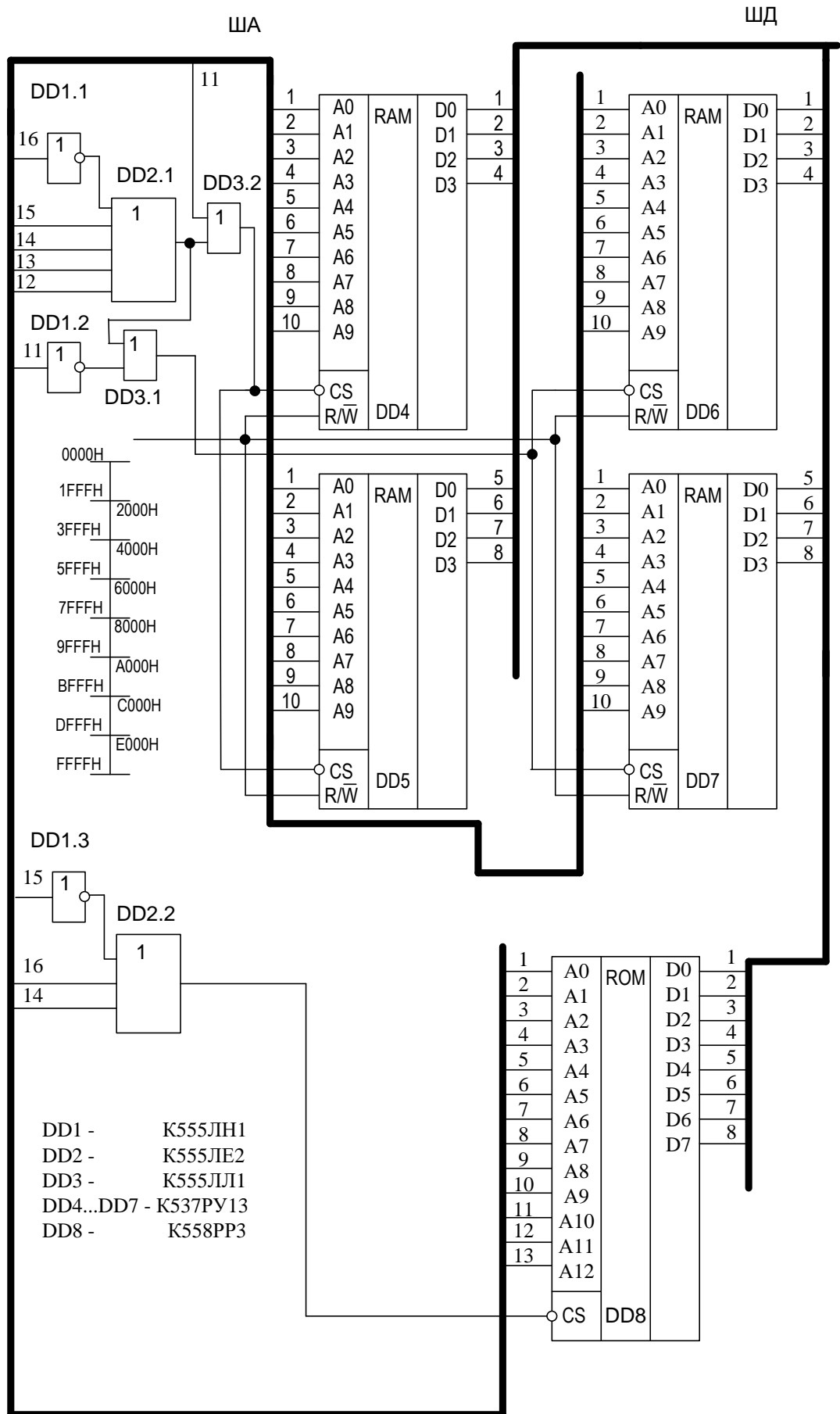


Рисунок 8 – Схема модулів пам'яті ОЗП та ПЗП мікропроцесорної системи

Для схеми:

