

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

МЕТОДИЧНІ ВКАЗІВКИ

**і завдання до лабораторного практикуму з курсу "Проектування
запам'ятовуючих пристроїв" для студентів спеціальностей "Комп'ютерні системи і
мережі" і "Системне програмування" денної і заочної форм навчання**

Донецьк ДНТУ 2008

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

МЕТОДИЧНІ ВКАЗІВКИ

і завдання до лабораторного практикуму з курсу "Проектування запам'ятовуючих пристроїв" для студентів спеціальностей "Комп'ютерні системи і мережі" і "Системне програмування" денної і заочної форм навчання

Затверджено
на засіданні кафедри
Електронних обчислювальних
машин.
Протокол № 1 від
31. 08. 2007 р.

Затверджено
на засіданні учбово-видавничої
ради ДНТУ.
Протокол № 1 від 25. 10. 2007 р.

Донецьк ДНТУ 2008

УДК 681.3

Методичні вказівки і завдання до лабораторного практикуму з курсу "Проектування запам'ятовуючих пристроїв" для підготовки фахівців і магістрів на базі напрямку "Комп'ютерна інженерія" / Укл. Ю.В. Губарь. – Донецьк: Видавництво ДНТУ, 2008. - 82 с.

Збірник містить методичні вказівки з вісьми лабораторних робіт. У збірник увійшли лабораторні роботи з проектування, аналізу і симуляції оперативних запам'ятовуючих пристроїв статичного та динамічного типів. Розглянуті питання функціонального контролю блоків ОЗП, регенерації інформації динамічних пристроїв, алгоритми синхронізації буферних запам'ятовуючих пристроїв та блоків кеш - пам'яті.

Укладач

доц. Губарь Ю.В.

Відповідальний
за випуск

проф. Святний В.А.

Рецензент

проф. Аверін Г. В.

Автор висловлює подяку студенту Маслову Е. Ю. (гр.КС-04а) за сприяння у виданні методичних вказівок до лабораторного практикуму.

ПЕРЕЛІК ОСНОВНИХ СКОРОЧЕНЬ

ОЗП – оперативний запам'ятовуючий пристрій.

ПЗП – постійний запам'ятовуючий пристрій.

МС – мікросхема пам'яті.

НК – накопичувач.

РА – регістр адреси.

DC DM – дешифратор вибірки мікросхем пам'яті.

СУ – схема узгодження.

РШ – розрядна шина.

ЕП – елемент пам'яті.

АФ – адресний формувач.

А – адреса.

$\overline{RAS}$ – строб адреси рядка.

$\overline{CAS}$ – строб адреси стовбця.

REF – сигнал регенерації пам'яті.

DC X – дешифратор рядків.

DC Y – дешифратор стовпців.

N_M – кількість чисел (адресів) в мікросхемі пам'яті.

N – кількість чисел (адресів) в модулі ЗП..

n_m – кількість розрядів в мікросхемі пам'яті.

n – кількість розрядів в модулі ЗП.



– вихід мікросхеми з трьома станами.



– вихід мікросхеми з відкритим колектором.



– вихід мікросхеми з відкритим емітером.

Лабораторна робота №1

РОЗРОБКА ТЕСТУ ФУНКЦІОНАЛЬНОГО КОНТРОЛЮ МОДУЛЯ ОПЕРАТИВНОГО ЗАПАМ'ЯТОВУЮЧОГО ПРИСТРОЮ

МЕТА РОБОТИ: опанувати принципи функціонування тесту контролю ОЗП; придбати практичні навички реалізації прикладів роботи заданого тесту; навчитися розробляти і виконувати аналіз отриманих результатів.

ЗАГАЛЬНІ ПОЛОЖЕННЯ

Справний стан модуля ОЗП визначається шляхом контролю: статичних параметрів (вхідних і вихідних струмів і напруг, струмів споживання в режимі збереження та у режимі звертання); динамічних параметрів (часу фронту і спаду сигналу, тривалості сигналу, часу збереження, утримання і відновлення); функціонування (функціональний контроль) [1].

Функціональний контроль (ФК) вирішує дві основні задачі: визначення факту наявності несправності в модулі пам'яті (задача контролю) і визначення місця несправності (задача діагностики).

Методи ФК засновані на порівнянні з еталонними сигналами вихідних реакцій сигналів тестуємої схеми пам'яті на задані вхідні впливи. У поняття тесту включають склад, параметри і порядок проходження електричних сигналів, подаваних на контрольну схему з метою виміру і контролю її працездатності [1]. Коди еталонних сигналів повинні відповідати вихідним кодам справного модуля пам'яті при заданих вхідних впливах. Еквівалентність вихідних і еталонних сигналів визначається шляхом логічного порівняння функціонування ВІС.

Ефективність ФК вирішальним образом визначається побудовою теста. Найбільше широко використовуються при контролі функціонування ЗП алгоритмічні функціональні тести, які містять послідовності елементарних тестів, змінюваним по відомому алгоритму. Це зв'язано з простотою генерації тесту, малим обсягом займаної пам'яті керуючої ЕОМ і великим результатом відтворюваності результатів ФК.

Безпосередній перебір усіх можливих станів ОЗП практично реалізувати неможливо. Тому алгоритми ФК ЗП мають обмежений набір вхідних тестових комбінацій (циклів звертання), які забезпечують виявлення типових відмовлень.

По кількості циклів звертання до контролюючої схеми алгоритми ФК умовно поділяються на три типи [1, 2]: N , N^2 і $N^{3/2}$ (N - ємність ЗП, біт). Лінійні алгоритми типу N використовуються звичайно для попередньої оцінки ОЗП на відсутність катастрофічних несправностей. Квадратичні алгоритми типу N^2 є ефективними для контролю функціонування всіх типів ЗП. Застосування цих алгоритмів обмежується різким ростом тривалості контролю зі збільшенням ємності ЗП. Алгоритми типу $N^{3/2}$ є компромісом між тривалістю і вірогідністю контролю ВІС пам'яті. Вони широко використовуються для контролю ОЗП великої ємності.

1.1. Послідовність виконання лабораторної роботи

1. Вивчити з використанням літературних джерел [1 - 3] і матеріалів додатку методичних вказівок алгоритм функціонування заданого тесту і перелік помилок, які даний тест здатний виявити.
2. Розробити блок - схему програми, яка реалізує алгоритм заданого тесту.

3. Розробити і налагодити програму заданого тесту.
4. Розрахувати теоретично час виконання тесту і порівняти його з експериментальним значенням.
5. Скласти звіт про виконану лабораторну роботу і захистити його.
6. Дати відповіді на контрольні запитання.

1.2. Варіанти індивідуальних завдань

Варіанти індивідуальних завдань наведено у табл. 1.1 і табл. 1.2, де N - номер завдання.

Таблиця 1.1

Варіант =(N) _{m9}	Найменування тесту	Тип тесту
0	Галоп	N ²
1	Марш	N
2	Попарне запис-зчитування з повним перебором	N ²
3	Попарне зчитування по стовбцю	N ^{3/2}
4	Галопуючий адресний код	N ^{3/2}
5	Батерфляй	N ^{3/2}
6	Хрест	N
7	Зсув діагоналі	N ^{3/2}
8	Пінг - понг	N ²

Таблиця 1.2

Варіант = (N) _{m7}	Організація модуля ОЗП
0	512 x 8
1	1K x 8
2	2K x 16
3	256 x 8
4	64 x 4
5	128 x 8
6	512 x 8

1.3. Методичні вказівки

Припустимо, що в системі є два модулі ОЗП: еталонний і який перевіряється. Обидва модулі підключені через системний інтерфейс до мікропроцесора K1810BM86 (рис.1.1). Потрібно розробити і налагодити за допомогою персональної ЕОМ програму заданого тесту на мові Асемблера.

Розробка блок-схеми програми тесту "Шаховий код"

Спрощена блок-схема програми тесту "Шаховий код" наведена на рис.1.2. На блок-схемі показані перші два цикли роботи з модулями ОЗП і використовуються наступні позначки:

N – кількість комірок в модулі ОЗП;

MEMR³[I] – зміст I - і комірки еталонного модулю ОЗП;

MEMR^{II}[I] – зміст I - і комірки перевіряемого модулю ОЗП.

Лістинг програми тесту "Шаховий код"

TITLE TEST

```
STACK      SEGMENT
            DW 256 DUP      (?)
            TOP LABEL      WORD
STACK      ENDS
```

```
DATA      SEGMENT
MEMTEST    DB 256 DUP(?)
MEMORIG     DB 256 DUP(?)
ERROR      DB 'В ОЗП ЗВІЙНА КОМІРКА, ODH, 0AH, $'
OKEJ       DB 'ТЕСТ ОЗП ПРОЙШОВ, ODH, 0AH, $'
TIME_BEGIN DB 'ТЕСТ ПОЧАВСЯ В:', '$'
TIME_END   DB 'ТЕСТ ЗАКІНЧИВСЯ:', '$'
ERRMESG    DB 'ЗВІЙНА КОМІРКА:'
ENTRMESG   DB '$'
T          DB ' : : : '$'
BUF        DW 0
TEN        DB 10
DATA      ENDS
```

```
CODE      SEGMENT
PROG      PROC      FAR
ASSUME CS : CODE, DS : DATA, SS : STACK
NCON      EQU 255    ; Лічильник
PUSH      DS         ;
MOV       AX, 0       ;
PUSH      AX         ;
MOV       AX, DATA   ;
MOV       DS, AX      ;
            ;
MOV       AX, STACK   ;
MOV       SS, AX      ;
MOV       SP, OFFSET TOP ;
            ;
LEA DX, TIME_BEGIN   ; Вивід
INT 21H              ; початкового
CALL TIME_PROC       ; часу роботи тесту

ORG = 4010H          ; Початкова адреса програми
MOV      CX, 255     ; Завантаження лічильника
```


LEA DX,TIME_END	;	
INT 21H	;	
CALL TIME_PROC	;	Вивід кінцевого часу
MOV AX,0	;	роботи тесту
INT 16H	;	
RET	;	Повернення керування
	;	налагоджувачу DEBUG

PROG	ENDP	
TIME_PROC	PROC NEAR	
	MOV SI, 11	
	MOV AH, 2CH	
	INT 21H	
	MOV BUF, DX	
	LEA DX, T	
	MOV BX, DX	
	MOV DX, BUF	
LABEL6:	MOV AX, 0	
	MOV AL, DL	
	DIV TEN	
	ADD AX, 12336	
	MOV [BX+SI], AH	
	DEC SI	
	MOV [BX+SI], AL	
	SUB SI, 2	
	MOV DL, DH	
	JNP LABEL6	
	MOV AX, 0	
	MOV AL, CL	
	DIV TEN	
	ADD AX, 12336	
	MOV [BX + SI], AH	
	DEC SI	
	MOV [BX + SI], AL	
	SUB SI, 2	
	MOV AX, 0	
	MOV AL, CH	
	DIV TEN	
	ADD AX, 12336	
	MOV [BX + SI], AH	
	DEC SI	
	MOV [BX + SI], AL	
	LEA DX, T	
	MOV AH, 9	
	INT 21H	
	RET	
TIME_PROC	ENDP	
CODE	ENDS	
	END	PROG

Процедура зчитування
та вивіду часу роботи тесту

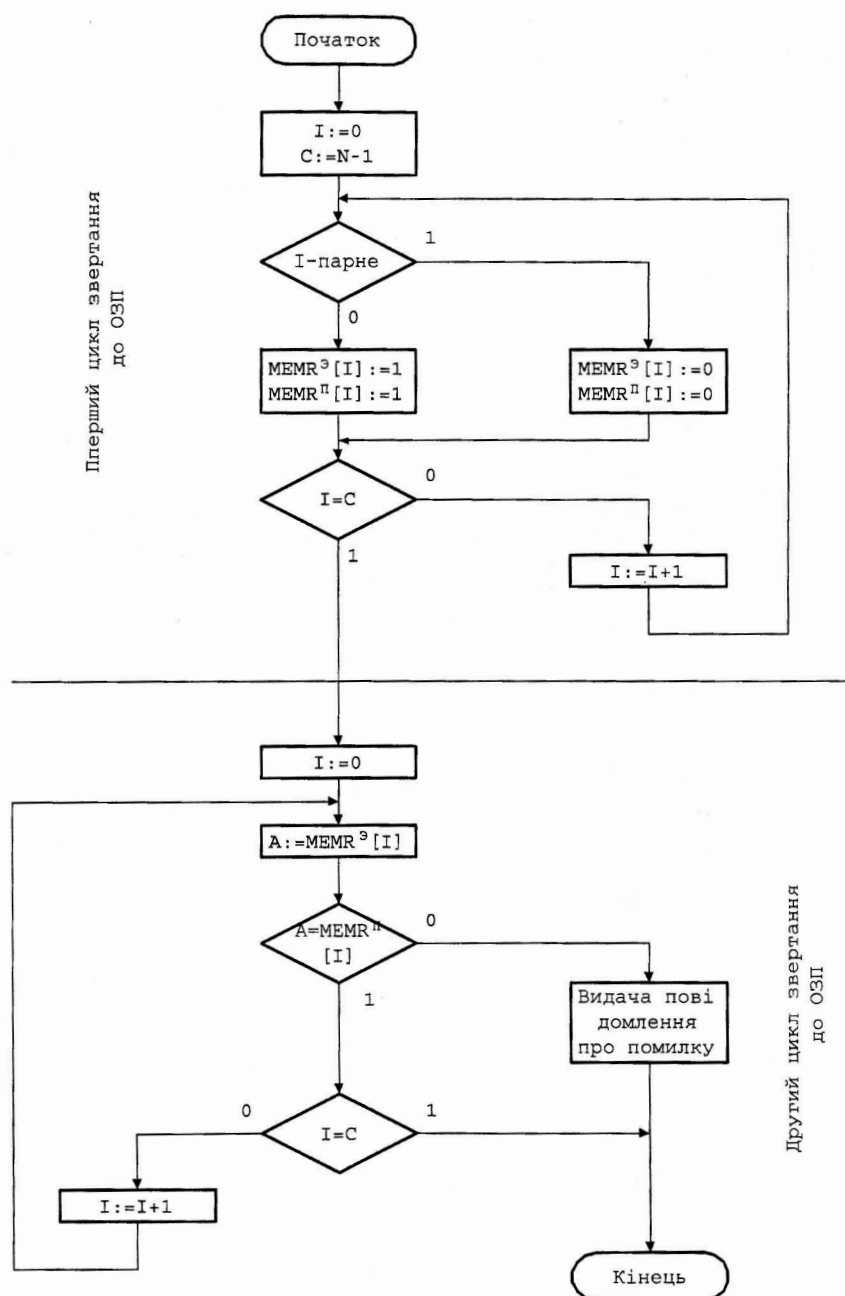


Рис 1.2 Спрощена блок - схема програми "Шаховий код"

Розрахунок часу роботи тесту

Теоритично розрахунок часу роботи тесту виконується з використанням значень часу виконання кожної команди мікропроцесором K1810BM86 [3]. У розглядаєму випадку тест "Шаховий код" виконується за 0,015с. Експериментально розрахунок часу роботи теста виконується з використанням процедури Time PROC. Викликом цієї процедури перед запуском тестової програми визначається стартовий час. По закінченні роботи тесту ця процедура викликається ще раз. При цьому визначається кінцевий час роботи тесту. Різниця між стартовим і кінцевим значеннями відповідає часу роботи тесту.

1.4. Зміст звіту

1. Словесний опис тесту. Указати перелік помилок, які цей тест дозволяє виявити.
2. Приклади, які пояснюють алгоритм функціонування тесту.
3. Блок-схема програми реалізації тесту з коментарями.
4. Лістинг програми на мові Асемблера.
5. Розрахунок часу роботи тесту.
6. Список використаних джерел.

1.5. Контрольні запитання

1. Обґрунтуйте тип помилок, які дозволяє відокремити тест, який був заданий в індивідуальному завданні.
2. Яким чином здійснюється імітація помилок у тестуємому модулі ОЗП?
3. Запропонуйте алгоритм, який дозволяє не тільки виявляти, але і виправляти одиночну (подвійну) помилку.
4. У яких випадках рекомендується використовувати лінійні алгоритми тестів ФК типу N ?
5. У яких випадках рекомендується використовувати квадратичні алгоритми тестів ФК типу N^2 ?
6. У яких випадках рекомендується використовувати алгоритми тестів ФК типу $N^{3/2}$?

Література

1. Полупроводниковые БИС запоминающих устройств. Справочник/ В.В. Баранов, Н.В. Бекин, А.Ю. Гордонов и др., Под. ред. А.Ю. Гордонова и Ю.Н. Дьякова. - М.: Радио и связь. - 1987. - 360 с.
2. Огнев И.В., Сарычев К.Ф. Надежность запоминающих устройств. - М: Радио и связь. - 1988. - 224 с.
3. Скэнлон Л. Персональные ЭВМ IBM PC и XT. Программирование на языке ассемблера: пер. с англ. - М.: Радио и связь. - 1989. - 336 с.
4. Алексеев А. Г., Галицын А. А., Иванников А. Д. Проектирование радиоэлектронной аппаратуры на микропроцессорах. - М.: Радио и связь, 1984. - 272 с.
5. Самофалов К. Г., Викторов О. В. Микропроцессоры. - Библиотека инженера - К.: Техника, 1989. - 312 с.
6. Абель П. Язык ассемблера для IBM PC и программирования. - М.: Высшая школа, 1992.
7. Зубков С. В. Assembler для DOS, Windows и Unix. - М.: ДМК Пресс, 2000. - 608 с.
8. Юров В. Assembler. - СПб.: Питер, 2001. - 624 с.

Лабораторна робота №2

ПРОЕКТУВАННЯ МОДУЛЯ ОЗП СТАТИЧНОГО ТИПУ

МЕТА РОБОТИ: опанувати принципи функціонування ОЗП статичного типу; придбати практичні навички розраховувати параметри модуля на базі заданого типу мікросхеми пам'яті; навчитися розробляти часові діаграми роботи модуля та виконувати аналіз отриманих результатів.

ЗАГАЛЬНІ ПОЛОЖЕННЯ

Основними факторами, які визначають структуру побудови модуля пам'яті, є вхідні і вихідні навантажувальні характеристики ВІС ЗП і схем узгодження, а також часові характеристики [1 - 4].

Навантаження на схеми керування модулем пам'яті, який побудований на елементах біполярних ВІС, визначається вхідними струмами логічних 0 і 1 та вхідними ємністями. Якщо нагромаджувач виконано на ВІС ЗП на основі МОП-структур, то вихідними струмами (вони визначаються струмами витоку) можливо зневажити.

Вихідні каскади ВІС ЗП дозволяють поєднувати їх за схемою провідного АБО, якщо ВІС ЗП виконано з відкритим колектором або трьома станами. При цьому об'єднанні повинні дотримуватися вимоги технічних умов на ВІС з вихідних навантажень. ВІС ЗП на МОП-структурах, як правило, будуються за схемою з трьома станами і мають високу ємнісну навантажувальну здатність ($C_H > 100$ пф) і мале токове навантаження (один вихід ТТЛ-схеми).

Більшість вихідних каскадів ВІС ЗП забезпечують сумісність із ТТЛ-схемами. Якщо це не дотримується (наприклад, для ЗП на р-МОП або КМОП), то застосовують спеціальні схеми узгодження, або ТТЛ-схеми з відкритим колектором [2]. У додатку Д2 наведені схеми і відповідні розрахункові формули при різних варіантах організації вихідних каскадів ВІС ЗП.

2.1. Послідовність виконання лабораторної роботи

1. Вивчити з використанням літературних джерел [1 - 3] склад, режими роботи і часові діаграми функціонування заданої мікросхеми пам'яті.
2. Розробити функціональну схему модуля ОЗП заданої організації.
3. Виконати розрахунок вхідних і вихідних ланцюгів модуля.
4. Виконати розрахунок часових параметрів модуля пам'яті.
5. Виконати розрахунок споживаної потужності, кількості корпусів використовуваних мікросхем пам'яті та електронного обрамлення.

2.2. Варіанти індивідуальних завдань

Варіанти індивідуальних завдань наведено у табл. 2.1 - табл. 2.3, де N - номер завдання.

Таблиця 2.1

Варіант (N) _{m1}	Тип мікросхеми	Організація мікросхеми N _M × n _M	Технологія виготовлення	Література
0	KM185PY7	256 × 4	ТТЛ	3
1	KM185PY9	64 × 9	ТТЛ	3
2	K589PY1	16 × 4	ТТЛШ	3
3	K1500PY474	1024 × 4	ЕСЛ	3
4	KM132PY9A	1024 × 4	пМОП	3
5	K1809PY1Б	256 × 16	пМОП	3
6	K1809PY1A	512 × 16	пМОП	3
7	KP537PY11A	256 × 16	КМОП	3
8	KP537PY10	1024 × 8	КМОП	3
9	KP537PY13A	1024 × 4	КМОП	3
10	KM1603PY1	256 × 4	КМОП	3

Таблиця 2.2

Варіант=(N) _{m8}	Організація модуля ОЗП N × n
0	2K × 16
1	4K × 32
2	8K × 64
3	16K × 16
4	2K × 32
5	4K × 32
6	8K × 64
7	16K × 32

Таблиця 2.3

Варіант=(N) _{m4}	Серія елементів керування ОЗП	Література
0	KP134	6,7
1	K155	6,7
2	K531	6,7
3	K555	6,7

Примітка. Дані, які не приведені в довідковій літературі, але необхідні для розрахунку, обираються студентом самостійно за узгодженням з викладачем.

2.3. Методичні вказівки

Розглянемо приклад проектування модуля статичного ОЗП ємністю 1024 16 - розрядних слів на основі мікросхем пам'яті K185PY4 з організацією 256 чисел по 1 розряду кожне [1]. У якості схем обрамлення приймемо мікросхеми серії K133 [6].