1. Наведіть організацію ІМС ОЗП
2. Наведіть організацію ІМС ПЗП
3. Визначте об'єм ОЗП МПС.
4. Визначте об'єм ПЗП МПС.
5. Намалюйте адресний простір МПС та покажіть де розташовані субмодулі ОЗП та ПЗП.
6. Визначте, який діапазон шістнадцятирічних адрес можуть активізувати модулі ОЗП.
7. Визначте, який діапазон шістнадцятирічних адрес можуть активізувати модулі ПЗП.
8. Визначте, чи можуть шістнадцятирічні адреси 607F, 57FA, 5F00, 15D0, AFF3 активізувати модулі пам'яті, і якщо так, то які з них.
9. Наведіть схему керування субмодулями ОЗП, якщо початковий об'єм ОЗП збільшений у два рази.
10. Наведіть схему керування субмодулями ПЗП, якщо початковий об'єм ПЗП збільшений у два рази.

Відповідь:

1. Організація ІМС ОЗП – 1КБіт x 4
2. Організація ІМС ПЗП – 8КБіт x 8
3. ОЗП МПС має об'єм 2 КБайт
4. ОЗП МПС має об'єм 8 КБайт
5. Адресний простір МПС наведений на рисунку 9.
6. Для активізації модулів ОЗП на старших розрядах шини адреси повинні бути встановлені такі рівні сигналів 1000 0xxx xxxx xxxx, що відповідає діапазону адрес: 8000h-87Ffh.
7. Для активізації модулів ПЗП на старших розрядах шини адреси повинні бути встановлені такі рівні сигналів 010x xxxx xxxx xxxx, що відповідає діапазону адрес: 4000h-5FFFh.
8. Шістнадцятирічні адреси 57FA, 5F00 можуть активізувати модуль ПЗП.
9. Якщо початковий об'єм ОЗП збільшити у два рази, то для керування модулями потрібно застосувати дешифратор 2 на 4, схема застосування якого наведена на рисунку 10.
10. Якщо початковий об'єм ПЗП збільшити у два рази, то для керування модулями потрібно застосувати дешифратор 2 на 4, схема застосування якого наведена на рисунку 11 і задіяти додатково третій адресний простір МПС

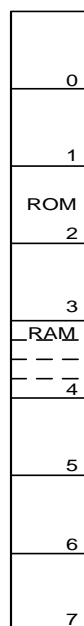


Рисунок 9 – Розподіл адресного простору (до рисунку 8)

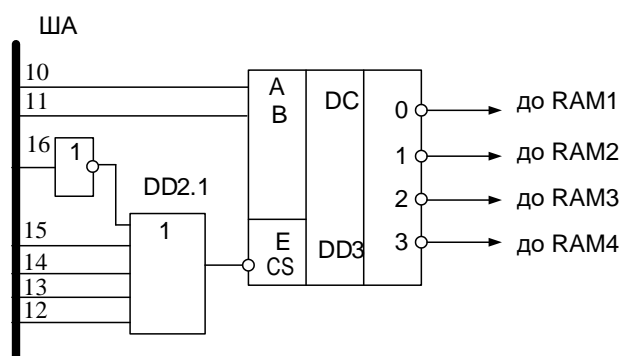


Рисунок 10 – Схема керування модулями ОЗП

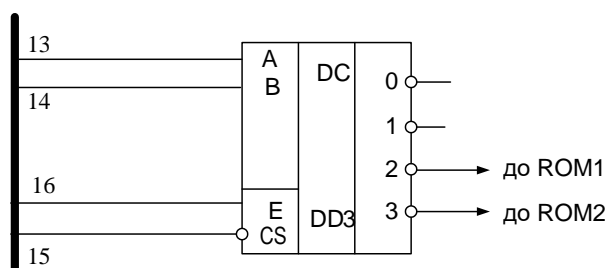


Рисунок 11 – Схема керування модулями ПЗП

Приклад 8

Розробити блок пам'яті, що підключається до шин мікропроцесорної системи (шина адреси – 16розрядна, шина даних – 8 розрядна), розробити функціональну схему, навести опис роботи:

Область ОЗП	Організація ОЗП	Об'єм ОЗП
4-7 (32-64КБайт)	8Кx8	32КБайт

Відповідь: блок пам'яті, що проектується, повинний підключатись до 16 розрядної шини адреси, 8 розрядної шини даних мікропроцесорної системи. Крім того мікросхема ОЗП має три режими роботи: читання, запис, зберігання. Для забезпечення приведених режимів роботи потрібно задіяти сигнал MR/\overline{MW} шини керування. Організація ІМС ОЗП 8Кх8 означає, що мікросхема має 8 розрядну шину даних та 13 адресних виводів ($2^{13}=8К$). Таким чином для організації об'єму 32кБ потрібно застосувати «нарощування по вертикалі» й використати 4 ІМС ОЗП ($32/8 = 4$). Для керування їх роботою потрібно застосувати адресний дешифратор 2 на 4.

Вибір області пам'яті 32-64 кБайт забезпечує самий старший розряд шини адреси A_{15} . Коли його стан 1 включається дешифратор DD1 і за допомогою $A_{14, 13}$ здійснюється вибір однієї з чотирьох ІМС ОЗП (DD2-DD5).

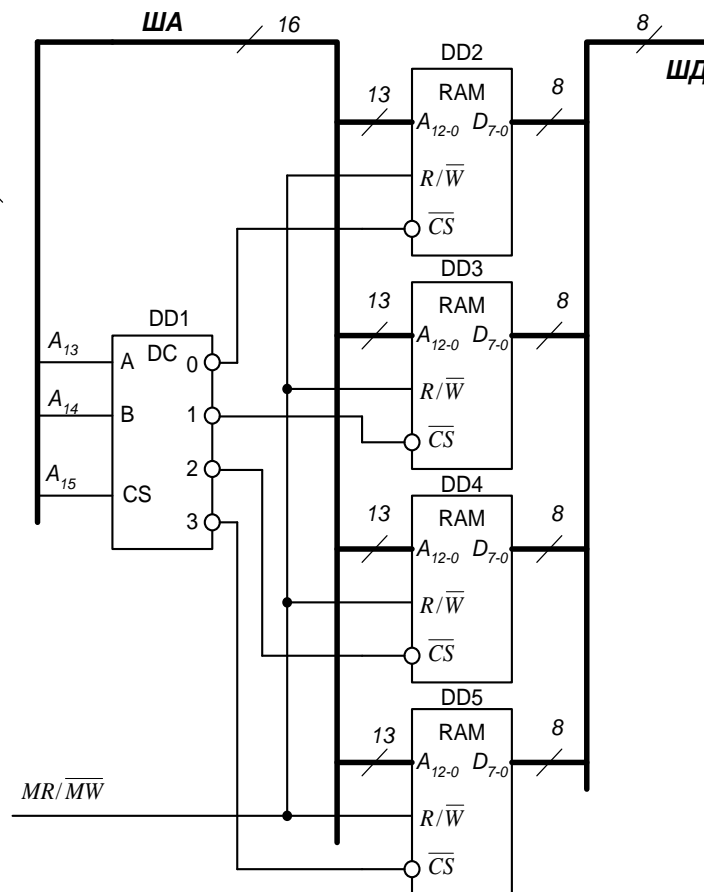


Рисунок 12 – Організація блоку пам'яті ОЗП мікропроцесорної системи

Варіанти індивідуальних завдань

Спроекувати блок пам'яті, розробити функціональну схему підключення до шин мікропроцесорної системи (шина адреси – 16-розрядна, шина даних – 8 розрядна), навести опис роботи (таблиця 2).