2.3.1. Експлуатаційні характеристики і режими роботи мікросхеми K185PY4

Мікросхема пам'яті **K185PY4** виконана на біполярних тиристорах і має вихід типу відкритий колектор [1]. Основні параметри мікросхеми наступні:

- вхідний струм логічної 1 по усіх входах $I_{BX}^1 \leq 25 \text{ мкА}$;
- вхідний струм логічного 0 по адресних входах $I_{BXA}^0 \leq 45 \text{ мА}$;
- вхідний струм логічного 0 по входу ВМ $I_{BXBM}^0 \leq 0,9 \text{ мА}$;
- вхідний струм логічного 0 по інформаційному входу $I_{BXH}^0 \leq 45 \text{ мА}$;
- вхідна ємність $C_{BX} \leq 4 \text{ нФ}$;
- вхідна напруга логічного нуля $U_{BX}^0 \leq 0,4 \text{ В}$;
- вхідна напруга логічної 1 $U_{BX}^1 \geq 2,4 \text{ В}$;
- вихідна напруга логічного 0 $U_{BbX}^0 \leq 0,4 \text{ В}$;
- вихідна напруга логічної 1 $U_{BbX}^1 \geq 2,4 \text{ В}$ (мікросхема має вихід типу відкритий колектор);
- вихідний струм логічного 0 $I_{BbX}^0 \leq 5 \text{ мА}$;
- вихідний струм логічної 1 $I_{BbX}^1 \leq 0,2 \text{ мА}$;
- вихідний струм необраної схеми (струм витоку) $I_{YT} \leq 30 \text{ мкА}$;
- вихідна ємність $C_{BbX} \leq 4 \text{ нФ}$;
- ємність навантаження $C_H \leq 80 \text{ нФ}$;
- струм споживання в режимі збереження $I_{XP} \leq 30 \text{ мА}$;
- струм споживання в режимі звертання $I_{OBP} \leq 75 \text{ мА}$;
- час вибірки адреси $t_{B_A} \leq 200 \text{ мс}$;
- час вибірки відносно ВМ $t_{B_BM} \leq 200 \text{ мс}$;
- час зсуву сигналу ВМ щодо адреси $t_{C_BM_A} \geq 0 \text{ мс}$;
- час збереження адреси після сигналу ВМ $t_{CX_A_BM} \geq 0 \text{ мс}$;
- час збереження вихідної інформації після сигналу ВМ $t_{CX_H_BM} \leq 180 \text{ мс}$;
- тривалість сигналу запису $\tau_{ЗП} \geq 230 \text{ мс}$;
- тривалість сигналу ВМ $\tau_{BM} \geq 230 \text{ мс}$.

Функціональна схема мікросхеми **K185PY4** наведена на рис.2.1. Часова діаграма її роботи в режимах запису і зчитування – на рис.2.2. Мікросхема має два дешифратори **DCX** і **DCY** на 4 входи і 16 вихідів кожний, розрядні формувачі **F**, матрицю накопичувача 16 x 16 (виконана на тиристорних елементах), вихідний підсилювач з відкритим колектором і схеми керування, які забезпечують режими запису, зчитування і збереження інформації.

Особливістю роботи мікросхеми **K185PY4** є відсутність командних сигналів запису - зчитування інформації. Зчитування даних здійснюється коли на обох інформаційних входах **Vx_И0** і **Vx_И1** подані сигнали логічного нуля, запис - коли

СУ – схема узгодження; МС – мікросхема пам'яті; ДС ВМ – дешифратор вибірки мікросхем пам'яті; РА – регістр адреси; RD – регістр вихідних даних; А – адреса; ВМ – сигнал вибірки модуля; Зп / Зч – сигнал завдання режиму роботи (запис / читування); DI – входи даних; DO – виходи даних.

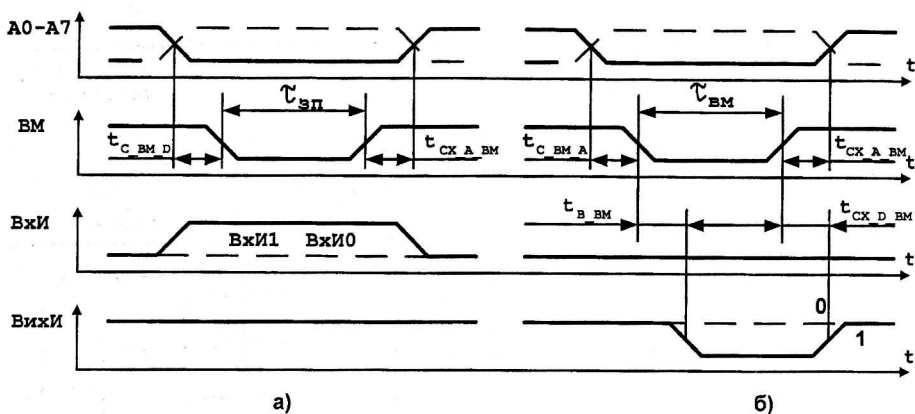


Рис. 2.2. Часова діаграма роботи ВІС К185РУ4:
а) – режим запису; б) – режим читування

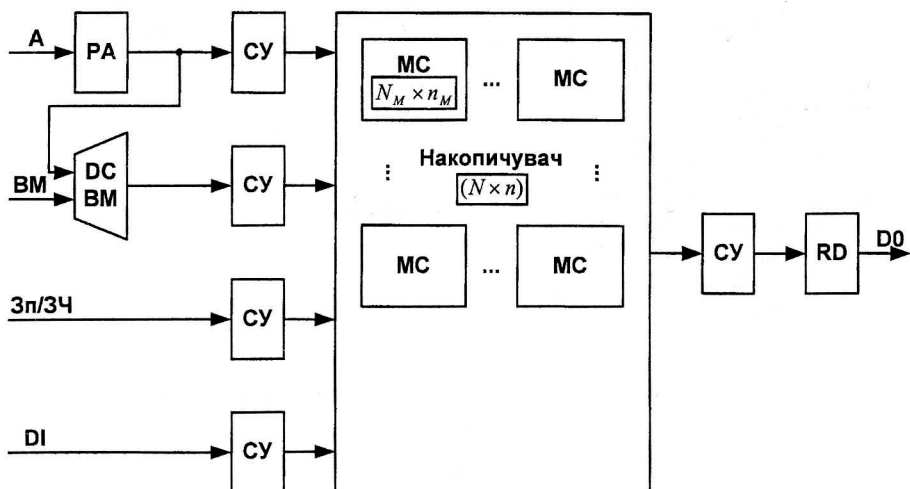


Рис. 2.3. Узагальнена структурна схема модуля ОЗП

Розрахунок модуля включає визначення наступних параметрів.

а) - кількість мікросхем у модулі:

$$Q = \frac{N}{N_M} \times \frac{n}{n_M} = \frac{1024}{256} \times \frac{16}{1} = 64.$$

б) - коефіцієнти об'єднання щодо вхідних ланцюгів накопичувача модуля:

$$\begin{aligned} K_A = Q &= ; \\ K_{BM} &= \frac{n}{n_M} = \frac{16}{1} = 16; \\ K_H &= \frac{N}{N_M} = \frac{1024}{256} = 4. \end{aligned}$$

в) - кількість вихідів дешифратора вибірки мікросхеми DC BM:

$$S_{BM} = \frac{N}{N_M} = \frac{1024}{256} = 4.$$

г) - значення навантаження щодо адресних ланцюгів:

$$\begin{aligned} I_A^0 &= K_A \cdot I_{BXA}^0 = 64 \cdot 0,45 = 28,8 \text{ мА}; \\ I_A^1 &= K_A \cdot I_{BXA}^1 = 64 \cdot 25 = 1600 \text{ мкА} = 1,6 \text{ мА}; \\ C_A &= K_A \cdot C_{BXA} + C_{MA} = 64 \cdot 4 + 15 = 271 \text{ пФ}, \end{aligned}$$

де I_A^0, I_A^1 - струми навантаження логічного "0" і "1" відповідного елемента згідності (вхідні струми накопичувача у одному адресному ланцюзі модуля);
 C_A - вхідна ємність одного вхідного адресного ланцюга накопичувача модуля;
 C_{BXA} - вхідна ємність одного адресного входу ВІС ОЗП;
 C_{MA} - монтажна ємність по одному адресному ланцюзі накопичувача модуля ОЗП ($C_M = 15$ пФ).

д) - значення навантаження щодо ланцюгів BM:

$$\begin{aligned} I_{BM}^0 &= K_{BM} \cdot I_{BXXM}^0 = 16 \cdot 0,9 = 14,4 \text{ мА}; \\ I_{BM}^1 &= K_{BM} \cdot I_{BXXM}^1 = 16 \cdot 25 = 400 \text{ мкА} = 0,4 \text{ мА}; \\ C_{BM} &= K_{BM} \cdot C_{BXXM} + C_{MBM} = 16 \cdot 4 + 15 = 79 \text{ пФ}; \end{aligned}$$

е) - значення навантаження щодо інформаційних ланцюгів:

$$\begin{aligned} I_H^0 &= K_H \cdot I_{BXXH}^0 = 4 \cdot 0,45 = 1,8 \text{ мА}; \\ I_H^1 &= K_H \cdot I_{BXXH}^1 = 4 \cdot 25 = 100 \text{ мкА} = 0,1 \text{ мА}; \\ C_H &= K_H \cdot C_{BXXH} + C_{MH} = 4 \cdot 4 + 15 = 31 \text{ пФ}. \end{aligned}$$

2.3.3. Розрахунок кількості схем елементів узгодження

У якості схем узгодження задані логічні мікросхеми серії 133. Одна інтегральна схема узгодження має коефіцієнт розгалуження 10, $I_{ВХ}^0 = 1,6 \text{ мА}$, $I_{ВХ}^1 = 0,04 \text{ мА}$ і $C_{НСОГЛ} = 100 \text{ пФ}$.

Якщо навантаження на будь-якому вході нагромаджувача (НК) більше припустимої, потрібно розділити однойменні ланцюги нагромаджувача на групи і поставити додаткові схеми згодження.

Коефіцієнт розгалуження m по будь-якому вхідному ланцюзі НК визначається кількістю однойменних входів ВІС ЗП, які можливо одночасно підключити до виходу схеми. Коефіцієнт розгалуження кожного ланцюга визначається окремо з умови забезпечення узгодження струмів логічних 0 і 1 та ємкостей.

Кількість груп по кожним ланцюзі визначається зі співвідношення коефіцієнтів розгалуження і приймається найближче більше ціле число. Якщо модуль пам'яті має велику ємність, розкладання адреси здійснюється за допомогою пірамідальної схеми розмноження, яка містить кілька ярусів.

а) - для адресних ланцюгів:

$$m_{АВВВХ}^0 \geq \frac{I_A^0}{I_{ВВВХСОГЛ}^0} = \frac{28,8}{1,6 \cdot 10} = 1,8;$$

$$m_{АВВВХ}^1 \geq \frac{I_A^1}{I_{ВВВХСОГЛ}^1} = \frac{1,6}{0,04 \cdot 10} = 4;$$

$$m_{АС} \geq \frac{C_A}{C_{НСОГЛ} - C_{МА}} = \frac{271}{100 - 15} = 3,2.$$

Обираючи максимальне ціле значення $m_A = 4$, визначаємо, що щодо адресних ланцюгах усі ВІС ОЗП поділяються на 4 групи, причому всі однойменні адресні ланцюги групи приєднуються до виходу одного згідного елемента. ВІС К185РУ4 має 8 адресних входів, тому загальна кількість елементів узгодження по всім адресним входам дорівнює 32.

б) - для ланцюгів вибірки мікросхеми пам'яті:

$$m_{ВМВВВХ}^0 \geq \frac{I_{ВМ}^0}{I_{ВВВВХСОГЛ}^0} = \frac{14,4}{1,6 \cdot 10} = 0,87;$$

$$m_{ВМВВВХ}^1 \geq \frac{I_{ВМ}^1}{I_{ВВВВХСОГЛ}^1} = \frac{0,4}{0,04 \cdot 10} = 1;$$

$$m_{ВМС} \geq \frac{C_{ВМ}}{C_{НСОГЛ} - C_{МВМ}} = \frac{79}{100 - 15} = 0,93.$$

Обираємо максимальне ціле значення $m_{ВМ} = 1$. Тому кількість згідних елементів по всім виходам DC ВМ дорівнює 1.

в) - для інформаційних ланцюгів:

$$m_{\text{ивых}}^0 \geq \frac{I_{\text{и}}^0}{I_{\text{выхсогл}}^0} = \frac{14,4}{1,6 \cdot 10} = 0,87;$$

$$m_{\text{ивых}}^1 \geq \frac{I_{\text{и}}^1}{I_{\text{выхсогл}}^1} = \frac{0,1}{0,04 \cdot 10} = 0,25;$$

$$m_{\text{ис}} \geq \frac{C_{\text{и}}}{C_{\text{исогл}} - C_{\text{ми}}} = \frac{31}{100 - 15} = 0,37.$$

Обираємо $m_{\text{и}} = 1$. Таким чином, ВІС ОЗП з боку інформаційних вхідних ланцюгах поділяти на групи не потрібно.

2.3.4. Розрахунок вихідних ланцюгів модуля ОЗП

а) - кількість ВІС накопичувача, які необхідно об'єднати щодо вихідних ланцюгах:

$$P = \frac{N}{N_M} = \frac{1024}{256} = 4.$$

б) - сумарна ємність вихідного інформаційного ланцюга накопичувача у кожному розряді:

$$C_H = (P-1) \cdot C_{\text{вых}} + r \cdot C_{\text{вх}} + C_M = 4 \cdot 3 + 1 \cdot 3 + 15 = 30 \text{ нФ},$$

де $C_{\text{вых}}$ - вихідна ємність однієї ВІС ОЗП;

$C_{\text{вх}}$ - ємність одного вхідного ланцюга навантаження;

r - кількість вхідних ланцюгів навантаження;

C_M - монтажна ємність.

Перевіряємо умову: $C_H \leq C_{\text{ндоп}} = 80 \text{ нФ}$.

в) - розрахунок параметрів резистора R_H . Оскільки мікросхема К185РУ4 має вихід типу відкритий колектор, треба визначити значення навантажувального опору. З використанням матеріалів в додатку Д2 одержимо:

$$R_{H \max} = \frac{U_{\text{п min}} - U_{\text{min}}^1}{I_{\text{вых}}^1 + r \cdot I_{\text{вх}}^1 + (P-1) \cdot I_{\text{ут}}} = \frac{4,5 - 2,4}{0,2 + 0,04 + 3 \cdot 0,03} = 6,4 \text{ кОм};$$

$$R_{H \min} = \frac{U_{\text{п max}} - U_{\text{max}}^0}{I_{\text{вых}}^0 - r \cdot I_{\text{вх}}^0 + (P-1) \cdot I_{\text{ут}}} = \frac{5,5 - 0,4}{5 - 1,6 + 3 \cdot 0,03} = 1,5 \text{ кОм};$$

Обираємо $R_H = 2 \text{ кОм}$.

2.3.5 Функціональна схема модуля ОЗП

Функціональна схема модуля ОЗП наведена на рис.2.3. До складу модуля входять: нагромаджувач, який складається з 64-х мікросхем (4×16) серії **K185PY4**; схеми згідних елементів, які забезпечують розпаралелювання навантажувальних ланцюгів; дешифратор вибірки мікросхем; вихідний регістр даних на 16 розрядів. Адресний регістр у розглянутому модулі не передбачений.

Дешифратор **ВМ** побудований на елементі **K133ЛБ4**. На один із входів дешифратора подається сигнал вибіру модуля **ВМ**, який формується згідно базової адреси модуля.

В якості згідних схем при запису інформації застосовані елементи **K133ЛР1**. На кожний із входів цих елементів подається сигнал запису. Це забезпечує формування режиму запису інформації в обрані елементи пам'яті. Вихідний регістр побудований на **DE** - тригерах **K133ТМ5**.

2.3.6. Розрахунок часових параметрів модуля

Узагальнені часові діаграми роботи модуля **ОЗП** в режимах запису і зчитування інформації приведені на рис.2.4 і рис.2.5 відповідно.

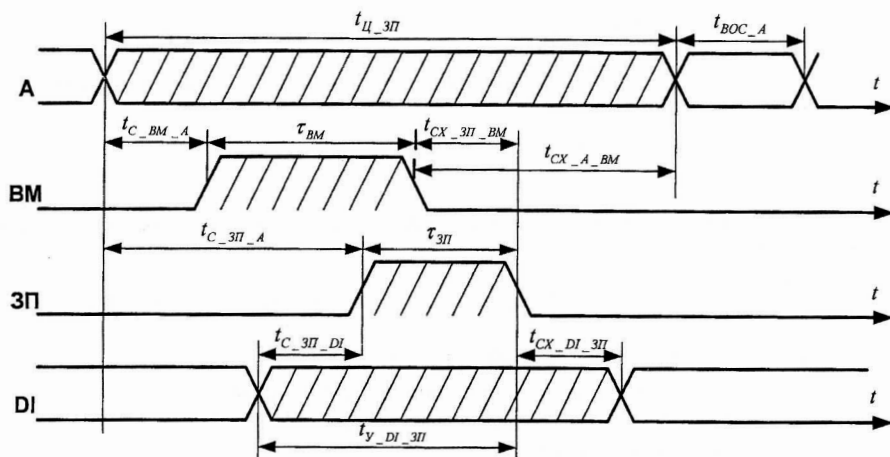


Рис. 2.4. Діаграма запису інформації

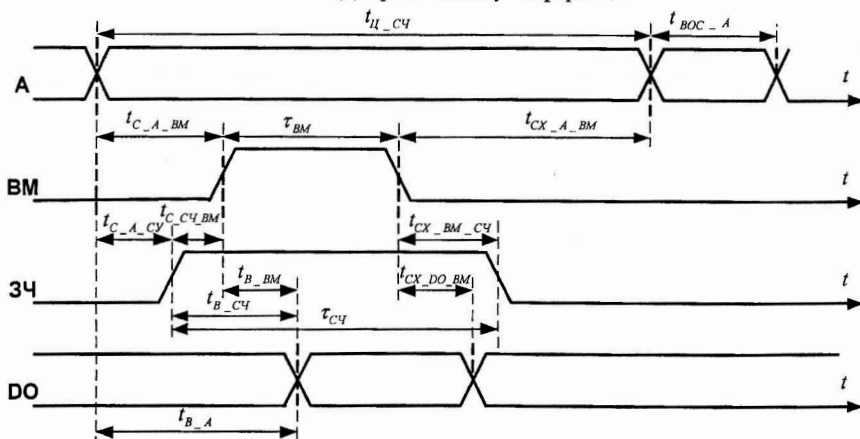


Рис. 2.5. Діаграма зчитування інформації

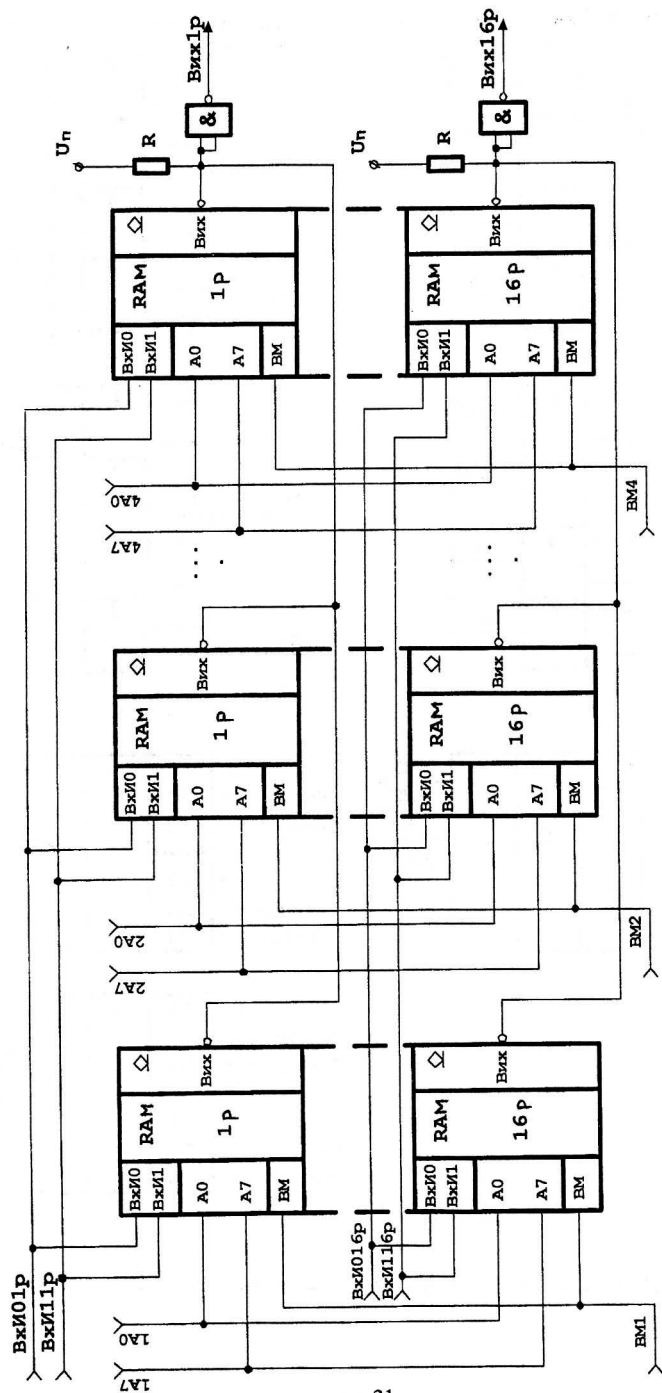


Рис. 2.3, а. Фрагмент функциональной схемы модуля ОЗП (початок)

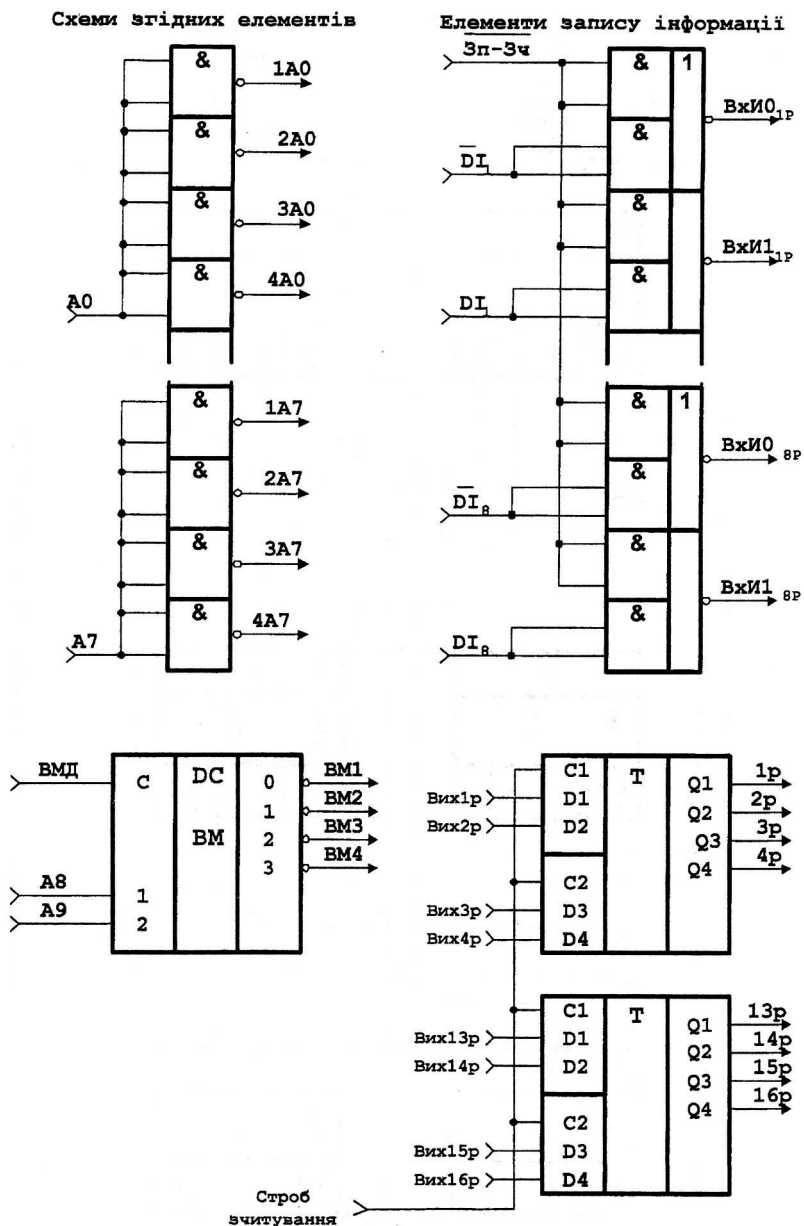
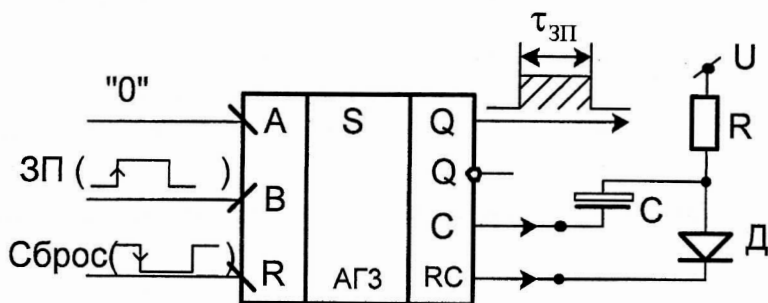


Рис. 2.3, б. Фрагмент функціональної схеми модуля ОЗП (продовження)



Наведені часові діаграми роботи модуля ОЗП є узагальненими. Це означає, що для конкретних типів мікросхем пам'яті затримки між окремими сигналами можуть бути відсутніми або можуть навіть мати інший знак. На діаграмах уведені наступні позначення сигналів і часових інтервалів:

Керуючі сигнали, які треба подавати на входи модуля, можуть бути імпульсного або потенційного типу. Розглянемо формули для розрахунку часових співвідношень у режимі зчитування інформації.

$$t_{BA_O3Y} = t_{PA} + t_{CD_BM} + t_{CC_BM} + \underbrace{t_{C_BM_A} + t_{B_BM}}_{\text{накопичувач}} + t_{CC_BYX},$$

- t_{CC_BM} - час затримки в схемах згідних елементів на вході накопичувача ОЗП;
 $t_{C_BM_A}$ - час зсуву між сигналами А і ВМ;
 t_{B_BM} - час вибірки щодо сигналу вибіру мікросхем модуля;
 $t_{CC_ВЫХ}$ - час затримки в схемах згідних елементів на виходах накопичувача ОЗП.