Таблиця 2 – Варіанти індивідуальних завдань

№	Область ОЗП	Організація ОЗП	Об'єм ОЗП
1	0 (0-4Кбайт)	1Кх4	4КБайт
2	0 (0-8КБайт)	1Кх8	8КБайт
3	7 (60-64КБайт)	2Кх4	4КБайт
4	7 (56-64КБайт)	2кХ8	8КБайт
5	4 (32-36КБайт)	4Кх1	4КБайт
6	4, 5 (32-48КБайт)	4Кх8	16КБайт
7	7 (56-64КБайт)	8Кх1	8КБайт
8	0-3 (0-32КБайт)	8Кх8	32КБайт
9	4 (32-36КБайт)	1Кх4	4КБайт
10	1 (8-12КБайт)	1Кх8	4КБайт
11	0 (0-4Кбайт)	2Кх4	4КБайт
12	0, 1 (0-16КБайт)	2кХ8	16КБайт
13	0 (0-4КБайт)	4Кх1	4КБайт
14	4-7 (32-64КБайт)	4Кх8	32КБайт
15	0 (0-8КБайт)	2Кх8	8КБайт
16	6 (56-60КБайт)	2Кх4	4КБайт
17	5 (36-40КБайт)	4Кх1	4КБайт
18	2,3 (16-32КБайт)	8Кх1	16КБайт
19	4 (32-40КБайт)	4Кх1	8КБайт
20	4-7 (32-64КБайт)	16Кх1	32КБайт

2 Способи розширення адресного простору мікропроцесора

Досить часто можливості адресного простору мікропроцесорної системи вичерпуються (сумарний об'єм доступних мікросхем ОЗП і ПЗП перевищує об'єм адресного простору). У такому разі доводиться прибгати до методів розширення адресного простору.

2.1 Метод сторінкового розширення адресного простору

Для того, щоб зрозуміти, як можна збільшити адресний простір у вже існуючій мікропроцесорній системі, не міняючи мікропроцесор, звернемо увагу на те, що сигнали на адресних провідниках нічим не відрізняються від сигналів, що використовуються для управління зовнішніми пристроями, тому для розширення адресного простору мікропроцесорної системи можна скористатися схемою, подібною до паралельного порту. Виходи регістра в

цьому випадку 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.