б) При потенційних сигналах ВМ і ЗЧ, які подаються одночасно з сигналами адреси:

$$t_{BA_OЗУ} = t_{PA} + t_{CC_A} + t_{B_A} + t_{CC_ВЫХ}.$$

в) Час циклу зчитування при імпульсному сигналі ВМ може бути визначене з формули:

$$t_{Ц_ЧТ} = t_{PA} + t_{CД_ВМ} + t_{CC_ВМ} + \underbrace{t_{C_ВМ_A} + \tau_{ВМ} + t_{CX_A_ВМ}}_{\text{накопичувач}} + t_{CC_ВЫХ} + t_{PD},$$

- де $\tau_{ВМ}$ - тривалість сигналу ВМ;
 $t_{CX_A_ВМ}$ - час збереження сигналу адреси А щодо сигналу ВМ;
 t_{PD} - час прийому інформації в зовнішній регістр даних.

г) При сигналі ВМ потенційного типу можливо записати:

$$t_{Ц_ЧТ} = t_{PA} + t_{CC_A} + \underbrace{t_{B_A} + t_{CX_ДІ_A}}_{\text{накопичувач}} + t_{CC_ВЫХ} + t_{PD},$$

д) У режимі запису інформації при імпульсному сигналі ВМ і потенційному сигналі ЗП час циклу запису може бути визначене з формули:

$$t_{Ц_ЗП} = t_{PA} + t_{CC_A} + \underbrace{t_{C_ЗП_A} + \tau_{ЗП} + t_{CX_A_ЗП}}_{\text{накопичувач}}.$$

е) У випадку, якщо сигнал ВМ постійного рівня, а сигнал ЗП – імпульсного типу, маємо:

$$t_{Ц_ЗП} = t_{PA} + t_{CC_A} + \underbrace{t_{C_ЗП_A} + \tau_{ЗП} + t_{CX_A_ЗП}}_{\text{накопичувач}}.$$

У разі, якщо обидва сигнали ВМ і ЗП (ЗЧ) імпульсного типу, при розрахунку часових співвідношень приймають максимальні значення затримок сигналів.

Після підстановки в приведені вище формули вихідних даних, одержуємо:

$$t_{BA_OЗУ} = 320 \text{ нс}, \quad t_{Ц_ЧТ} = 580 \text{ нс}, \quad t_{Ц_ЗП} = 580 \text{ нс}.$$

Для реалізації схем часових затримок можуть бути використані мікросхеми одновібраторів типу К155АГ1 або К155АГ3 [7, 8]. Вони дозволяють розширити тривалість коротких імпульсів або сформувати імпульси потрібної тривалісті. У

додатку Д3 розглянуті режими роботи і часові діаграми роботи одновібратора АГЗ [7, 8]. На рис. 2.3,в наведена схема формування тривалісті сигналу зчитування даних на базі мікросхеми АГЗ. Для $\tau_{зп} = 230$ нс із формул додатку Д3 розраховано: $R = 10$ кОм, $C = 76,6 \cdot 10^6$ пФ.

2.3.7. Визначення кількості корпусів мікросхем в модулі

Нагромаджувач містить 64 корпуси мікросхем K185PY4; адресні ланцюги – 8 корпусів мікросхем K133ЛБ3; ланцюга сигналу ВМ – 2 корпуси мікросхем K133ЛБ4; вхідні інформаційні ланцюги – 8 корпусів мікросхем K133ЛР1 та 4 корпуси мікросхем K133ЛБ3; вихідний регістр – 4 корпуси мікросхем K133ТМ5; формувачі тривалісті зсуву та збереження між сигналами – 8 корпусів мікросхем K155АГ3; резистори навантаження в ВІС накопичувача ($R_H = 2$ кОм) – 16 штук.

2.3.8. Споживана потужність модуля

Визначемо споживану потужність модуля ОЗП. Мікросхема типу K185PY4 виконана за КМОП - технологією. Тому вона споживає різну потужність у режимах збереження і звертання. Споживана потужність нагромаджувача:

$$P_{HK} = P_{XP} \cdot \left(Q - \frac{n}{n_M} \right) + P_{ПОТ} \cdot \frac{n}{n_M} \cdot \frac{\tau_{BM}}{t_{ц}} + P_{XP} \cdot \frac{n}{n_M} \cdot \frac{(t_{ц} - \tau_{BM})}{t_{ц}} =$$

$$= 150 \cdot \left(64 - \frac{16}{1} \right) + 375 \cdot \frac{16}{1} \cdot \frac{370}{580} + 150 \cdot \frac{16}{1} \cdot \frac{(580 - 370)}{580} = 11,7 \text{ Вт}.$$

Споживана потужність ланцюгами керування:

$$P_{УПР} = P_A + P_{BM} + P_{И} + P_{ВЫХ} + P_{RG} + P_{АГЗ} =$$

$$= 8 \cdot 116 + 2 \cdot 95 + 8 \cdot 76 + 4 \cdot 116 + 4 \cdot 253 + 8 \cdot 330 = 3,4 \text{ Вт}.$$

Споживана потужність усього модуля ОЗП:

$$P_{ОЗУ} = P_{HK} + P_{УПР} = 11,7 + 3,4 = 15,1 \text{ Вт}.$$

Розроблений модуль ОЗП має наступні параметри:

$$N = 1024, \quad n = 16, \quad t_{BAOЗУ} = 320 \text{ нс}, \quad t_{цсч} = t_{цзп} = 580 \text{ нс}, \quad U_n = +5B \pm 10\%,$$

$$P_{ОЗУ} = 15,1 \text{ Вт}.$$

2.4. Зміст звіту

1. Словесний опис роботи заданої мікросхеми ОЗП, експлуатаційні параметри, функціональне позначення, призначення вхідних і вихідних сигналів, часові діаграми роботи мікросхеми в режимах запису, зчитування і збереження інформації.

2. Розрахунок модуля ОЗП включає визначення наступних параметрів:

- кількості мікросхем у модулі Q;
- коефіцієнти з'єднання K_{BM} , $K_{И}$, K_A ;
- кількості вихідів дешифратора ВМ S_{BM} ;

г) значення навантаження щодо адресних (I_A^0, I_A^1, C_A), інформаційних (I_n^0, I_n^1, C_n) ланцюгах і ланцюгу вибірки мікросхеми ($I_{vm}^0, I_{vm}^1, C_{vm}$).

3. Визначення кількості схем згідних елементів для кожного вхідного ланцюга накопичувача.

4. Розрахунок вихідних ланцюгів модуля ОЗП:

а) кількості мікросхем ОЗП Р, які з'єднуються по вихідним ланцюгам;

б) сумарну ємність навантаження;

в) опір R_n (для мікросхеми ОЗП з відкритим колектором).

5. Функціональна схема модуля ОЗП з коротким описом її елементів.

6. Розрахунок часових параметрів модуля ОЗП: $t_{\text{АОЗП}}, t_{\text{ДЗСЧ}}, t_{\text{ДЗП}}$.

7. Визначення кількості корпусів мікросхем, які використовувалися у модулі ОЗП з указівкою типу кожної мікросхеми.

8. Розрахунок споживаної потужності ОЗП.

9. Список використаних літературних джерел.

2.5. Контрольні запитання

1. Обґрунтуйте формули обчислення коефіцієнтів об'єднання щодо адресних входів і виходу вибірки мікросхеми пам'яті.
2. Яким чином здійснюється розширення інформаційної ємності модуля пам'яті?
3. Наведіть формулу та дайте пояснення для розрахунку потужності накопичувача модуля пам'яті.

Література

1. Полупроводниковые запоминающие устройства и их применение. В.П.Андреев, В.В.Баранов, Н.В.Бекин и др. Под ред. А.Ю.Гордонова. – М.: Радио и связь, 1981. – 344 с.
2. Полупроводниковые БИС запоминающих устройств: Справочник. Под ред. А.Ю.Гордонова. – М.: Радио и связь, 1987. – 360 с.
3. Большие интегральные схемы запоминающих устройств: Справочник. А.Ю.Гордонов, Н.В.Бекин, В.В.Цыркин и др.; Под ред. А.Ю.Гордонова и Ю.Н.Дьякова. – М.: Радио и связь, 1990. – 288 с.
4. Лебедев О.Н. Микросхемы памяти и их применение. – М.: Радио и связь, 1990. – 160 с.
5. Аналоговые и цифровые интегральные схемы. / С.В.Якубовский, Н.А.Барканов, Л.И.Ниссельсон и др.; Под ред. С.В.Якубовского. – М.: Радио и связь, 1985. – 432 с.
6. Интегральные микросхемы: Справочник/ В.В.Тарабрин, Л.Ф.Лукин, Ю.Н.Смирнов и др.; Под ред. В.В.Тарабрина. – М.: Энергоатомиздат, 1985. – 528 с.
7. Шило В.Л. Популярныe цифровые микросхемы. Справочник – М.: Радио и связь, 1988. – 352с.
8. Микросхемы серии K155. Справочный материал. - Северодонецк: НПО "Импульс" - 1978.

Лабораторна робота №3

ПРОЕКТУВАННЯ МОДУЛЯ ОЗП ДИНАМІЧНОГО ТИПУ (DRAM) ЗІ СХЕМОЮ РЕГЕНЕРАЦІЇ ІНФОРМАЦІЇ

МЕТА РОБОТИ: придбання практичних навичок проектування модулів динамічної пам'яті на основі мікросхем серії K565.

ЗАГАЛЬНІ ПОЛОЖЕННЯ

У даний час для побудови пам'яті мікропроцесорних систем широкого застосування використовуються інтегральні схеми ЗП динамічного типу (DRAM). Ці схеми мають високу щільність упакування і малу споживану потужність. Однак у DRAM крім звичайних режимів запису і зчитування даних, потрібно проводити також у спеціальному режимі регенерацію (відновлення) інформації пам'яті.

Найбільш розповсюдженими елементами DRAM є інтегральні схеми серії **K565**, на базі яких можуть бути побудовані модулі пам'яті різного обсягу й організації. Для керування модулем DRAM може бути використаний стандартний контролер динамічної пам'яті (КДП) типу **K1810BT03**. КДП виробляє всі необхідні сигнали керування режимами зчитування, запису і регенерації інформації.

3.1. Послідовність виконання лабораторної роботи

1. Вивчити з використанням літературних джерел і матеріалів методичних вказівок функціональні електричні схеми і режими роботи заданої мікросхеми DRAM і контролера динамічної пам'яті (КДП) **K1810BT03**.
2. Розробити функціональну електричну схему модуля DRAM організацією N чисел по n розрядів кожне зі схемою регенерації інформації (адресація здійснюється з точністю до байта).
3. Розробити часові діаграми роботи модуля DRAM у режимах запису, зчитування і регенерації інформації.
4. Виконати розрахунок часових параметрів модуля.

3.2. Варіанти індивідуальних завдань

Варіанти індивідуальних завдань наведено у табл. 3.1 - табл. 3.3, де N - номер завдання.

Таблиця 3.1

Варіант= $(N)_{m4}$	Тип мікросхеми DRAM	Організація мікросхеми $N_M \times n_M$	Література
0	K565PY1	4K × 1	2,3
1	K565PY3	16K × 1	2
2	K565PY5	64K × 1	2
3	K565PY6	16K × 1	2

Таблиця 3.2

Варіант=(N) _{m8}	Організація модуля ОЗП N×n
0	64K×16
1	128K×32
2	256K×24
3	512K×16
4	64K×32
5	128K×24
6	256K×16
7	512K×32

Таблиця 3.3

Варіант=(N) _{m2}	Режими роботи КДП K1810BT03
0	Внутрішня регенерація
1	Зовнішня регенерація
0	Випереджальне зчитування
1	Нормальне зчитування
0	Зовнішній генератор
1	Внутрішній генератор

3.3. Методичні вказівки

Контролер динамічної пам'яті **K1810BT03** виробляє всі необхідні сигнали керування зчитуванням, записом та регенерацією інформації для ОЗП ємністю 4K, 16K, 64K и більше, який виконано на елементах пам'яті серії **K565** [1]. При використанні додаткових схемних рішень КДП може бути використаний для реалізації ОЗП великої ємності.

Умове графічне позначення контролера наведено на рис.3.1. Мікросхема **K1810BT03** має наступне призначення виведів:

AL6-AL0 - адресні входи молодшого байта комірок ОЗП.

B0/AL7, B1/OP1/AN7 - входи вибіру банку даних, які виконують різні функції в залежності від обраного режиму. Для режиму 16K висновки **B0** і **B1** виконують функцію вибіру банку пам'яті і беруть участь у формуванні сигналів на одному з вихідів **RAS0#** - **RAS3#**. Для режиму 64K на ці входи подаються адресні лінії **AL7** і **AN7**.

PCS# - захищений вибір кристала. Сигнал низького рівня на цьому вході ініціалізує виконання функцій зчитування / запису в ОЗП; якщо сигнал сформований, то цикл пам'яті анулювати не можливо.

WR# - вхідний сигнал, який вказує КДП на те, що центральний процесор (ЦП) виконує функції запису даних в ОЗП. Бере участь у формуванні сигналу **WE#**.

RD#/S1 - вхідний сигнал, який вказує КДП на те, що ЦП виконує функцію зчитування даних із ЗП. Сигнал **S1** використовується в режимі випереджального зчитування разом із сигналом **ALE**.

REFRQ/ALE - запит регенерації. Для режиму випереджального зчитування цей вхід (**ALE**) використовується для ініціалізації циклу зчитування і підключається до вивіду **ALE** центрального процесора.

OUT6# - OUT0# - вихідні сигнали адреси рядків і стовпців ЗП. Виробляються в правильній послідовності (спочатку молодший, потім старший байти) у супроводі сигналів **RAS#** і **CAS#**. У режимі регенерації на цих входах формуються адреси регенеруємих рядків.

WE# - ініціалізація запису; вихідний сигнал (строб), який використовується для виконання функції запису даних у ЗП.

CAS# - строб адреси стовпця. Вихідний сигнал, який виробляється КДП після формування на виходах **OUT6# - OUT0#** старшого байта ЗП. За допомогою цього сигналу здійснюється запис во внутрішні регістри пам'яті старшого байта адреси.

RAS0#, RAS1#, RAS2#/OUT7#, RAS3#/B0 - строби адреси рядка. Вихідні сигнали, які виконують різні функції в залежності від обраного режиму. У режимі 16К всі чотири сигнали є стробами молодшого байта адреси ЗП для різних банків і виконують аналогічну функцію як і сигнал **CAS#**. Низький рівень сигналу на одному з вихідів **RAS0# - RAS3#** виробляється в залежності від коду на входах **B0** і **B1**. У режимі 64К стробами адреси є тільки лінії **RAS0#** і **RAS1#**. Вихід **OUT7#** є вихідною лінією старшого розряду адреси рядків і стовпців, а вихід **B0 (RAS3#)** - використовується для вибору банку ЗП. Якщо **B0=0** формується строб банку 0 (**RAS0#**), якщо **B0=1** - строб банку 1 (**RAS1#**).

XACK# - готовність даних. Вихідний сигнал, який виробляє КДП наприкінці циклу зчитування / запису. Цей сигнал повідомляє ЦП про закінчення циклу взаємодії.

SACK# - готовність системи. Вихідний сигнал, який виробляє КДП на початку циклу звертання до пам'яті. Якщо запит до пам'яті від ЦП приходить на цикл регенерації, сигнал затримується до початку циклу зчитування / запису.

X0/OP2, X1/CLK - лінії для підключення зовнішнього кварцевого резонатора. Якщо вхід **X0/OP2** підключити до шини харчування +5В або через резистор опору 1 кОм до шини харчування +12 В, то вхід **X1/CLK** використовується для підключення зовнішнього генератора.

16K/64K# - вхід вибору режиму. Сигнал високого рівня на цьому вході задає режим роботи КДП з ОЗП ємністю 16К слів, а сигнал низького рівня - режим роботи з ОЗП ємністю 64К слів.

Структурна схема контролера K1810BT03 наведена на рис.3.2. Вона включає два функціональних блоки. Один з цих блоків приймає адреси комірок пам'яті від ЦП і формує їх у мультиплексивному режимі на виходах **OUT**. Крім того, у циклі регенерації на цих виходах формуються адреси рядків, які треба регенерувати. Блок включає: два буферних регістри **RGB1** і **RGB2** для прийому від ЦП 16-розрядної адреси комірки ОЗП, лічильник адреси регенерації **CT**, мультиплексори **MX1** і **MX2**.

Другий функціональний блок виконує операції синхронізації, арбітражу і формування повного набору керуючих сигналів для динамічного ОЗП і включає:

- буфер **BD** для прийому сигналів адресації, запису / зчитування від ЦП;
- тригер **TG**, який забезпечує запам'ятовування запиту на регенерацію від зовнішніх джерел;
- лічильник / таймер **RT**, який забезпечує необхідні запиту на регенерацію;
- схему **SYNC**, яка забезпечує прив'язку вхідних сигналів щодо фронтів тактового генератора КДП;
- арбітр **A**, який вирішує конфлікти між запитами на регенерацію і до пам'яті від центрального процесора;
- генератор синхротактів **SGW**;
- логічну схему **LC**, яка забезпечує формування сигналів **RAS#**, **CAS#**, **WE#** керування елементами ОЗП, а також квитируючих сигналів **XACK#** і **SACK#**.

Ініціалізація КДП здійснюється по входу **PCS#**. При подачі на цей вхід сигналу низького рівня відбувається дозвіл запитів циклу пам'яті від ЦП. Якщо сигнал **PCS# = 0**,

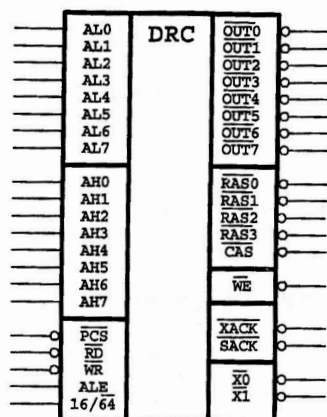


Рис. 3.1. Умовно графічне позначення КДП K1810BT03

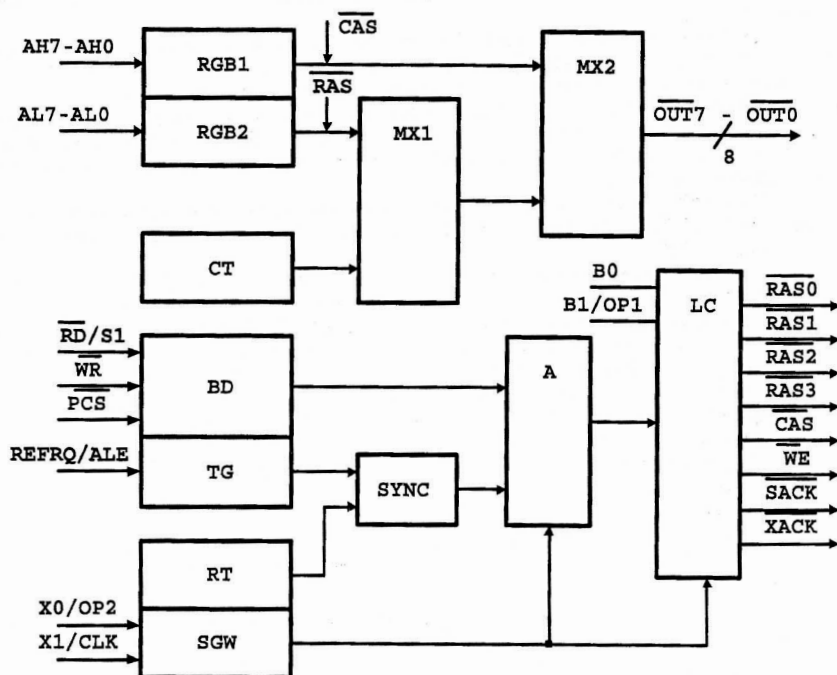


Рис. 3.2. Структура КДП 1810BT03

дозволяється прийом запитів на цикли зчитування або запису по входах **RD#** або **WR#**.

Регенерацію пам'яті контролер **K1810BT03** може виконувати двома засобами. Внутрішній цикл регенерації здійснюється за допомогою таймера регенерації **RT**, який розташований усередині КДП. Він забезпечує регенерацію всіх рядків ОЗП з періодом часу $2 / 4$ мс для 128 / 256 циклів відповідно. При виконанні зовнішній регенерації використовується вхід зовнішнього запиту на регенерацію **REFRQ**. При цьому може здійснюватися "прихована" для ЦП регенерація. Арбітр вирішує конфліктні ситуації при одночасних запитах від ЦП на цикли пам'яті і регенерацію.

При запиті циклів пам'яті від ЦП вхідні буфери **RGB1** і **RGB2** запам'ятовують адресу комірки пам'яті і видають її послідовно за допомогою адресних мультиплексорів **MX1** і **MX2** на вихіді **OUT7# - OUT0#** у супроводі сигналів **RAS#** і **CAS#**, які розділяють цикли пам'яті на два періоди (рис.3.3). У циклі запису на вихіді **WE#** формується негативний строб, який виконує функцію запису даних в елементи пам'яті (рис.3.4).

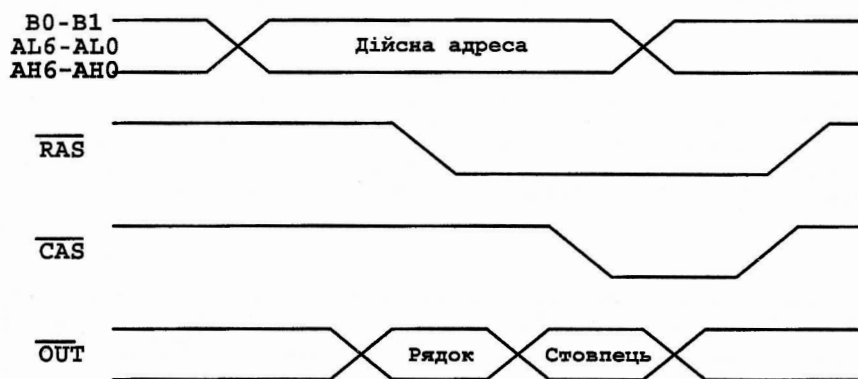


Рис. 3.3. Часові діаграми роботи КДП у циклі звертання до пам'яті

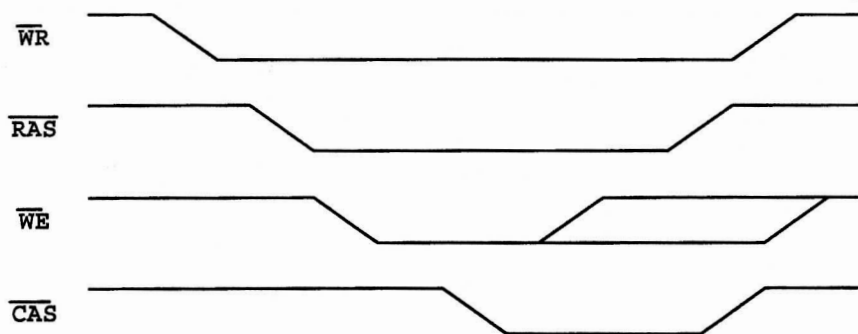


Рис. 3.4. Часові діаграми роботи КДП у циклі запису

У циклі регенерації на вихіді **OUT7# - OUT0#** видається адреса рядків регенерації (уміст лічильника СТ) і активізується стробовий сигнал рядків **RAS#**. Вихіді **CAS#**, **WE#**, **SACK#** та **XACK#** залишаються неактивними.

Цикли зчитування можуть виконуватися двома різними засобами: нормального або випереджального зчитування. Нормальний цикл зчитування даних виконується по запиту, який надходить на вхід **RD#** за умовою, коли сигнал на вході **PCS# = 0**. Запит на вході **RD#** повинний зберігатися доти, поки КДП не виставить сигнал готовності даних **XACK#**.

Випереджальне зчитування даних можливе тільки в режимі 16К. Цикли випереджального зчитування запитується по вхіду **ALE**, якщо сигнал **S1** знаходиться в активному стані. Фіксація запиту на цикли випереджального зчитування здійснюється по зрізу сигналу **ALE**. Режим випереджального зчитування даних задається підключенням входу **B1/OP1** до шини харчування (+12В) через резистор опору 5,1 кОм. У цьому режимі КДП може керувати тільки двома банками ОЗП (**RAS2#** та **RAS3#**) і не працює в режимі зовнішньої регенерації.

Мікросхема K1810BT03 може працювати в двох основних режимах, які задаються потенціалами на вході 16К / 64К# і орієнтують КДП на керування різними елементами пам'яті.

Режим 16К задається підключенням входу 16К / 64К# до шини харчування (+5В). Він орієнтований на керування модулем ОЗП, який виконано на елементах пам'яті **K565PY6**. Модуль ОЗП може бути ємністю 64К, яка розділена на чотири банки по 16К слів у кожному (у режимі нормального зчитування). Вибір банку в цьому режимі здійснюється адресним кодом на входах **B0** і **B1**. При цьому формується один з вихідних сигналів (**RAS0# - RAS3#**), які ініціалізують звертання до відповідного банку (табл.3.4).

Таблиця 3.4

Режим	Входи		Вихіді			
	B0	B1	RAS0#	RAS1#	RAS2#	RAS3#
16К	0	0	0	1	1	1
	0	1	1	0	1	1
	1	0	1	1	0	1
	1	1	1	1	1	0
64К	0	-	0	1	-	-
	1	-	1	0	-	-

Режим 64К задається підключенням входу 16К / 64К# до шини "Земля". Цей режим орієнтує контролер на керування модулем ОЗП, який виконано на елементах пам'яті **K565PY5**. При цьому забезпечується керування модулем ємністю 128К слів з двох банків по 64К кожний. Вибір банку здійснюється кодом на вході **B0**, який формує активний сигнал на вихіді **RAS0#** або **RAS1#** (табл.3.4).

Режим 4К орієнтує КДП на роботу з ОЗП на елементах пам'яті **K565PY1**. Він задається, як і режим 16К, за винятком того, що вивід **OP3** (лінія **AL6**) підключається до шини харчування (+12В) через резистор опору 5,1 кОм.

Операція випереджального зчитування можлива тільки у режимі 16К. Цей режим задається підключенням входу **OP1** до шини харчування (+12В) через резистор опір 5,1 кОм. У цьому випадку запит циклу зчитування виробляється зовнішнім сигналом по входах **ALE** і **S1**.

На рис.3.5 наведена структурна схема модуля ОЗП ємністю 64 Кбайт для роботи з КДП у режимі 16К нормального зчитування з зовнішнім генератором і внутрішньою регенерацією [1]. Адреси комірок модуля ОЗП займають простір від 80000 до 8FFFF. Модуль пам'яті включає у своєму складі 32 інтегральні схеми DRAM типу **K565PY6**, які розташовані у чотирьох банків по 8 мікросхем у кожному. Адресація до комірок

пам'яті кожного банку здійснюється сигналами **A13-A0** шини адреси. Ініціалізацію роботи модуля здійснює сигнал на вході **PCS#**, який формується шляхом дешифрації цих розрядів за допомогою логічних елементів АБО-НІ, І-НІ. На рис.3.6 наведено розподіл розрядів шини адреси коли виконується адресація до модуля пам'яті зі сторони ЦП.

Цикл регенерації пам'яті КДП виконується автономно. У випадку запитів циклів пам'яті з боку процесора КДП формує відповідні сигнали керування **RAS#**, **CAS#** і **WE#** модулем ОЗП в циклі зчитування, виставляючи дані на входи буфера **RG** і фіксує їх сигналом **XACK#**. Вихідні шини відкриваються на час дії сигналу **RD**. У циклі запису дані з мікропроцесора надходять безпосередньо на входи **D1** елементів пам'яті і фіксуються сигналом **WE#**.

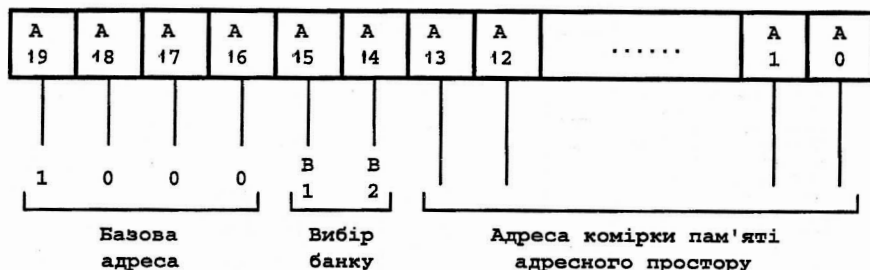


Рис. 3.6. Розподіл адресного простору модуля ОЗП

3.4. Зміст звіту

1. Словесний опис роботи заданої мікросхеми пам'яті, експлуатаційні параметри, функціональне позначення, призначення входних і вихідних сигналів.
2. Функціональна електрична схема розробленого модуля динамічної пам'яті зі схемою регенерації інформації (адресація з точністю до байту).
3. Короткий опис роботи модуля в режимах запису, зчитування і регенерації інформації.
4. Часові діаграми роботи модуля ОЗП в режимах запису, зчитування і регенерації інформації.
5. Список використаних літературних джерел.

3.5. Контрольні запитання

1. Чим і яким чином здійснюється формування сигналу **XACK#** ?
2. Поясніть виконання одного циклу зовнішньої регенерації **DRAM** з використанням контролера **K1810BT03**.
3. Наведіть часову діаграму випереджувального зчитування інформації з мікросхеми **DRAM**.

Література

1. Микропроцессорный комплект K1810: Структура, программирование, применение: Справочная книга / Ю.М.Казаринов, В.Н.Номоканов, Г.С.Подклетнов, В.Ф.Филипов; Под. ред. Ю.М.Казаринова.-М.:Высш. школа,1990.-269с.
2. Полупроводниковые БИС запоминающих устройств. Справочник / В.В.Баранов, Н.В.Бекин, А.Ю.Гордонов и др.; Под. ред. А.Ю.Гордонова и Ю.Н.Дьякова.-М.: Радио и связь.-1987.-360с.
3. Интегральные микросхемы. Справочник / Б.В.Тарабрин, Л.Ф.Лушин, Ю.Н.Смирнов и др.; Под. ред. Б.В.Тарабрина.-М.: Энергоатомиздат, 1985.-528с.

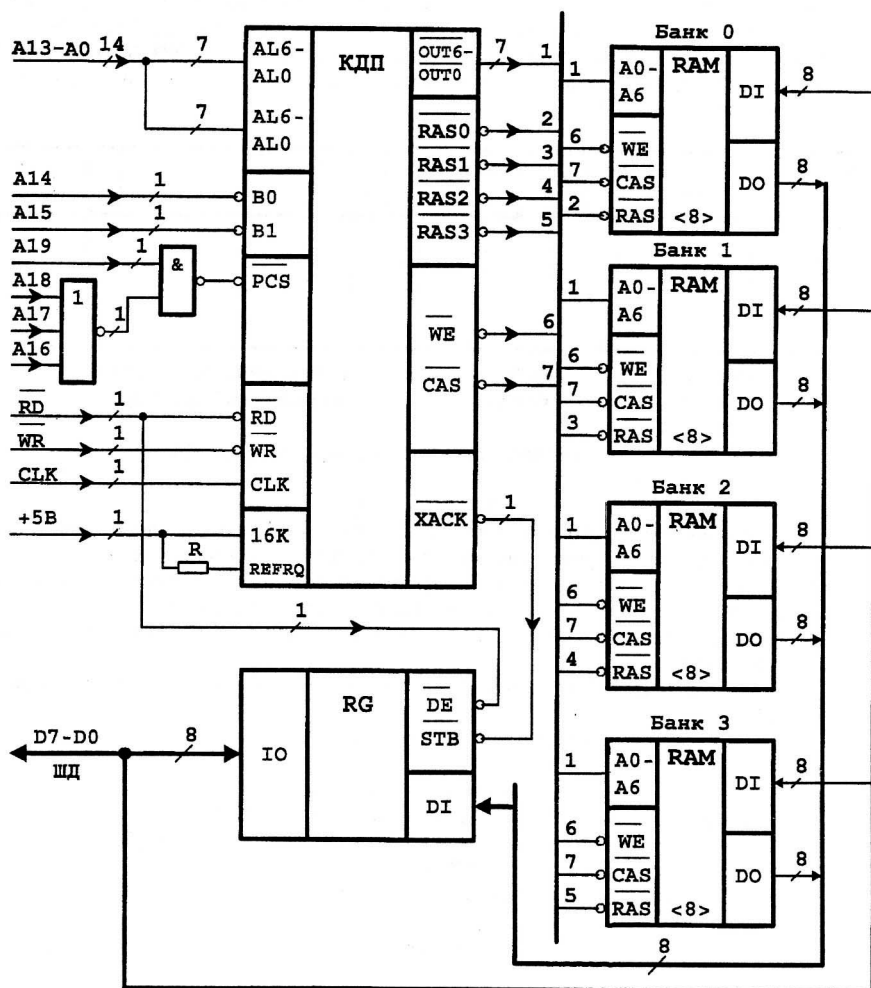


Рис. 3.5. Структурна схема модуля ОЗП в режимі 16К

Лабораторна робота №4

РЕГЕНЕРАЦІЯ DRAM ЗА ДОПОМОГОЮ КОНТРОЛЕРА DMA

МЕТА РОБОТИ: одержати навички розробки функціональних схем і програмування блоку регенерації динамічної пам'яті на основі таймера і контролера прямого доступу в пам'ять.

ЗАГАЛЬНІ ПОЛОЖЕННЯ

У деяких моделях персональних комп'ютерів для регенерації мікросхем динамічної пам'яті (оперативної пам'яті машини) використовується контролер прямого доступу до пам'яті - **DMA (Direct Memory Access)**. Такий контролер, набудований на визначений режим роботи, здатний підмінити контролер динамічної пам'яті і за рахунок цього зменшити устаткування всієї системи [1, 2, 3].

4.1. Послідовність виконання лабораторної роботи

1. Вивчити з використанням літературних джерел і матеріалів методичних вказівок функціональні схеми та алгоритми роботи таймера **i8254** і контролера DMA **i8237**.
2. Розробити функціональну схему блоку регенерації DRAM на основі мікросхем таймера і DMA. На мові асемблера привести програму ініціалізації контролера DMA і таймера, а також перевірки циклів регенерації пам'яті.
3. Розробити часову діаграму роботи вузлів блоку в заданому режимі.
4. Виконати розрахунок сумарного часу регенерації всієї пам'яті.

4.2. Варіанти індивідуальних завдань

Таблиця 4.1

Варіант = $(N)_{m4}$	Базова адреса мікросхеми таймера
0	01A7h
1	0236h
2	0156h
3	03C4h

Таблиця 4.2

Варіант = $(N)_{m3}$	Канал таймера
0	0
1	1
2	2

Таблиця 4.3

Варіант = $(N)_{m6}$	Режим роботи таймера
0	0
1	1
2	2
3	3
4	4
5	5

Таблиця 4.4

Варіант = $(N)_{m2}$	Лічильник таймера
0	Двоїчний
1	Двоїчно-десятковий

Таблиця 4.5

Варіант = $(N)_{m5}$	Базова адреса мікросхеми DMA
0	0A74h
1	236Ah
2	1234h
3	4AC2h
4	5FC8h

Таблиця 4.6

Варіант = $(N)_{m4}$	Канал DMA
0	0
1	1
2	2
3	3

Таблиця 4.7

Варіант = $(N)_{m3}$	Режим роботи DMA
0	Одиночна передача
1	Блокова передача
2	Передача за вимогою

Таблиця 4.8

Варіант = $(N)_{m2}$	Операція з CAR DMA
0	Інкремент
1	Декремент

Таблиця 4.9

Варіант = $(N)_{m6}$	Обсяг DRAM у кілобайтах
0	32K
1	16K
2	64K
3	128K
4	256K
5	512K

Таблиця 4.10

Варіант = $(N)_{m4}$	Час періоду регенерації, мкс
0	15,6
1	31,2
2	64,4
3	125

4.3. Методичні вказівки

Програмувальний таймер **i8254** (аналог **K1810ВИ54**) широко застосовується в системах на основі мікропроцесорів фірми **Intel** для організації служби системного часу та інших застосувань.

Таймер містить три незалежних канали (відповідно канал 0, канал 1 і канал 2). У середині кожного каналу знаходиться 16 - розрядний лічильник, який працює у режимі віднімання. Лічильник можливо програмним засобом налагодити на рахунок у двійковій або двоїчно-десятковій систем числення. Будь який канал таймера можливо запрограмувати на роботу в одному з шести режимів (відповідно режим 0, режим 1, ..., режим 5). Схема підключення таймера до системної шини наведена на рис. 4.1.

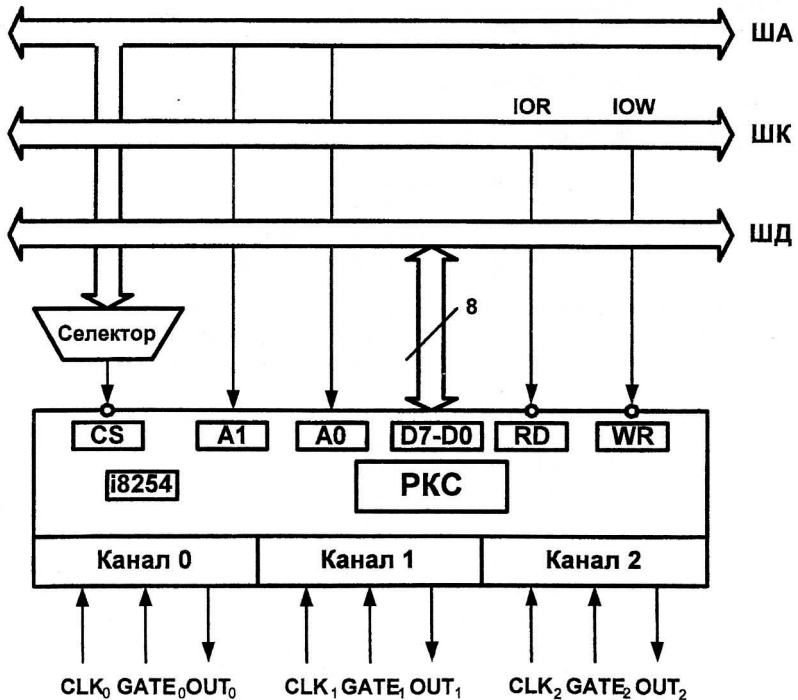


Рис. 4.1. Схема підключення таймера до системної шини

При програмуванні каналу в його регістр керуючого слова (**PKC**) передається керуюче слово (байт). Хоча в кожному каналі є свій **PKC**, усі вони мають одну

системну адресу і тому, з погляду програміста, усередині таймера знаходиться один **PKC**. Таймер визначає, якому каналу призначене керуюче слово по внутрішньому вмісту цього слова.

Адресна інформація на лініях **A1** і **A0** задає один з чотирьох пристроїв, які адресуються усередині таймера:

A1=0, A0=0 → канал 0;
A1=0, A0=1 → канал 1;
A1=1, A0=0 → канал 2;
A1=1, A0=1 → **PKC**.

CLK0 - CLK2 – вхідні лінії синхронізації. На ці входи надходять імпульси для управління лічильниками каналів таймера. Декремент лічильників відбувається по задньому фронту імпульсів **CLK**.

GATE0 - GATE2 – керуючі входи. Якщо на вході **GATE** сигнал "1" - дозволяється виконання функцій таймера. У деяких режимах зняття і подача одиниці на вході **GATE** ініціалізує перезапуск лічильника.

OUT0 - OUT2 – виходи каналів. Вигляд сигналу на цих виходах залежить від режиму, на який налагоджений канал.

Формат керуючого слова каналу наведено на рис. 4.2.

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

Рис. 4.2. Формат керуючого слова каналу таймера

Розряди **D7** і **D6** задають номер каналу, у який передається керуюче слово:

D7=0, D6=0 → канал 0;
D7=0, D6=1 → канал 1;
D7=1, D6=0 → канал 2;
D7=1, D6=1 → спеціальна команда.

Таймер зв'язаний з **CSH 8** - розрядною шиною даних (рис.4.1). У той же час формат лічильника каналу – два байти. Тому мають місто різні варіанти завантаження лічильника і зчитування його вмісту. Ці варіанти задають розряди **D5** і **D4** керуючого слова:

D5=0, D4=0 → спеціальний режим «зчитування на літу»;
D5=0, D4=1 → передається тільки старший байт;
D5=1, D4=0 → передається тільки молодший байт;
D5=1, D4=1 → передаються обидва байти (спочатку молодший, а потім старший).

Прочитати вміст лічильника каналу можливо двома засобами. Перший засіб вимагає зупинки рахунку. Для цього треба припинити подачу імпульсів на вхід **CLK** або зняти одиничний сигнал зі входу **GATE**. Другий засіб не вимагає зупинки рахунку і тому називається «зчитування на літу». У цьому випадку в налагоджений і працюючий канал необхідно передати ще одне керуюче слово (його формат наведено на рис.4.3).

D7	D6	0	0	*	*	*	*
----	----	---	---	---	---	---	---

Рис. 4.3. Формат керуючого слова для режиму «зчитування на літу»

Тут, як і раніше, розряди **D7** і **D6** задають номер каналу, а зірочки означають байдужне значення. Одержавши такий наказ, канал автоматично фіксує поточне значення лічильника в буфері, відкля його потім можливо зчитувати за одну або дві передачі.

Розряди **D3-D1** (рис.4.2) задають режим, у якому буде працювати канал (відповідно 000 – режим 0, 101 – режим 5).

Розряд **D0** задає систему числення, у якій буде працювати канал. При **D0** = 0 задається двійкова, при **D0** = 1 – десяткова система числення.

Як приклад розглянемо роботу таймера в другому режимі – програмувальний дільник (генератор) частоти. У цьому режимі лічильник таймера працює як дільник частоти F на N , де N – число, яке завантажується в лічильник. При цьому з сигналу, який надходить на вхід **CLK** мікросхеми, на її виході **OUT** формується сигнал з частотою F/N . Тривалість рівня логічної одиниці для формування вихідного сигналу дорівнює $(N-1) \cdot T$, а тривалість рівня логічного нуля – T , де T – один період синхронізації лічильника сигналом **CLK**. Іншими словами, щораз при досягненні лічильником одиниці, на виході **OUT** з'являється негативний імпульс з тривалістю в один такт. Вхід **GATE** використовується для апаратної синхронізації. Так по сигналу **GATE**, рівному нулю, на виході **OUT** установлюється сигнал 1, а з переходом **GATE** до значення 1 рахунок відновлюється з початкового значення N .

Прямий доступ до пам'яті

На час прямого доступу в пам'ять (**ПДП**) процесор відключається від **СШ**, а керування обміном даними між зовнішнім пристроєм (**ЗП**) і оперативною пам'яттю (**ОП**) бере на себе контролер **ПДП** (**КПДП**).

Нехай, наприклад, потрібно передати байт інформації із **ОП** в порт **ЗП**. Попередньо **КПДП** повинний бути задати адресу комірки **ОП**, відкля буде передаватися інформація і напрямком передачі. Коли **ЗП** буде потрібно прочитати даний байт, він формує запит на **ПДП** (**DREQ**), який надходить на контролер **ПДП**. Останній формує активний сигнал **HRQ** на вхід **HOLD** мікропроцесора (**МП**). Одержавши цей сигнал, **МП** припиняє виконання програми, відключається від системної шини і виробляє сигнал підтвердження захоплення шини **КПДП** **HLDA**. З цього моменту керування системною шиною бере на себе **КПДП**.