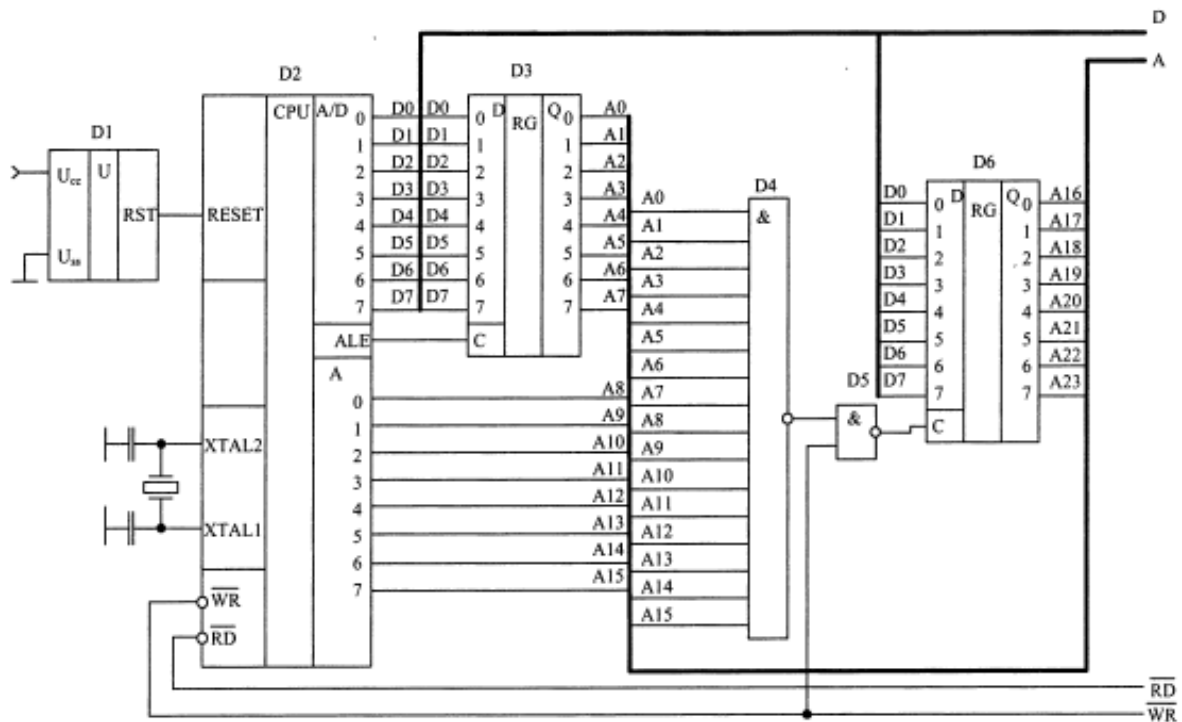


Рисунок 13 – Схема центрального процесора з перемикачем сторінок пам'яті, реалізованим за схемою паралельного порту

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. Паралельний порт, при його застосуванні для розширення адресного простору мікропроцесорної системи, називається диспетчером пам'яті, а його регістр даних — перемикачем сторінок. Схема центрального процесора з диспетчером пам'яті, реалізованим за схемою паралельного порту, приведена на рисунку 13. На цій схемі для економії зовнішніх виводів шина даних і шина адреси 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. Подібний принцип застосовується в

мікропроцесорах K580, K1821, K1816, K1830 і ряду зарубіжних мікропроцесорів, наприклад, сімействі мікроконтролерів AVR. В результаті застосування паралельного регістра D3 формується 16-розрядна шина адреси. Як регістр даних перемикача сторінок в схемі на рисунку 13 використаний паралельний регістр D6. Для звернення до цього регістра використовується дешифратор адреси, зібраний на мікросхемі D4. Дешифратор адреси налаштований на число 0FFFFh, що дозволяє звертатися до перемикача сторінок на кожній сторінці як до самого старшого елемента пам'яті сторінки, а це означає, що втрачається тільки один осередок адресного простору кожної сторінки.

При застосуванні восьмирозрядного перемикача сторінок в мікропроцесорній системі з'явилися додаткові вісім ліній адреси. В результаті адресний простір мікропроцесорної системи збільшився до $2^{24} = 16$ Мбайт. Структура нового адресного простору наведена на рисунку 14, а принцип формування нової адреси з використанням перемикача сторінок пояснений на рисунку 15.

Повна адреса	Адреса сторінки всередині	Адреса комірки всередині
FFFFFF	FFFF	1111111111111111
	Сторінка 255	
FF0000	0000	0000000000000000
	...	
02FFFF	FFFF	1111111111111111
	Сторінка 2	
020000	0000	0000000000000000
01FFFF	FFFF	1111111111111111
	Сторінка 1	
010000	0000	0000000000000000
00FFFF	FFFF	1111111111111111
	Сторінка 0	
000000	0000	0000000000000000

Рисунок 14 – Структура сторінкового адресного простору

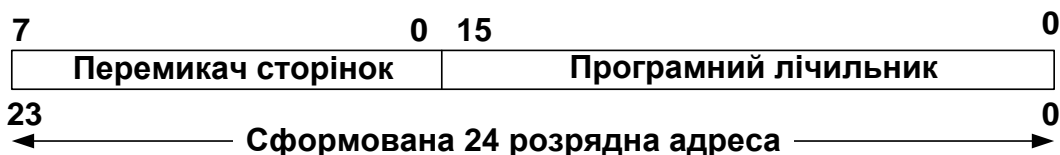


Рисунок 15 – Формування адреси з використанням перемикача сторінок

2.2 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.