Для нашого приклада **КПДП** виконує наступні дії: виробляє сигнал підтвердження **ПДП** (**DACK**), який підключає порт **ЗП** до **ШД**, виставляє на **ША** адресу комірки **ОП**, виробляє сигнали **MEMR#** (читання **ОП**) і **IOW#** (запис у порт **ЗП**), які задають вид операції відповідно для **ОП** і **ЗП**. Після передачі байта **КПДП** знімає активний сигнал зі входу **HOLD** **МП**, повертаючи останньому керування системною шиною.

Передача байта з **ЗП** в **ОП** виробляється аналогічним чином, тільки міняються сигнали, які задають вид операції для **ОП** і **ЗП**.

При **ПДП** можлива передача цілого блоку інформації. У цьому випадку **КПДП** попередньо задає напрямком передачі, початкову адресу блоку в **ОП** і розмір цього блоку. Після передачі чергового байта сигнал **HOLD** не знімається, уміст лічильника байт зменшується на одиницю, а адреса комірки **ОП** зменшується (або збільшується) на одиницю у залежності від налагодження **КПДП**. Процес обміну продовжується доти, поки лічильник байт не стане рівним нулю.

Контролер **i8237** (аналог **K1810BT37**) містить у собі чотири незалежних канали **ПДП**: відповідно канал 0, канал 1, канал 2 і канал 3. Кожен канал може знаходитися в двох основних режимах: у режимі програмування (ініціалізації) і режими **ПДП**. У процесі програмування канал може бути набудований на роботу в наступних режимах:

1. **Режим одиночної передачі.** У цьому режимі канал здійснює передачу байта інформації між ОП і портом ЗП, після чого КППД повертає шину МП. У цьому режимі можливо задавати блок інформації, але передаватися він буде побайтно.

2. **Режим блокової передачі.** У цьому режимі канал робить передачу блоку інформації. КППД не повертає шину МП доти, поки не буде переданий весь блок.

3. **Режим блокової передачі за вимогою.** За змістом це та ж блокова передача, однак, після передачі кожного байта КППД перевіряє вхід запиту на ПДП (DREQ) каналу. Якщо на ньому є активний сигнал, ЗП готов до подальшого обміну і передача продовжується. Якщо на вході DREQ з'являється пасивний сигнал, ЗП до подальшого обміну не готов. Передача у цьому разі припиняється (до нового запиту від ЗП) і шина повертається МП.

4. **Каскадний режим.** У системі може знаходитися декілька КППД, які при цьому з'єднуються каскадно (один ведучий, інші відомі). Сигнал з вихіду HRQ відомого заводиться не на вхід HOLD МП, а на вхід DREQ одного з каналів ведучого. Якщо будь-який канал КППД налагоджений на каскадний режим, значить до входу DREQ цього каналу підключений відомий контролер.

Звичайно, по закінченні передачі блоку даних канал контролера треба перепрограмувати. Однак якщо канал налагоджений на режим автоініціалізації, його перебудова не потрібна, оскільки канал перезавантажується вихідними даними.

Щоб уникнути конфліктних ситуацій, усім каналам КППД привласнюється свій пріоритет. За замовчуванням старший пріоритет має канал 0, однак є можливість програмно змінювати пріоритети каналів.

Частина внутрішніх регістрів КППД мають формат вісім біт, інша частина – 16 біт. Для завантаження 16-розрядного регістра, потрібно виконати дві передачі даних. Перед початком завантаження (або зчитування) 16-бітного регістра контролера i8237 необхідно програмним засобом установити в "0" тригер «старший/молодший», який знаходиться усередині контролера.

Системні адреси 0h, 2h, 4h і 6h виділені 16-розрядним регістрам адреси (CAR) відповідно каналам 0, 1, 2 і 3.

Системні адреси 1h, 3h, 5h і 7h привласнені лічильникам байт (CWR) відповідних каналів. По цим адресам завантажуються розміри блоків даних (кількість байт у блоці мінус 1). Лічильники CWR мають формат 16 біт, але максимальний розмір блоку 2^{14} . Всі інші регістри КППД мають формат 8 біт.

За адресою 8h завантажуються командне слово КППД. Воно відноситься відразу до всіх каналів і має формат, який показано (у спрощеному вигляді) на рис.4.4.

1	0	0	0	0	0	D1	D0
---	---	---	---	---	---	----	----

Рис. 4.4. Формат командного слова КППД

D0 = 1 → режим «пам'ять - пам'ять»;

D0 = 0 → нормальна робота;

D1 = 1 → адреса джерела незмінна;

D1 = * → якщо D0 = 0.

За адресою 0Bh виробляється налагодження каналів на режими роботи (для кожного каналу). Керуюче слово має формат, який наведено на рис.4.5.

D7	D6	D5	D4	D3	D2	N1	N0
----	----	----	----	----	----	----	----

Рис. 4.5. Формат керуючого слова

N1 і N0 задають двійковий номер каналу.

D7 і D6 задають режими роботи каналу (00 – передача за вимогою, 01 – одиночна передача, 10 – блокова передача, 11 – каскадний режим).

D5 = 1 – автодекремент адреси.

D5 = 0 – автоінкремент адреси.

D4 = 1 – є режим автоініціалізації.

D4 = 0 – немає режима автоініціалізації.

D3 і D2 – задають напрямок передачі (01 – з ЗП в ОП, 10 – з ОП в ЗП).

Передача будь-якого байта за адресою 0Ch скидає тригер старший/молодший. Передача будь-якого байта за адресою 0Dh викликає програмне скидання контролера (аналогічно дії сигналу RESET).

Блок регенерації DRAM

Спрощена функціональна схема блоку регенерації динамічної пам'яті наведена на рис.4.6. До складу схеми входять генератор тактових імпульсів CLK – G, інтервальний таймер i8254, тригер керування ТК, контролер DMA i8237 і блок динамічної пам'яті (на 64К комірки).

Період регенерації визначається системним таймером i8254. Власне регенерацію пам'яті керує контролер прямого доступу до пам'яті. Період регенерації T_{рег} прийнято рівним 15 мкс. Тому кожні 15 мкс таймер активізує сигнал DREQ0, який викликає цикл ПДП. Нульовий канал DMA набудований на регенерацію динамічної пам'яті. Він налагоджений на режим зчитування даних з DRAM деякому фіктивному пристрою (ФП).

Після операції зчитування здійснюється цикл регенерації запам'ятовуваних елементів DRAM по заданій адресі рядка. В якості сигналу RAS# можливо у режимі одиночної передачі використовувати сигнал DASK0# з контролера DMA, який набудовано на низький активний рівень.

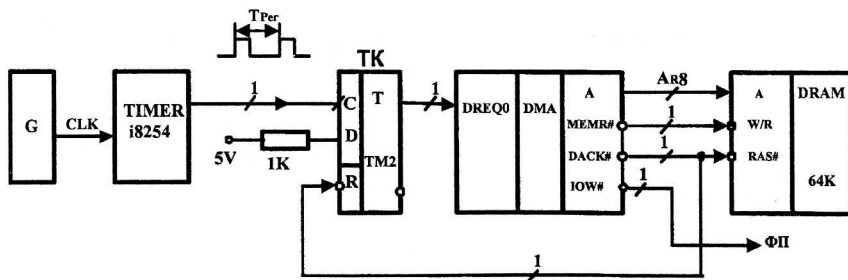


Рис. 4.6. Спрощена функціональна схема блоку регенерації пам'яті

Нижче, як приклад, приведена програма ініціалізації блоків пристрою регенерації.

; Ініціалізація DMA

```

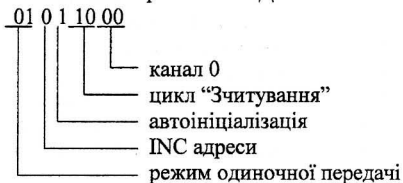
OUT  DMA+0Ch,AL
MOV  AL,0FFh
OUT  DMA+1,AL
OUT  DMA+1,AL
MOV  AL,58h

```

0 → T_{МЛ/СТ}

Налагодження лічильників CWR і WCR на 64K

Установлення режимів ПДП:



```

OUT  DMA+0Bh,AL
MOV  AL,00h

```

058h → WR

```

OUT  DMA+0,AL

```

0 → CAR_{МЛ}, 0 → BAR_{МЛ}
(адреса рядка регенерації A_R)

```

MOV  AL,0
OUT  DMA+8,AL

```

Установлення регістра CR:

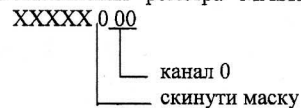


```

OUT  DMA+10,AL

```

Установлення регістра MASK:



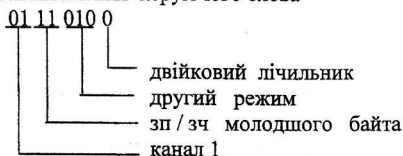
; Ініціалізація таймера

```

MOV  AL,74h

```

Завантаження керуючого слова



```

OUT  TIMER+3,AL

```

```
MOV AL,0FFh ;
OUT TIMER+1,AL ;      Видача константи перерахування
```

У програмі **DMA** і **TIMER** – символічні базові адреси контролера **DMA** і таймера.

4.4. Зміст звіту

1. Докладна функціональна схема блоку регенерації з коментарями та вказівкою всіх сигналів. Привести опис режимів ініціалізації та роботи блоків системи.
2. Програма (на мові асемблера) ініціалізації таймера, контролера DMA, а також завдання кількості циклів регенерації **DRAM** у заданому режимі роботи.
3. Часова діаграма роботи вузлів блоку регенерації.
4. Розрахунок часових параметрів.

4.5. Контрольні запитання

1. Яким чином здійснюється регенерація однієї строки динамічної пам'яті?
2. Поясніть алгоритм взаємодії процесора з контролером регенерації при виконанні циклів запису та регенерації.
3. Чим і яким чином здійснюється запуск таймера i8254 у блоці регенерації DRAM?

Література

1. Столлинг У. Структурная организация и архитектура компьютерных систем. - М.: Издательский дом "Вильямс", 2002. - 896 с.
2. Микропроцессорный комплект K1810. Справочная книга / Под ред. Ю.М. Казаринова. - М.: Высшая школа, 1990. - 269с.
3. Алексенко А.Г. и др. Проектирование радиоэлектронной аппаратуры на микропроцессорах. - М.: Радио и связь, 1984. - 270с.
4. Хвоц С.Т., Варлинский Н.Н., Попов Е.А. Микропроцессоры и микроЭВМ в системах автоматического управления: Справочник. - Л.: Машиностроение, 1987. - 640с.
5. Самофалов К.Г., Викторов О.В. Микропроцессоры. - Библиотека инженера. - 2-е изд. перераб. и доп. - К.: Техника, 1989 - 312с.
6. Гузик В.Ф., Каляев В.А., Костюк А.И., Третьяков С.В. Организация ЭВМ и систем: Учебное пособие. - Таганрог: Издательство ТРТУ, 1999. - 127с.

Лабораторна робота № 5

МОДЕРНІЗАЦІЯ ОПЕРАТИВНОЇ ПАМ'ЯТІ КОМП'ЮТЕРА З ЗАСТОСУВАННЯМ МОДУЛІВ SIMM'S

МЕТА РОБОТИ: вивчити принцип функціонування і режими роботи оперативної пам'яті машини, а також можливості її модернізації з використанням стандартних модулів SIMM'S.

ЗАГАЛЬНІ ПОЛОЖЕННЯ

Мікросхеми пам'яті можуть установлюватися безпосередньо на плату, але для основної пам'яті частіше використовуються модулі SIMM або DIMM [3]. Сучасні модулі пам'яті мають шину даних розрядністю 1, 4 або 8 байт. Набір сигналів SIMM - модуля в основному збігається із сигналами мікросхем динамічної пам'яті. Кількість і тип мікросхем визначається необхідною розрядністю та обсягом збережених даних. Архітектура модулів забезпечує можливість побайтного звертання. Вибір байт виробляється окремими входами CAS# для кожного байта.

5.1. Послідовність виконання лабораторної роботи

1. Вивчити з використанням літературних джерел і матеріалів методичних вказівок функціональні схеми, режиму роботи і часові діаграми блоків оперативної пам'яті сучасних ПЕОМ.
2. Розробити структурну схему карти розширення основної пам'яті з використанням модулів SIMM's у заданих режимах функціонування з указівкою регістрів DRB, слотів SIMM's, сигналів RAS# і CAS# . Дати короткі пояснення розробленої схеми.
3. Привести часові діаграми роботи пам'яті в режимах запису, зчитування і регенерації інформації.
4. Привести формули і виконати розрахунок часових параметрів модернізованої пам'яті.

5.2. Варіанти індивідуальних завдань

Таблиця 5.1.

Варіант = (N) _{m2}	Режим роботи DRAM
0	FPM
1	EDO

Таблиця 5.2.

Варіант = (N) _{m3}	Використання інтерлівінга
0	0 банків
1	2 банки
2	4 банки

Таблиця 5.3.

Варіант = (N) _{m3}	Обсяг пам'яті на материнській платі, N _m
0	4 Мбайт
1	8 Мбайт
2	16 Мбайт

Таблиця 5.4.

Варіант = (N) _{m10}	Використовувані модулі SIMM'S
0	256 К×9 (30-0), 256 К×36 (72-Д)
1	256 К×9 (30-Д), 256 К×36 (72-0)
2	256 К×9 (30-0), 256 К×36 (72-0)
3	1 М×9 (30-Д), 512 К×36 (72-Д)
4	1 М×9 (30-0), 512 К×36 (72-0)
5	1 М×9 (30-0), 512 К×36 (72-0)
6	1 М×9 (30-Д), 1М×36 (72-Д)
7	4 М×9 (30-Д), 1М×36 (72-0)
8	4 М×9 (30-0), 2М×36 (72-Д)
9	1 М×9 (30-Д), 4М×36 (72×Д)

Таблиця 5.5.

Варіант = (N) _{m4}	Обсяг модернізованої пам'яті
0	2 × N _m
1	4 × N _m
2	8 × N _m
3	16 × N _m

Таблиця 5.6.

Варіант = (N) _{m3}	Специфікація швидкодії DRAM
0	-5
1	-6
2	-7

Таблиця 5.7

Характеристики типів DRAM		
Специфікація	FPM	EDO
	-5, -6, -7	-5, -6, -7
Час доступу, нс	50, 60, 70	50 60 70
Тривалість циклу CAS, нс	30, 35, 40	20 25 30
Максимальна частота при пакетному циклі зчитування, МГц	66, 50, 40, 5-3-3-3	66 50 40 5-2-2-2

Інші дані, які необхідні для виконання роботи, обоснуються і приймаються студентом самостійно.

5.3. Методичні вказівки

У режимі швидкого сторінкового обміну **FPM (Fast Page Mode)** адреса рядка виставляється на шині тільки один раз і сигнал **RAS#** утримується на низькому рівні на час усіх наступних звертань до стовпчиків по сигналам **CAS#** (це можуть бути цикли запису або зчитування даних).

Режим **EDO (Extended Data Output) DRAM** – пам'ять з розширеним вивідом даних. Іноді цей режим називають гіперсторінковим режимом обміну **HPM (Hyper Page Mode)**. У цьому режимі забезпечується конвеєризація роботи пам'яті при зчитуванні даних за рахунок уведення на виході додаткового регістра на **DE** - тригерах типу «засувка». Регістр прозорий при низькому рівні сигналу **CAS#**, а по фронту цього сигналу відбувається фіксація поточного значення вихідних даних до наступного спаду сигналу **CAS#**.

Для зменшення часу звертання до пам'яті використовується режим інтерлівінга (**Interleaving**). Він припускає рівнобіжну адресацію пам'яті зі зсувом на слово. Для організації інтерлівінга використовують кількість банків пам'яті, яка кратна ступені двійки (2, 4, 8 і т.д.) Як приклад, на рис. 5.1. наведена часова діаграма зчитування даних у режимі **FPM** і використанні інтерлівінга з двома банками пам'яті.

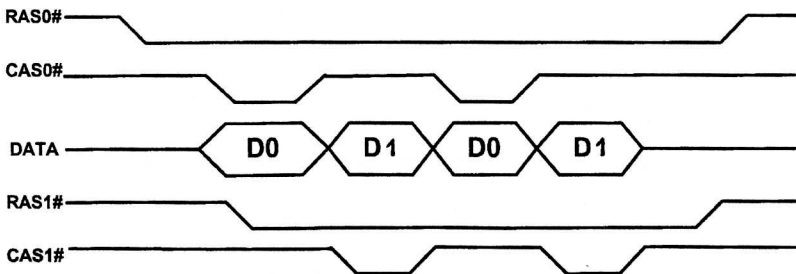


Рис. 5.1. Часова діаграма зчитування даних із двох банків пам'яті

Специфікація швидкодії **DRAM** – це останній елемент позначення мікросхеми. Він вказує час доступу в десятках наносекунд.

Час доступу визначається від початку операції читання (спаду сигналу **RAS#**) до появи достовірних даних на виході.

Тривалість циклу **CAS#** визначає період надходження чергових даних на вихідні шини в середині пакетного циклу.

Максимальна частота визначається як частота системної шини, при якій кількість тактів на цикл не перевищує значень, проголошених оптимальними для даного типу пам'яті. Оскільки вона визначається не тільки мікросхемами пам'яті, але і затримками елементів обрамлення, у реальних системних платах можливі і інші співвідношення.

Дані табл.5.7 свідчать, що час доступу і тривалість першого циклу читання для режимів **FPM** і **EDO** однакова (5 періодів сигналу **CLK** системної шини). Різниця спостерігається в наступних трьох тактах пакетного циклу.

Регістри **DRB (Dram Boundary Register)** визначають верхні і нижні границі адрес для кожного рядка **DRAM** у масштабі по 4М байтів. Системний контролер **TSC** (з **PCI** набору **82430FX**) підтримує п'ять рядків **DRAM**. Кожен рядок має розрядність в 64 біта.

DRB0 визначає обсяг пам'яті в рядку 0 блоками по 4М байт. Регістр **DRB1** визначає сумарний обсяг пам'яті в рядках 0 і 1. Регістр **DRB2** визначає сумарний обсяг пам'яті в рядках 0, 1 і 2. Регістр **DRB3** визначає сумарний обсяг пам'яті в рядках 0, 1, 2, 3. Регістр **DRB4** визначає сумарний обсяг пам'яті в рядках 0, 1, 2, 3 і 4. **DRB4** визначає максимальну кількість пам'яті в системі в 4М байтних блоках.

Масив **DRAM** може складатися з односторонніх (О) або двосторонніх (Д) **SIMM**'s розміром 512К × 32, 1М × 32, 2М × 32 і 4М × 32.

Масив пам'яті повинний бути сконфігурований (приклад наведено на рис.5.2). Контролер **TSC** керує сигналами **RAS#** і **CAS#**. Якщо використовуються односторонні **SIMM**'s, то задіяний парний сигнал **RAS#** і непарний **RAS#** не приєднується. Якщо ж використовуються двосторонні **SIMM**'s, то подаються обидва сигнали **RAS#** (з парним і непарним номерами).

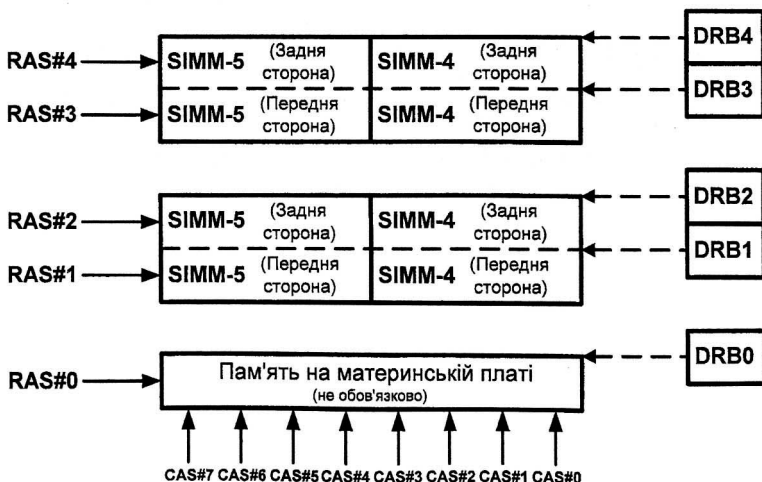


Рис. 5.2. Приклад організації оперативної пам'яті

Нехай, наприклад, масив пам'яті складається з чотирьох однобічних **SIMM's**, утворюю усього обсяг **DRAM** у 16М байтів. Два **SIMM's** необхідні для створення одного рядка пам'яті розміром у 8М байтів. У цьому випадку **DRB** регістри програмуються в такий засіб:

DRB 0 – 02 h (2**SIMM's** – 8 Mb у рядку);

DRB 1 – 04 h (2**SIMM's** – 8 Mb у рядку);

DRB 2 – 04 h

DRB 3 – 04 h

DRB 4 – 04 h

} порожні рядки

5.4. Зміст звіту

1. Структурна схема карти розширення основної пам'яті машини. Привести розподіл адресних ліній пам'яті.
2. Часові діаграми роботи пам'яті в різних режимах.
3. Розрахунок часових параметрів модернізованої пам'яті.

5.5. Контрольні запитання

1. Яким чином розрахувати кількість адресних ліній, які потрібні для адресації обсягу блока пам'яті (блок виконано на модулях **SIMM's**) ?
2. Поясніть різницю у режимах роботи пам'яті **EDO** та **FPM**.
3. З яких міркувань здійснюється вибір кількості ліній **RAS#** та **CAS#** у блоці динамічної пам'яті ?

Література

1. Intel 82430 FX PCI set.
2. Intel 430 MX PCI set.
3. Гук М. Аппаратные средства IBM PC. Энциклопедия. – СПб.: «Питер», 2000. – 816с.
4. Айден К., Колесниченко О., Крамер М. и др. Аппаратные средства PC. 2-е, перер. и допол. – СПб.: BHV, С. Петербург, 1998. – 608с.
5. Руководство по архитектуре IBM PC AT / Под ред. М.Л. Мархасина. – Минск: ООО «Консул», 1992. – 949с.
6. Столлинг У. Структурная организация и архитектура компьютерных систем. - М.: Издательский дом "Вильямс", 2002. - 896 с.

Лабораторна робота №6

ПАМ'ЯТЬ ТИПУ ЧЕРГИ І СТЕКА

МЕТА РОБОТИ: вивчити принцип роботи блоків пам'яті типу черги і стека, а також розробити функціональні електричні схеми і програмну модель пам'яті.

ЗАГАЛЬНІ ПОЛОЖЕННЯ

Пам'ять з послідовним доступом будується з використанням просування даних у ланцюгах елементів (на подоби з регістрами зсування) або зі збереженням даних в адресному ЗП з необхідним керуванням адресою доступу. Основними представниками цих видів пам'яті є буфер **FIFO** і стек. Останній широко використовується в мікропроцесорах при організації підпрограм і циклів [1].

6.1. Послідовність виконання лабораторної роботи

1. Вивчити з використанням літературних джерел і матеріалів методичних вказівок функціональні схеми і алгоритми роботи блоків пам'яті типу **FIFO** і **LIFO**.
2. На основі вихідних даних (табл.6.1 – табл.6.4) розробити функціональну електричну схему заданого блоку пам'яті. Привести призначення всіх сигналів розробленого блоку і дати короткі пояснення його функціонування.
3. Розробити часову діаграму блоку пам'яті в режимах запису і читування даних.
4. Розробити і налагодити на ПЕОМ програмну модель заданого блоку пам'яті.
5. Розрахувати часові параметри блоку пам'яті.

6.2. Варіанти індивідуальних завдань

Таблиця 6.1

Варіант = (N) _{мз}	Тип блоку пам'яті
0	FIFO
1	LIFO (1-й варіант)
2	LIFO (2-й варіант)

Таблиця 6.2

Варіант = (N) _{мс}	Організація пам'яті (N × n)
0	32×8
1	64×16
2	128×32
3	256×64
4	512×16

Таблиця 6.3

Варіант = (N) _{m4}	Тип тригера
0	K155TM7
1	K155TM5
2	K155TB1
3	K155TM2

Таблиця 6.4

Варіант = (N) _{m4}	Тип логічного елемента
0	K155ЛР1
1	K155ЛА3
2	K155ЛА4
3	K155ЛА6

6.3. Методичні вказівки

Блок пам'яті типу черга являє собою лінійний список, доступ до елементів якого здійснюється за принципом **FIFO** (first in, first-out). Таким чином, першим з черги віддається елемент, який був поміщений туди першим, потім – елемент, поміщений у чергу другим і так далі.

Функціональна схема буфера **FIFO** наведена на рис.6.1.а, де введені наступні позначення:

- ЛЧХВ** – лічильник «хвоста» черги;
- ЛЧГОЛ** – лічильник «голови» черги;
- RG** – сукупність регістрів буфера;
- К** – компаратор (пристрою порівняння);
- КШЧ** – кодова шина числа.

Черговість звертання до **FIFO** у режимах запису і зчитування встановлюється за допомогою лічильників **ЛЧХВ** і **ЛЧГОЛ**. Адреса запису в буфер задається за допомогою лічильника **ЛЧХВ**. По сигналу **ЗП** дані з **КШЧ** надходять у той регістр буфера, номер якого зазначений у **ЛЧХВ**.

По зрізу сигналу **ЗП** код **ЛЧХВ** збільшується на одиницю. Цім підготовлюється адреса для запису наступного коду.

При надходженні сигналу витягу з черги (**ЗЧ**) на **КШУ** виставляється вміст того регістра, номер якого зазначений у лічильнику **ЛЧГОЛ**.

Переповнення лічильника «хвоста» черги настають при зміні стану лічильника з коду всі одиниці в код усі нулі. Черга пересунеться через нульову оцінку. У такий же засіб згодом «переповзе» і «голова» черги.

Перед початком роботи обидва лічильники за сигналом скидання **СКД** встановлюються в нульовий стан.

У процесі роботи буфера можливі дві ситуації: буфер повний (у нього не можливо більше записувати) і буфер порожній (з нього не можливо вибирати дані). Обидві ці ситуації мають загальну ознаку: рівність вмісту лічильників після зникнення вхідного сигналу **ЗП** (або **ЗЧ**). Ознака рівності встановлюється компаратором **К**. Якщо рівність наступила після чергового витягу з черги (**ЗЧ** = 1), то черга висохнула. Характер останнього звертання до буфера запам'ятовується на **RS** – тригері.

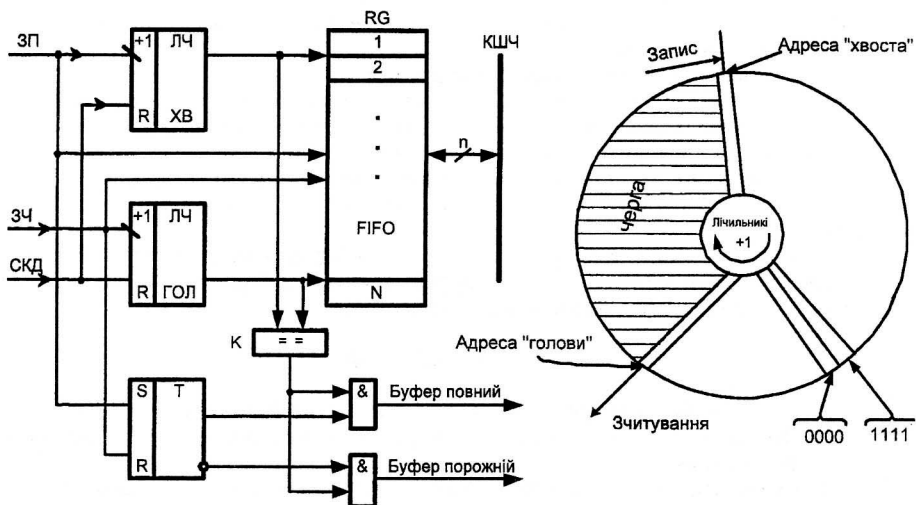


Рис 6.1. Функціональна схема буфера FIFO (а) і діаграма його роботи (б)

Буфер типу стек за змістом протилежний черзі, тому що використовує протилежний метод доступу - LIFO (last-in, first-out). Можливі два варіанти побудови буфера даного типу. У першому варіанті (рис.6.2.) буфер має N регістрів зсування даних, які розташовані вертикально. Керуючими сигналами ЗП і ЗЧ інформація в регістрах підлягає зсуву відповідно вниз або нагору.

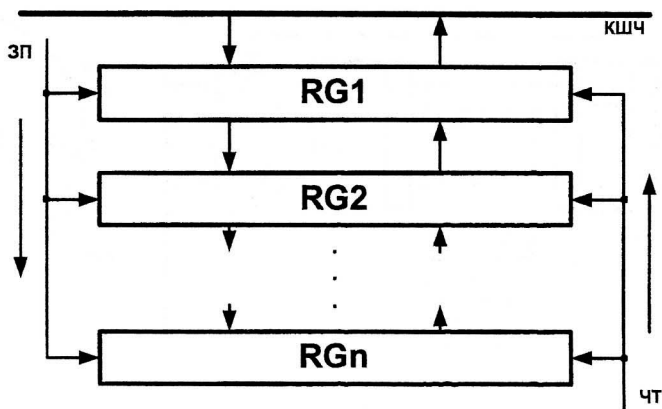


Рис. 6.2. Функціональна схема буфера типу LIFO (перший варіант)

Запис нового слова з КШЧ виробляється в комірку з номером один. При цьому номери усіх раніше записаних слів збільшуються на одиницю (зсув вниз).

Зчитувати можливо тільки слово, яке розташоване у верхньому регістрі буфера. Після зчитування вміст цієї комірки губиться, а номери всіх інших слів зменшуються на одиницю (зсув нагору).

Функціональна схема буфера **LIFO** по другому варіанту наведена на рис.6.3.а, діаграма роботи - на рис.6.3.б. При кожній записі в стек уміст лічильника **ЛЧ** збільшується на одиницю, після кожного зчитування-зменшується на одиницю. Адреса, по якій відбувається записання інформації в стек, завжди більше на одиницю адреси, по якій здійснюється витяг із стека. Зсув на одиницю адреси запису щодо адреси зчитування виконується за рахунок спрацьовування лічильників по різних фронтах вхідних сигналів.

Якщо лічильник знаходиться в стані 00.....0 - буфер порожній (зчитування даних заборонено), якщо в стані 11.....1 - буфер заповнений (заборонено виконувати запис даних). Ці ситуації відслідковує схема аналізу (**СА**).

6.4. Зміст звіту

1. Функціональна електрична схема і діаграма розробленого блоку пам'яті.
2. Листінг програмної моделі з результатами моделювання і короткими коментарями.
3. Часова діаграма роботи блоку пам'яті в режимах запису і зчитування даних.
4. Розрахунок часових параметрів буфера пам'яті.

6.5. Контрольні запитання

1. При якій умові в буфер типу **FIFO** не можливо записувати нову інформацію?
2. Чому запис і зчитування даних у буфер типу **FIFO** по другому варіанту виконується по різним фронтам командних сигналів?
3. Приведіть порівняльну оцінку буферів типу **FIFO** і **LIFO** по швидкодії і витратам обладнання.

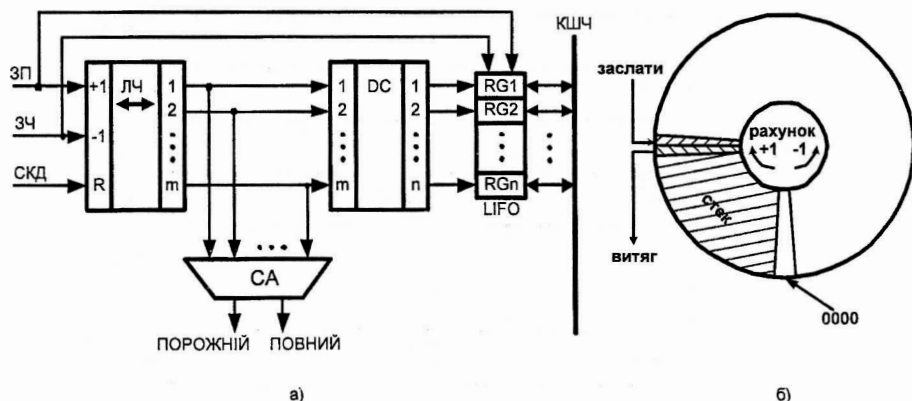


Рис. 6.3. Функціональна схема буфера **LIFO** по другому варіанту (а) і діаграма його роботи (б)

Література

1. Потемкин И.С. Функциональные узлы цифровой автоматики. - М.: Радио и связь, 1989.

Лабораторна робота №7

БУФЕРНІ ЗАПАМ'ЯТОВУЮЧІ ПРИСТРОЇ

МЕТА РОБОТИ: аналіз алгоритмів функціонування буферної пам'яті, а також розробка функціональної електричної схеми і програмної моделі пам'яті.

ЗАГАЛЬНІ ПОЛОЖЕННЯ

Буфер FIFO являє собою ЗП для збереження черг даних (списків) з порядком вибірки слів таким же, як і їх надходження. Інтервали між надходженнями слів можуть бути зовсім різними, тому що моменти запису слова в буфер і зчитування з нього задаються зовнішніми сигналами керування незалежно друг від друга [3, 4].

Можливість мати різний темп прийому і видачі слів необхідна, якщо приймач здатний приймати дані з деякою частотою, а джерело інформації видає слова в більш швидкому темпі, і може бути, до тому ж не регулярно. Такі дані надходять у їхньому темпі проходження в буфер FIFO, а з нього зчитуються регулярно з необхідної для приймача даних частотою.

7.1. Послідовність виконання лабораторної роботи

1. Вивчити з використанням літературних джерел і матеріалів методичних вказівок алгоритми роботи буферної пам'яті (БП).
2. На основі вихідних даних (табл.7.1. – табл.7.5.) розробити функціональну електричну схему буфера, а також горизонтального і вертикального розширень **FIFO**. Дати призначення всіх сигналів розробленої схеми і короткі пояснення її функціонування.
3. Розробити блок-схему алгоритму функціонування буфера і його програмну модель (мовою асемблера або C++).
4. Розробити часову діаграму передачі одного слова даних між двома сусідніми комітками буферної пам'яті.
5. Розрахувати час передачі 100 слів даних через буфер від підсистеми 1 до підсистеми 2.

7.2. Варіанти індивідуальних завдань

Таблиця 7.1.

Варіант = $(N)_{\text{мб}}$	Розрядність даних
0	2
1	4
2	6
3	8
4	10
5	12

Таблиця 7.2.

Варіант = $(N)_{\text{мб}}$	Кількість комірок у буфері
0	8
1	16
2	32
3	64

Таблиця 7.3.

Варіант = $(N)_{m4}$	Тип тригера
0	K155TM2
1	K155TB1
2	K155TM5
3	K155TM7

Таблиця 7.4

Варіант = $(N)_{m3}$	Тип логічного елемента
0	K155ЛА3
1	K155ЛА4
2	K155ЛА2

Таблиця 7.5

Варіант = $(N)_{m6}$	Частота f_1 , МГц	Частота f_2 , МГц
0	1	4
1	2	8
2	4	6
3	5	8
4	6	10
5	8	10

7.3. Методичні вказівки

Розглянемо взаємодію двох підсистем з різною швидкістю передачі даних. Обмін даними між такими підсистемами організується з використанням буферної пам'яті (рис.7.1).

Дані від підсистеми 1 тимчасово запам'ятовуються в буферній пам'яті до готовності підсистеми 2 сприйняти їх.

Відмінною рисою БП є виконання запису даних зі швидкодією і під керуванням підсистеми 1, а виконання зчитування – зі швидкодією і під керуванням підсистеми 2 («еластична» пам'ять). БП виконує операції запису і зчитування незалежно і навіть одночасно. Це усуває необхідність у синхронізації між підсистемами. БП повинна зберігати порядок надходження даних від підсистеми 1, тобто працювати за принципом FIFO – «перше записане слово зчитується також першим».



Рис 7.1. Структура розглянутої системи

Розглянемо принцип дії буфера FIFO організації 64 комірки по 4 розряди кожна (рис.7.2).

Буфер складається з 64 4-бітних регістрів з незалежними ланцюгами зсування, 64-бітового керуючого регістра (PrK), а також схеми керування. Вхідні дані надходять на входи $DI_0 \div DI_3$. Запис даних виробляється керуючим сигналом ЗП, а зчитування – сигналом вивіду ЗЧ.

Уведення даних у буфер можливо здійснити тільки при наявності сигналу готовності даних ГОТ₁, а виведення – при наявності сигналу готовності ГОТ₀. Керуючий сигнал СКД робить скидання буфера FIFO.

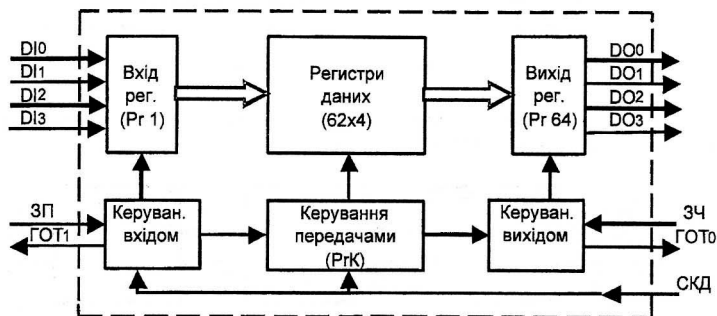


Рис. 7.2. Структура буфера FIFO

При введенні 4-бітного слова (по сигналу ЗП від підсистеми 1) під впливом внутрішніх сигналів автоматично пересуваються в найближчий до виходу порожній (не утримуючих даних) регістр. Стан регістра даних відображається у відповідному йому керуючому тригері, сукупність яких утворюють 64-бітний регістр керування PrK. Якщо i -й регістр даних містить дані, керуючий тригер знаходиться в стані логічної "1", якщо даних немає – "0". Слово даних не можливо завантажувати в зайнятий регістр, але як тільки керуючий біт сусіднього праворуч регістра зміниться на "0", слово даних автоматично зсувається до виходу.

Перед початком роботи в FIFO подається сигнал скидання СКД, і всі керуючі тригери переводяться в "0" – усі регістри вільні. Сигнал готовності входу ГОТ₁ фактично є інверсним вихідним керуючого тригера Pr1. Тому, коли цей тригер скинутий, на вихіді ГОТ₁ "1", тобто FIFO готовий сприймати вхідні дані. При дії сигналу ЗП вхідне слово завантажується в Pr1, а керуючий тригер цього регістра встановлюється в "1" (на лінії ГОТ₁ "0").

Якщо керуючий тригер Pri містить "0", а сусідній ліворуч керуючий тригер Pr(i-1) знаходиться в стані "1", дані з Pr(i-1) завантажуються в Pri. При цьому тригер Pr(i-1) скидається в "0", а керуючий тригер Pri встановлюється в "1". Така схема керування приводить до того, що слово, яке надійшло в Pr1, "спонтанно" копіюється у всіх регістрах буфера і з'являється на вихідних лініях D0.

Стан керуючого тригера Pr64 виведено на лінію готовності виходу ГОТ₀. Тому коли в цей тригер записується 1, з'являється сигнал готовності, який сигналізує про наявність даних FIFO.

Як тільки дані з Pr1 передаються в Pr2, керуючий тригер Pr1 скидається в "0". Це приводить до появи сигналу ГОТ₁ (можливо записувати нові дані).

При подачі сигналу ЗЧ (від підсистеми 2) відбувається зчитування даних з ліній D0, а керуючий тригер Pr64 переводиться в стан "0". Це створює новий порожній регістр даних. Тому, уміст Pr63 автоматично копіюється в Pr64, керуючий тригер переводиться в стан "1", на лінії ГОТ₀ з'являється "1". Цей процес повторюється для всіх інших регістрів (див. рис.7.3).

Збільшити кількість збережених у буфері слів можливо з використанням горизонтального розширення FIFO (рис.7.4).

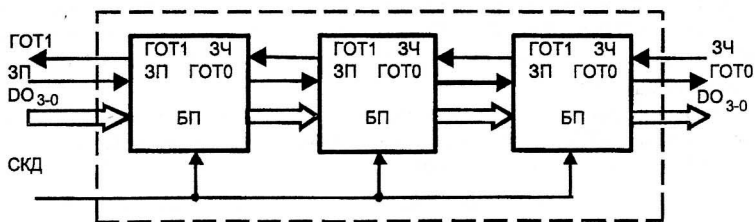


Рис. 7.4. Горизонтальне розширення **FIFO**

Збільшення розрядності оброблюваних слів можливо досягти з використанням вертикального розширення **FIFO** (рис. 7.5).

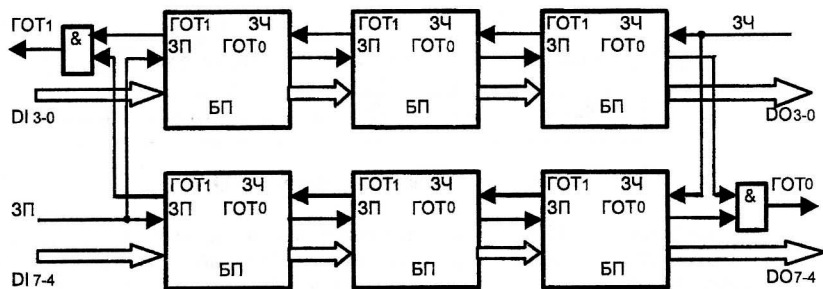


Рис.7.5. Вертикальне розширення **FIFO**

7.4. Зміст звіту

1. Функціональні електричні схеми буфера, а також горизонтального і вертикального розширення **FIFO**.
2. Блок – схема алгоритму роботи буферної пам'яті. Листінг програми з результатами роботи і короткими коментарями.
3. Часова діаграма передачі одного слова даних.
4. Розрахунок часу передачі 100 слів даних від підсистеми 1 через буфер в підсистему 2.

7.5. Контрольні запитання

1. Розробіть блок-схему алгоритму функціонування буферної пам'яті в режимах запису і зчитування інформації.
2. Для чого в структуру буферної пам'яті введений регістр керування?

Література

1. Шило В.Л. Популярные цифровые микросхемы. Справочник. – М.: Радио и связь, 1988. – 352с.
2. Шилдт Г. Теория и практика C++. Пер. с англ. – СПб.: BHV- Санкт-Петербург, 1996. – 416с.
3. Угрюмов Е. Цифровая схемотехника. - СПб.: - БХВ - Петербург, 2004. - 528 с.
4. Григорьев В.Л. Микропроцессорные системы (аппаратные средства) : Учебное пособие. – Рязань, РРИ, 1980. – 80с.

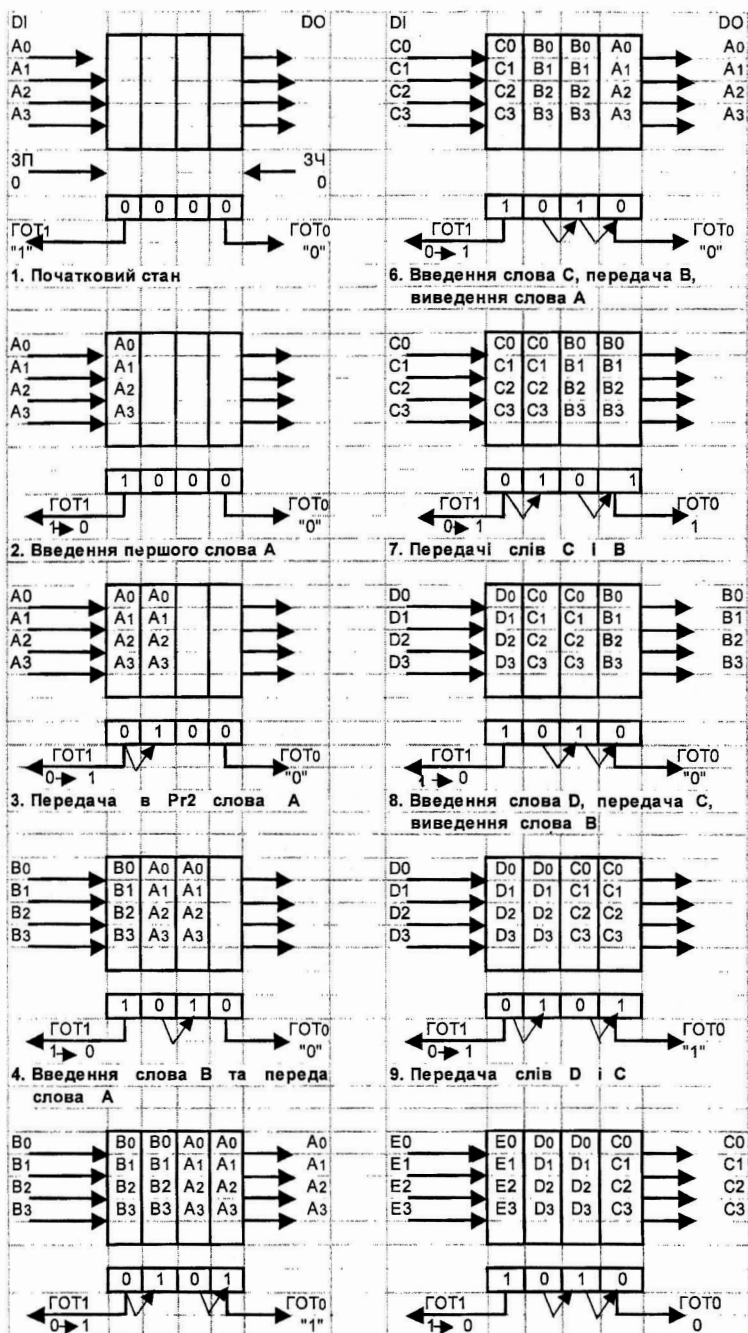


Рис 7.3. Приклад роботи буферної пам'яті

Лабораторна робота № 8

АНАЛІЗ АЛГОРИТМІВ І ПРОЕКТУВАННЯ МОДУЛІВ КЕШ-ПАМ'ЯТІ

МЕТА РОБОТИ: вивчення алгоритмів функціонування кеш-пам'яті сучасних комп'ютерів і одержання навичок розробки її основних блоків.