При використанні цього методу 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. Правда 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 рисунку 16. 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. Такий метод розширення адресного простору дозволяє економніше використати доступні ресурси пам'яті мікропроцесора.

Для формування фізичної адреси, при сегментному виді розширення адресного простору мікропроцесорної системи, 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.

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

Рисунок 16 – Приклад адресного простору з використанням трьох програмних сегментів

Розглянуті принципи формування адреси при сегментній адресації відображені рисунком 17. Замість програмного лічильника можуть виступати регістри покажчика даних і покажчика стека.

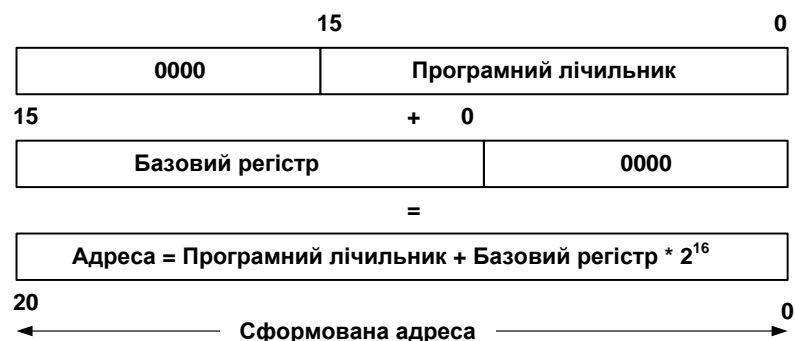


Рисунок 17 – Формування адреси при сегментній адресації

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.

Кількість одночасно сегментів, що використовуються, визначається кількістю базових реєстрів. Сегменти можуть перекриватися в адресному просторі, і тим самим можна регулювати розмір пам'яті, який відводиться під кожен конкретний сегмент пам'яті. У комп'ютерах сімейства IBM PC 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.

2.3 Метод розширення адресного простору за допомогою вікон

Ще одним поширеним способом збільшення адресного простору мікропроцесорної системи є застосування вікон. При використанні вікон робиться розширення не усього адресного простору, а тільки його частини. У середині 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 (див. схему на рисунку 13). 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.

Принцип використання віконної адресації при відображенні сторінок додаткової пам'яті в основний адресний простір можна легко зрозуміти по рисунку 18.

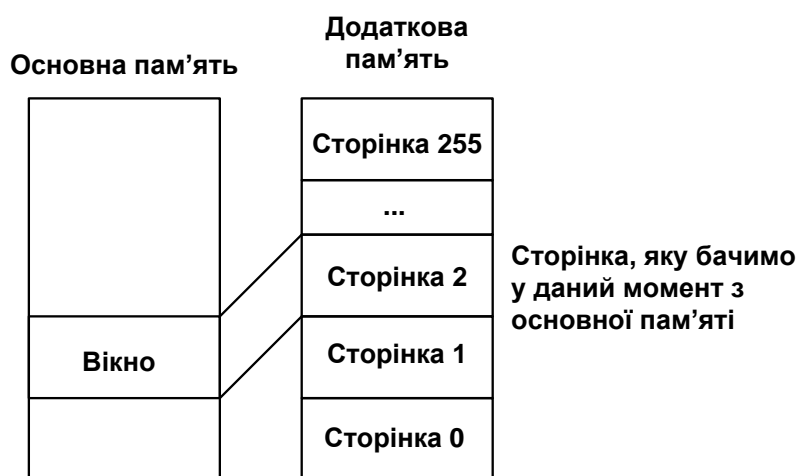


Рисунок 18 – Застосування вікна для розширення адресного простору

Віконна адресація часто використовується при розвитку мікропроцесорних сімейств, 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. Як приклад можна привести мікросхеми 181С96 фірми INTEL або TM8320С5410 фірми Texas Instrument, де для розширення області регістрів спеціальних функції використовується віконна адресація.