ЗАГАЛЬНІ ПОЛОЖЕННЯ

Ефективність кешування обумовлена тим, що більшість прикладних програм мають циклічний характер і багаторазово використовують ті самі дані. Тому після першого використання даних з відносно повільної основної пам'яті повторні звертання вимагають менше часу. До того ж при використанні процесором кеш-пам'яті основна пам'ять звільняється від роботи і тому можуть виконуватися цикли регенерації даних у динамічному ЗП [1 - 5].

8.1. Послідовність виконання лабораторної роботи

1. Вивчити з використанням літературних джерел і матеріалів методичних вказівок склад і функціонування кеш-пам'яті.
2. Розробити графу-схему алгоритма функціонування системи з кеш-пам'яттю, процесором і динамічною пам'яттю.
3. Синтезувати і привести функціональну схему керуючого автомата для кеш-пам'яті заданої організації.
4. Привести формули і виконати розрахунок часу виконання заданого алгоритму (скласти програму мовою асемблера) з кеш-пам'яттю і без неї.

8.2. Варіанти індивідуальних завдань

Таблиця 8.1

Варіант= $(N)_{m_2}$	Кількість блоків кеш-пам'яті
0	2
1	4

Таблиця 8.2

Варіант= $(N)_{m_3}$	Організація одного блоку кеш-пам'яті
0	$256 \times (32 + \text{Tag} + V)$
1	$512 \times (32 + \text{Tag} + V)$
2	$1024 \times (32 + \text{Tag} + V)$

Таблиця 8.3

Варіант= $(N)_{m_4}$	Організація DRAM
0	$1M \times 32$
1	$2M \times 32$
2	$4M \times 32$
3	$8M \times 32$

Таблиця 8.4

Варіант= $(N)_{m4}$	Тип керуючого автомата
0	Милі
1	П – природ.
2	Мура
3	П – примус.

Таблиця 8.5

Варіант= $(N)_{m8}$	Алгоритм множення
0	А ПК
1	Б ДК
2	В ПК
3	Г ДК
4	А ДК
5	Б ПК
6	В ДК
7	Г ПК

8.3. Методичні вказівки

Приклад структури системи з кеш-пам'яттю наведена на рис. 8.1 [3]. До її складу входять процесор CPU, динамічне ОЗП (**DRAM**) організації $0,5\text{M} \times 32$ бітів з часом доступу 70 нс, кеш - пам'ять організації 256×91 біт з часом доступу 7 нс та мультіплексор **MX**. Звичайно кеш-пам'ять виконують на елементах статичного типу (**SRAM**), у яких немає необхідності в регенерації інформації.

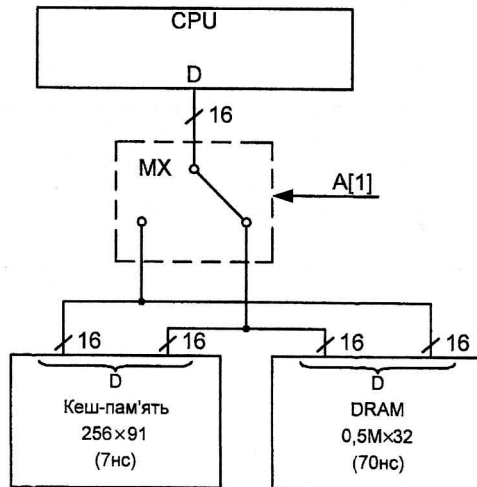


Рис. 8.1. Структура системи з кеш-пам'яттю

Спочатку інформація зберігається в **DRAM**, швидкодія якої в 10 разів менше швидкодії статистичної кеш-пам'яті. Відразу після включення напруги харчування або

після роботи каналу прямого доступу в пам'ять, у кеш-пам'яті зберігається недостовірна інформація (ознаки **V** всіх комірок встановлені в нульовий стан).

При виконанні програми **CPU** здійснює зчитування з **DRAM** команд і даних, які попутно записуються також в кеш-пам'ять. Причому записується інформація з адресуємої і сусідній з нею комірок **DRAM**. Таким чином, у кеш-пам'яті накоплюються копії збережених у **DRAM** слів. Зчитування цих слів відбувається в 10 разів швидше ніж зчитування слів - оригіналів.

Умови збереження і відновлення інформації в комітках **DRAM** і кеш-пам'яті наведені в табл. 8.6.

Таблиця 8.6

Режим роботи	Наявність копії комірки DRAM в кеш-пам'яті	Інформація	
		У комірці кеш-пам'яті	У комірці DRAM
Зчитування	Копія є	Не змінюється	Не змінюється
	Копії немає	Обновляється	Не змінюється
Запис	Копія є	Обновляється	Обновляється
	Копії немає	Не змінюється	Обновляється

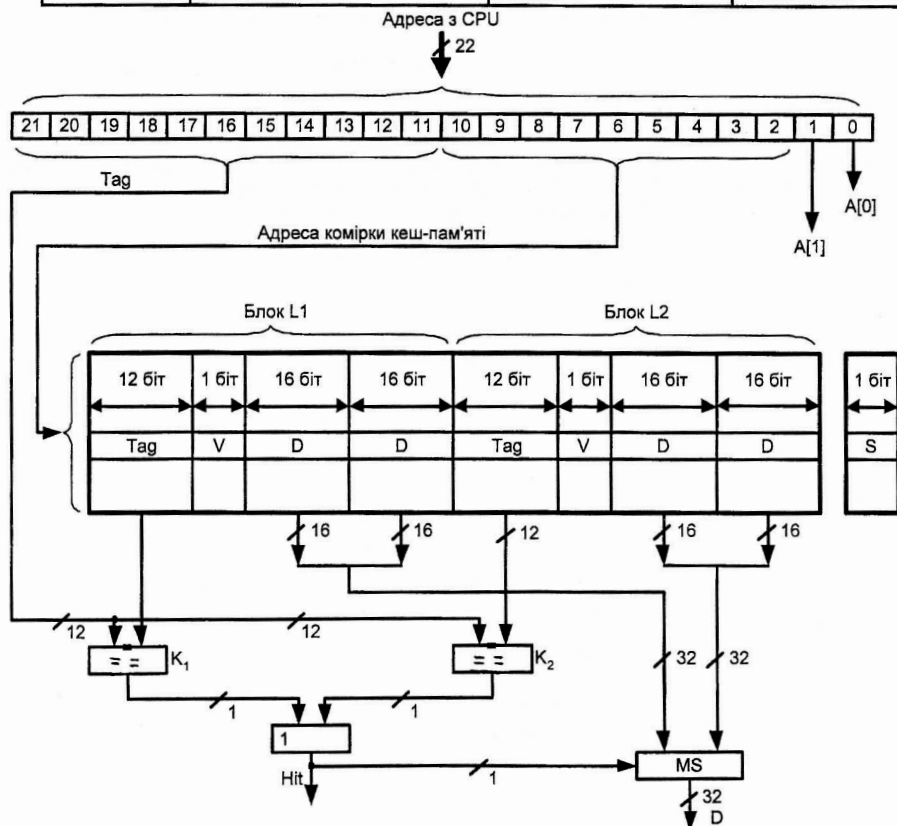


Рис. 8.2. Структура кеш - пам'яті

При записі інформації в комірку **DRAM** і наявності копії цієї комірки в кеш-пам'яті оригінал і копія обновляються (третій рядок табл. 8.6). Сусідні 16-розрядні комірки як у **DRAM**, так і кеш-пам'яті не змінюються і зберігають стару інформацію. Якщо копії в кеш-пам'яті немає, обновляється тільки комірка **DRAM**. Тобто копія в кеш-пам'яті в цьому випадку не створюється (четвертий рядок табл. 8.6).

Таким чином, кеш-пам'ять прискорює зчитування команд і даних, які недавно зчитувалися або коректувалися (перший рядок табл. 8.5). В інших ситуаціях вигравшу в часі немає. При виконанні програм процесором операції запису в **DRAM** складають 10%, а операції зчитування – 90% від загальної кількості звертань до пам'яті [4]. Тому використання кеш-пам'яті може істотно підвищити продуктивність системи.

У прикладі, наведеному на рис. 8.2, кеш-пам'ять містить два однакових блоки **SRAM L1** і **L2** організацією 256 комірок по 45 розрядів кожний і пам'ять ознак **S** недавнього використання блоків організацією 256×1 розряд [3].

У кожному блоці **SRAM** є розряди для збереження коду **Tag** (ознака), який відбиває старшу частину адреси комірки-оригіналу, розряди **D**, у яких міститься копія 32-розрядної комірки **DRAM** і ознака **V** вірогідності інформації в комірці блока.

Формат адреси, яка надходить з **CPU**, має вигляд, наведений на рис. 8.3.

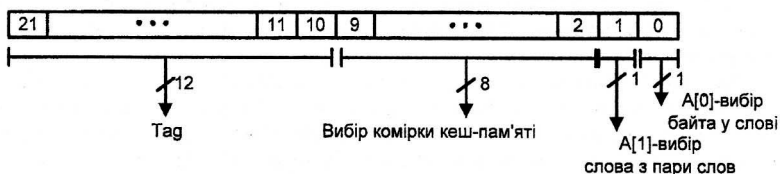


Рис. 8.3. Формат адреси кеш-пам'яті

Розряд **A[0]** адреси дозволяє вибрати байт у 16-розрядному слові. Якщо в цьому розряді нуль, вибирається молодший байт, якщо одиниця – старший байт. При роботі з 16-розрядними словами в розряді **A[0]** записано 0, тобто використовуються тільки парні адреси.

Розряд **A[1]** дозволяє вибрати тільки одне з двох 16-розрядних слів у 32-розрядних комірках **DRAM** і кеш-пам'яті. Якщо в цьому розряді присутній нуль, вибирається праве, якщо одиниця – ліве 16-розрядне слово.

Вісім розрядів **A[9] ÷ A[2]** визначають адресу однієї з 256 81-розрядних комірок кеш-пам'яті.

Старші розряди адреси **A[21] ÷ A[10]** при зчитуванні інформації з кеш-пам'яті порівнюються з відповідними розрядами коду ознаки **Tag** з обраної комірки першого і другого блоків кеш-пам'яті. При записі нової інформації в перший або другий блок кеш-пам'яті розряди **A[21] ÷ A[10]** адреси **CPU** запам'ятовуються як ознаки нового коду в полі **Tag** комірки **SRAM**.

Алгоритми роботи кеш-пам'яті

Нехай, наприклад, **CPU** установив на системній шині адресу деякої комірки **DRAM** і виставив командний сигнал читання **MERM#**. Модуль кеш-пам'яті, одержавши цю адресу, виконує наступні дії.

Розряд **A[1]** набуває мультіплексор **MX** (рис. 8.1) на підключення 16-розрядної шини даних **D CPU** до однієї з половин 32-розрядної шини даних **DRAM** і кеш-пам'яті. На виходах кеш-пам'яті з'являється інформація з обох блоків **SRAM**: **TAG**, **V**, **D** і **S**.

За допомогою двох компараторів K_1 і K_2 порівнюються коди ознак **Tag** надійшовших з **CPU** адреси і лічених з першого і другого блоків **SRAM**. Якщо ні компаратор K_1 , ні компаратор K_2 не знайшли збіг ознак, це означає відсутність копії викликаного комірки **DRAM** у кеш-пам'яті.

Якщо відбувся збіг кодів на входах одного з компараторів, формується сигнал **Hit** = 1 і через мультіплексор **MX** на 32-розрядну шину даних надсилаються два слова з відповідного блока кеш-пам'яті. Копія інформації створюється лише в одному екземплярі, тому одночасно обидва компаратори не можуть зафіксувати порівняння кодів.

Процесор одержує необхідні дані, а розряд **S** кеш-пам'яті встановлюється в 0 або 1 у залежності від того, з якого блоку була видана інформація. Цей розряд визначає черговість звертання до блоків (яка інформація більш «свіжа»). Нульовий розряд **A[0]** при зчитуванні даних не має значення.

У **CPU** посилається два байти, і він вибере той з них, який йому потрібний. Звертання до **DRAM** при **Hit** = 0 немає.

Якщо **Hit** = 0, здійснюється вибір з **DRAM** 32-розрядного слова за адресою $A[21] \div A[2]$ (одночасно витягається необхідне 16-розрядне слово і сусіднє з ним). Лічене з **DRAM** 32-розрядне слово записується в комірку кеш-пам'яті за адресою $A[9] \div A[2]$. Блок **L1** або **L2** визначається шляхом аналізу розряду **S** обраної комірки кеш-пам'яті. Більш «застаріла» інформація замінюється новою, а вміст розряду **S** інвертується.

При записі 32-розрядного коду в комірку першого або другого блоків кеш-пам'яті активізуються обидва нагромаджувачі слів **D** незалежно від значення розряду **A[1]**. У розрядах коду **Tag** запам'ятовується група розрядів $A[21] \div A[10]$ адреси **CPU**, а ознака вірогідності інформації **V** встановлюється в 1. Праве або ліве 16-розрядне слово ліченого з **DRAM** 32-розрядного коду (у відповідності зі значенням розряду **A[1]**) посилається в **CPU**. На цьому операція зчитування даних з побіжним створенням нової копії в кеш-пам'яті завершується.

У кеш-пам'яті з чотирма блоками (**L1, L2, L3** і **L4**) кожному блоку відповідають визначені значення бітів **S0, S1** і **S2**, які модифікуються при кожному влученні і заповненні інформацією кеш-пам'яті [4]:

- якщо останнє звертання в комірку кеш-пам'яті було до блоку **L1** або **L2**, то біт **S0** встановлюється в 1, а при звертанні до блоків **L2** або **L3** – у 0;
- якщо останнє звертання в комірку в парі блоків **L1, L2** було до блоку **L1**, то біт **S1** встановлюється в 1, а при звертанні до блоку **L2** – у 0;
- якщо останнє звертання в парі **L3, L4** було до блоку **L3**, то біт **S2** зводиться в 1, а при звертанні до блоку **L4** біт **S2** скидається в 0.

Вибір замінного блоку (коли всі блоки в комірку достовірні) визначається згідно табл.8.7.

Таблиця 8.7

S0	S1	S2	Замінний блок
0	0	*	L1
0	1	*	L2
1	*	0	L3
1	*	1	L4

8.4. Зміст звіту

1. Функціональна електрична схема системи і кеш-пам'яті з коротким описом призначення всіх блоків і сигналів.

2. Граф-схема алгоритму функціонування системи з кеш-пам'яттю.
3. Функціональна схема синтезованого керуючого автомата.
4. Програма (мовою асемблера) заданого алгоритму.
5. Розрахунок продуктивності системи для заданого алгоритму за рахунок використання кеш - пам'яті.

8.5. Контрольні запитання

1. За рахунок чого використання кеш-пам'яті приводить до підвищення продуктивності роботи комп'ютера? Як оцінити цю продуктивність?
2. Чим відрізняються структури та алгоритми роботи кеш-пам'яті з асоціативним доступом і прямим відображенням?
3. Розробити блок-схему алгоритму функціонування кеш-пам'яті в режимах запису і зчитування інформації.

Література

1. Угрюмов Е. Цифровая схемотехника. - СПб.: БХВ - Петербург, 2004. - 528 с.
2. Гук М. Аппаратные средства IBM PC. Энциклопедия. – СПб.: «Питер», 2000. – 816с.
3. Айден К. и др. Аппаратные средства PC. – СПб.: ВHV, С. – Петербург, 1998. – 608с.
4. Шевкопляс Б.В. Микропроцессорные структуры. Инженерные решения: Справочник – М.: Радио и связь, 1990. – 512с.
5. Григорьев В.Л. Микропроцессор i486. Архитектура и программирование (в 4-х книгах) – М.: ГРАНАЛ, 1993 г.

ДОДАТОК

Д1. Опис тестів функціонального контролю ОЗП

Нижче наведені умовні позначки, які використовуються при описі роботи тестів модуля ОЗП.

"Контроль" – порівняння зчитуваної інформації з еталоном;

A_i – поточна адреса комірки i ;

A_D – доповнююча адреса ($A_D = A_{N-1} - A_i$);

$[A_i]$ – вміст комірки з адресою A_i ;

T – інформація логічного нуля;

$\bar{T}$ – інформація логічної одиниці.

В структурних схемах алгоритмів тестів елементи матриці пам'яті можуть мати один індекс i , який змінюється від 0 до $(N - 1)$, де A_K – контролююча адреса, або подвійний індекс (i, j) , де індекс i змінюється від 0 до $\sqrt{N}$ (матриця накопичувача – квадратна) – по стріцям матриці, а індекс j – змінюється від 0 до $\sqrt{N}$ – по стовбцям матриці. У цьому випадку A_{SR} – контролюючі адреси пам'яті, де S – змінюється по стріцям, а R – по стовбцям; A_{LM} – кінцева адреса по стрічці L і стовбцю M .

1. Тест "Галоп"

У першу контрольну адресу $A_K = A_0$ записується інформація $\bar{T}$, а в усі інші адреси ($A_i = A_1$), ... , ($A_i = A_{N-1}$) записується інформація T . Після цього послідовно зчитуються і перевіряються дані адрес $A_i = A_1, A_i = A_K, A_i = A_1, A_i = A_2, A_i = A_K, A_i = A_2, A_i = A_3, A_i = A_K, A_i = A_3$ і далі, доки всі пари переходів (включаючи адресу $A_K = A_0$) не будуть перевірені. Після цього в адресу $A_K = A_0$ записується інформація T . Ця послідовність повторюється для адрес $A_K = A_1$ і так далі до $A_K = A_{N-1}$. Після цього виробляється інверсія інформації в контрольній адреси A_K і поточних адресах A_i і цикл перевірки повторюється.

Блок-схема алгоритму теста "Галоп" наведена на рис. Д1.

2. Тест "Марш"

Послідовно по всім адресам виробляється запис фону 0 . Після цього для кожної адреси зчитується інформація T і записується $\bar{T}$ при зміні адрес від A_0 до A_{N-1} . Далі, починаючи з адреси $A_i = A_0$ до $A_i = A_{N-1}$ для кожної комірки зчитується інформація $\bar{T}$ і записується T . Після цього для кожної адреси зчитується інформація T і записується $\bar{T}$ при зміні адрес від A_{N-1} до A_0 (зворотний перебір адрес). Далі для кожної адреси зчитується інформація $\bar{T}$ і записується T при зміні адрес від A_{N-1} до A_0 . Після цього виробляється інверсія фоновий інформації (запис фону одиниці) і цикл повторюється.

Блок-схема алгоритму теста "Марш" наведена на рис. Д2.

3. Тест "Попарна запис-зчитування з повним перебіром"

Послідовно по всім адресам виробляється запис фону 0 . В адресу $A_i = A_1$ записується інформація $\bar{T}$, а в адресу $A_K = A_0$ записується інформація T . Після цього відбувається зчитування інформації з адрес $A_i = A_1$ і $A_K = A_0$. Далі в адреси $A_i = A_1$ і

$A_K = A_0$ записується інформація T з наступним зчитуванням її з цих адрес. Це повторюється для адрес $A_1 = A_2$, $A_1 = A_3$ і далі до $A_1 = A_{N-1}$ з контрольною адресою $A_K = A_0$. Далі цикл перевірки повторюється для адрес $A_K = A_1$, $A_K = A_2$ і далі до $A_K = A_{N-1}$. Після цього послідовно по всім адресам виробляється запис інформації $\bar{T}$ і цикл перевірки повторюється.

Блок-схема алгоритму теста "Попарна запис-зчитування з повним перебіром" наведена на рис.Д3.

4.Тест "Попарне зчитування по стовбцю"

Спочатку в усі стовпці всіх рядків ОЗП записується фон T . Потім по черзі для всіх рядків виробляється послідовно для всіх стовпців рядка запис у поточний стовпець значення $\bar{T}$ та попарне зчитування і контроль вмісту поточного і всіх інших стовпців рядка.

Блок-схема алгоритму теста "Попарне зчитування по стовпцю" наведена на рис.Д4.

5.Тест "Галопуючий адресний код"

Спочатку в матрицю пам'яті записується фонові інформація. Після цього встановлюються адреси поточної комірки A_i і контрольної комірки A_K . Різниця адрес контрольної і поточної комірок змінюється послідовно $[A_K - A_i = 0, 1, 2, \dots, (N-1)]$, причому зміна відбувається циклічно. В комірку з адресою $(A_K + A_i)$ відбувається запис інформації T і зчитування інформації $\bar{T}$. Після цього з комірок з адресою $(A_K - A_i)$ відбувається тільки зчитування інформації T . В кожному наступному циклі відбувається збільшення адреси контрольної комірки на 1. Інформація T змінюється для кожної поточної адреси і для кожного розряду і визначається як сума в двоїчному коді номерів циклу і адреси, значення якої записується в уявлений послідовний циклічний n -розрядний регістр і зчитується з розряду регістру, номер якого відповідає номеру циклу або кратний йому.

Блок-схема алгоритму теста "Галопуючий адресний код" наведена на рис.Д5.

6.Тест "Батерфляй"

В першу контрольну адресу $A_{SR} = A_{00}$ записується інформація $\bar{T}$, а в усі інші адреси (поточні адреси A_{IJ}) записується інформація T . Після цього адреси A_{SR} попарно зчитуються з адресами першого рядка і першого стовпця. Після цього в адресу A_{SR} записується інформація T і потім зчитується по цієї адреси. Далі інформація $\bar{T}$ записується в адресу $A_{SR} = A_{01}$; попарне зчитування адреси A_{01} відбувається з адресами першого рядка і другого стовпця. Ця послідовність здійснюється для всіх адрес A_{SR} від першого до останнього рядка; при цьому попарне зчитування здійснюється для контрольної адреси A_{SR} і поточних адрес A_{IJ} рядка і стовпця, на якому розміщена контрольна адреса A_{SR} . Після цього відбувається запис інверсії інформації в адресу A_{SR} і поточні адреси A_{IJ} і цикл перевірок повторюється.

Блок-схема алгоритму теста "Батерфляй" наведена на рис.Д6.

7. Тест "Хрест"

Послідовно по всім адресам виробляється запис фону 0. Після цього з контрольні адреси $A_{SR} = A_{00}$ зчитується інформація T і записується інформація $\bar{T}$ за адресою A_{IJ} (в сусідню адресу відносно A_{SR} по рядку). Далі зчитується інформація T за адресою $A_{SR} = A_{00}$ і записується інформація $\bar{T}$ за адресою A_{IJ} (в сусідню адресу відносно A_{SR} по стовпцю). Така операція виробляється для адреси $A_{SR} = A_{00}$ зі всіма сусідніми адресами по "хресту" (по стовпцю і рядку). Далі переходять до адреси $A_{SR} = A_{01}$ і виробляють аналогічну перевірку, як і адреси A_{00} . Така перевірка здійснюється для всіх адрес послідовно. Після цього виробляється запис фону 1 по всім адресам і цикл перевірок повторюється.

Блок-схема алгоритму теста "Хрест" наведена на рис.Д7.

8. Тест "Зсув діагоналі"

Послідовно по всім адресам записується інформація T . Потім в адреси центральної діагоналі записується інформація $\bar{T}$. Далі відбувається зчитування інформації з адрес по стовпцям. Аналогічним чином операція повторюється для всіх адресів нецентральної діагоналей, кількість яких дорівнює $(\sqrt{N} - 1)$. Потім відбувається інверсія інформації і цикл перевірок повторюється.

Блок-схема алгоритму теста "Зсув діагоналі" наведена на рис.Д8.

9. Тест "Пінг - понг"

В комірку з адресою A_K (напочатку $K = 0$) записується інформація $\bar{T}$. Далі послідовно по всім адресам A_i (коли $i \neq K$) записується код T . Після цього при $i \neq K$ (в циклі змінювання i відбувається зчитування і контроль інформації з комірок A_i і A_K . Коли $i = (N - 1)$ в комірку A_K записується код T , індекс K інкрементується на 1 і в наступну комірку A_K записується інформація $\bar{T}$. Індекс i встановлюється рівним 0 і цикл перевірок повторюється. Коли i досягає значення $(N - 1)$ відбувається інверсія змінної T і повторення перевірок.

Блок-схема алгоритму теста "Пінг - понг" наведена на рис.Д9.

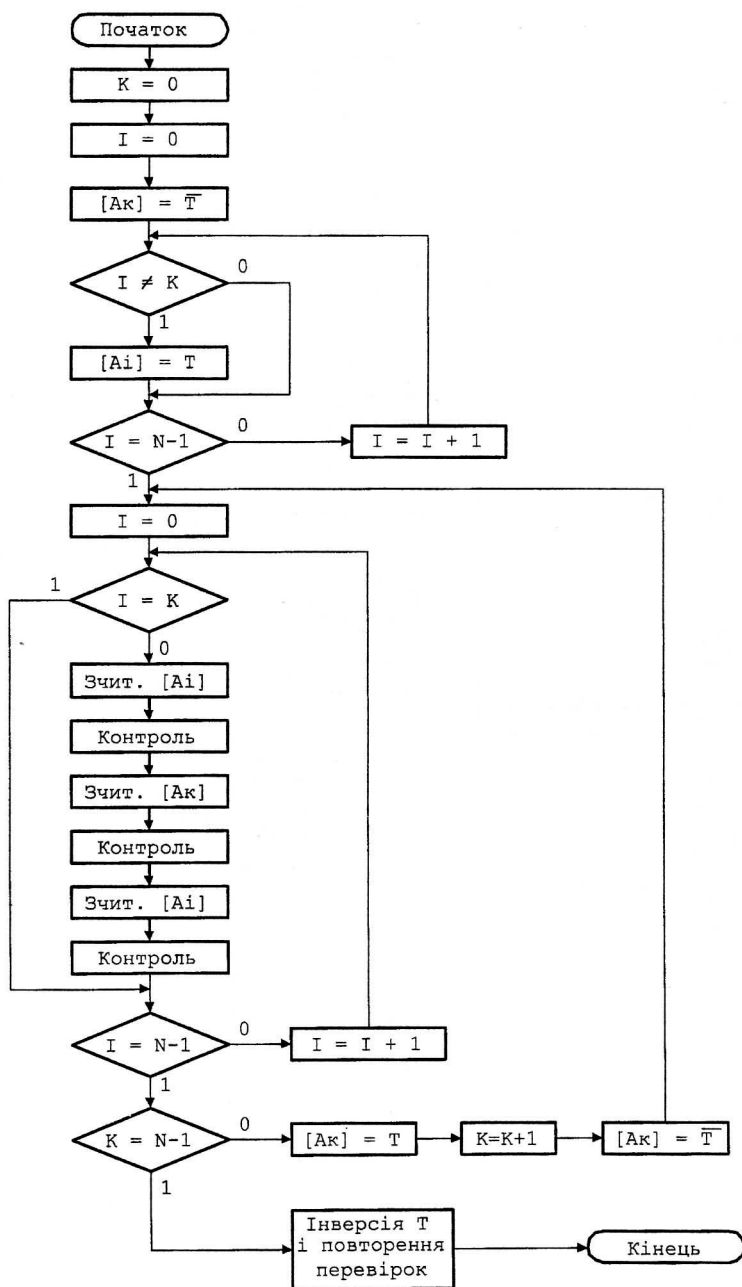


Рис.Д.1. Блок-схема алгоритму теста "Галоп"

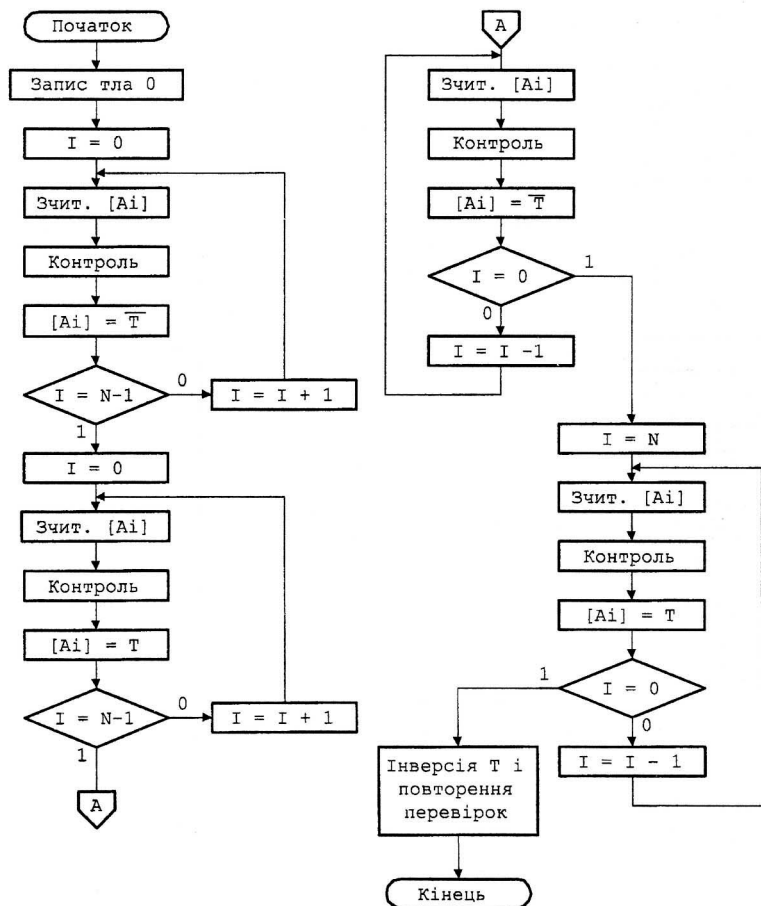
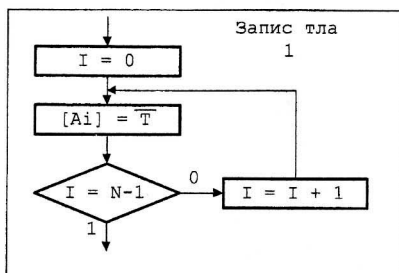
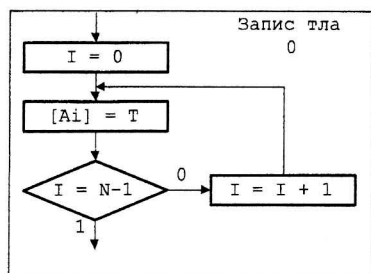


Рис.Д.2. Блок-схема алгоритму теста "Марш"

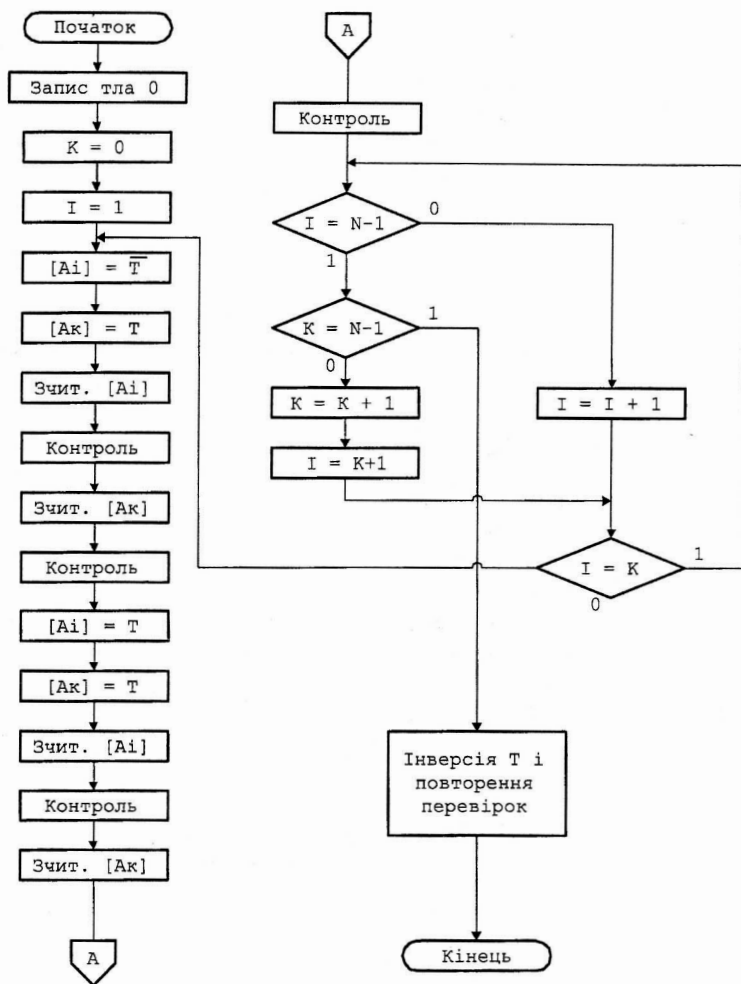
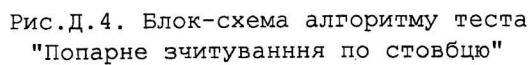


Рис.Д.3. Блок-схема алгоритму теста "Попарне запис - зчитування з повним перебором"



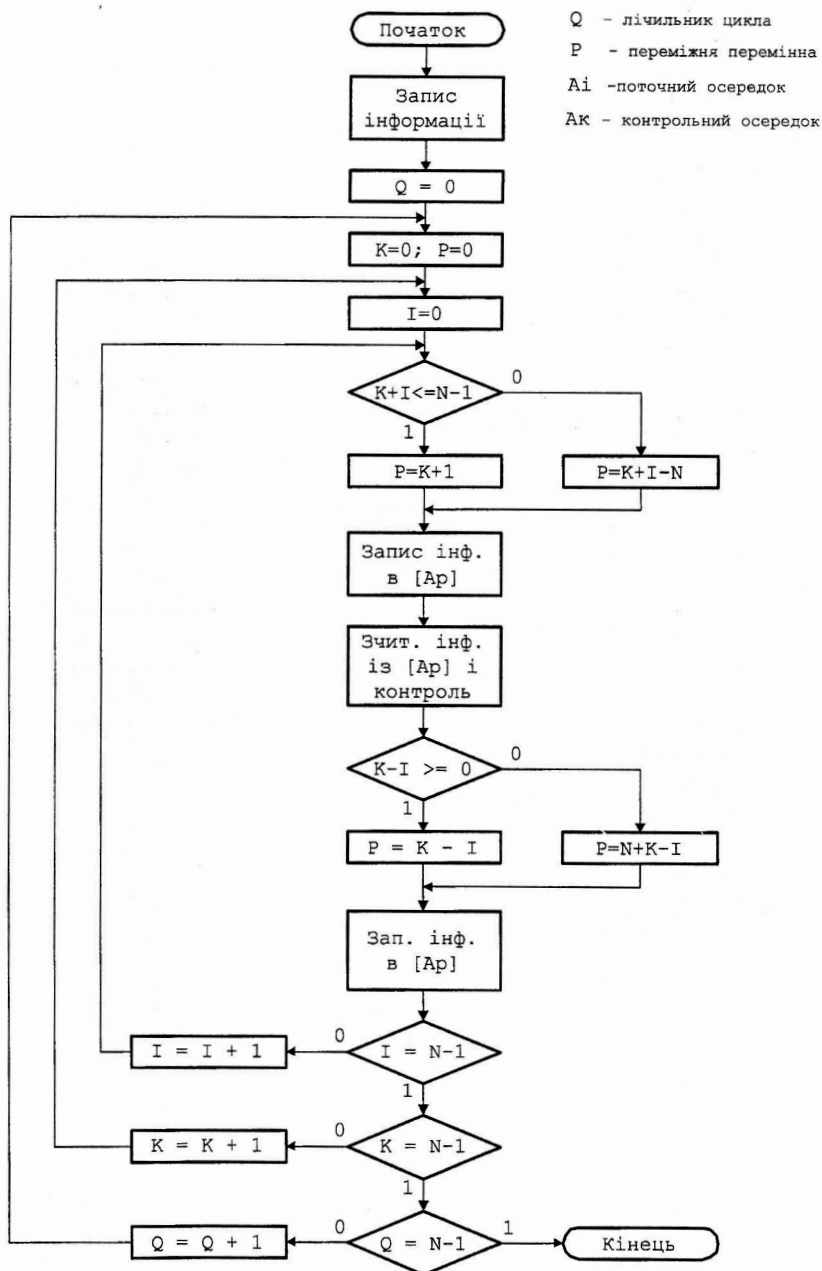


Рис.Д.5. Блок-схема алгоритму теста "Галопуючий адресний код"

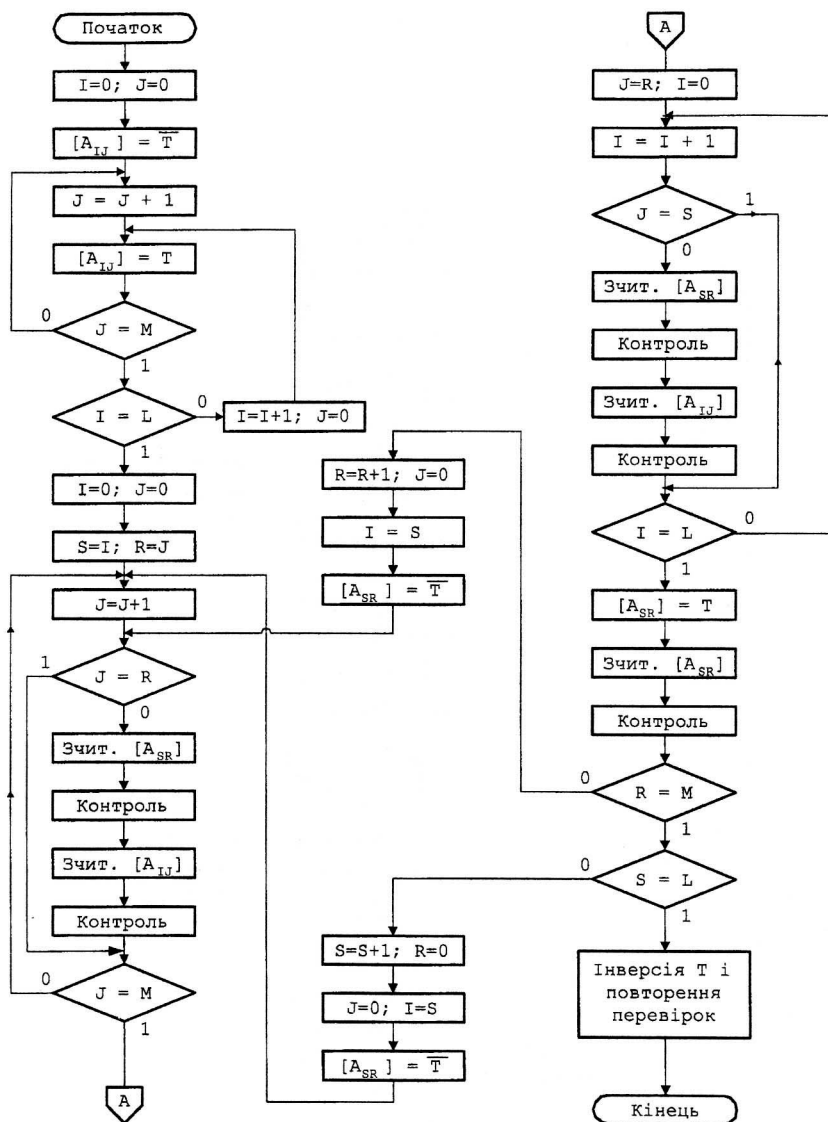


Рис.Д.6. Блок-схема алгоритму теста "Батерфляй"

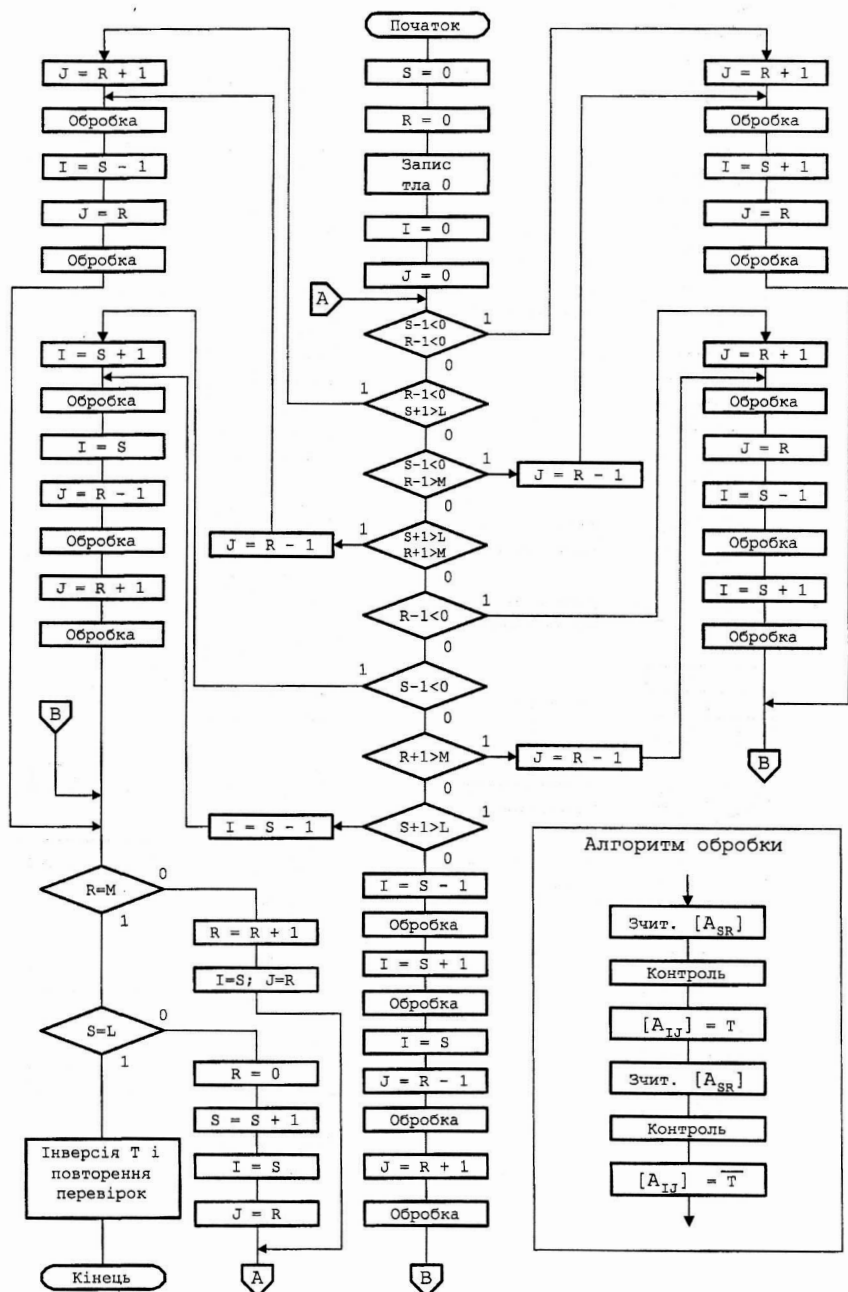


Рис.Д.7. Блок-схема алгоритму теста "Хрест"

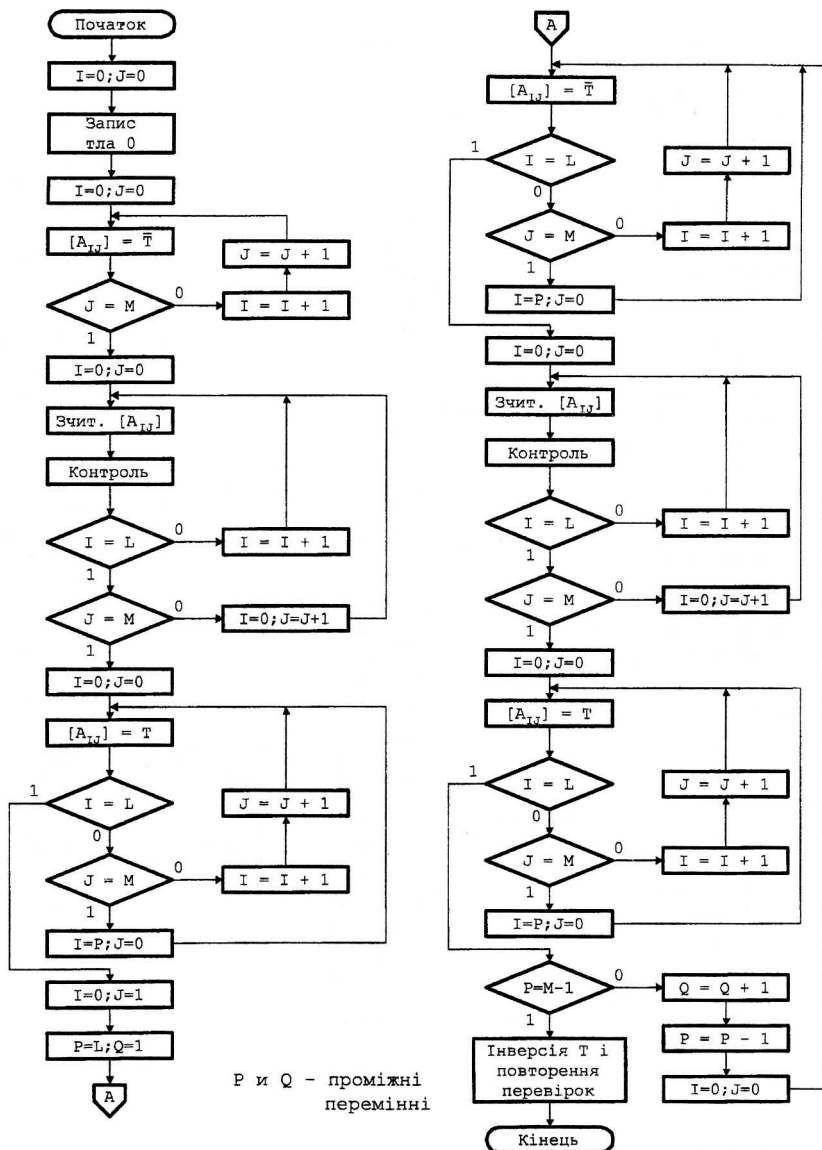


Рис.Д.8. Блок-схема алгоритму теста "Зсув діагоналі"

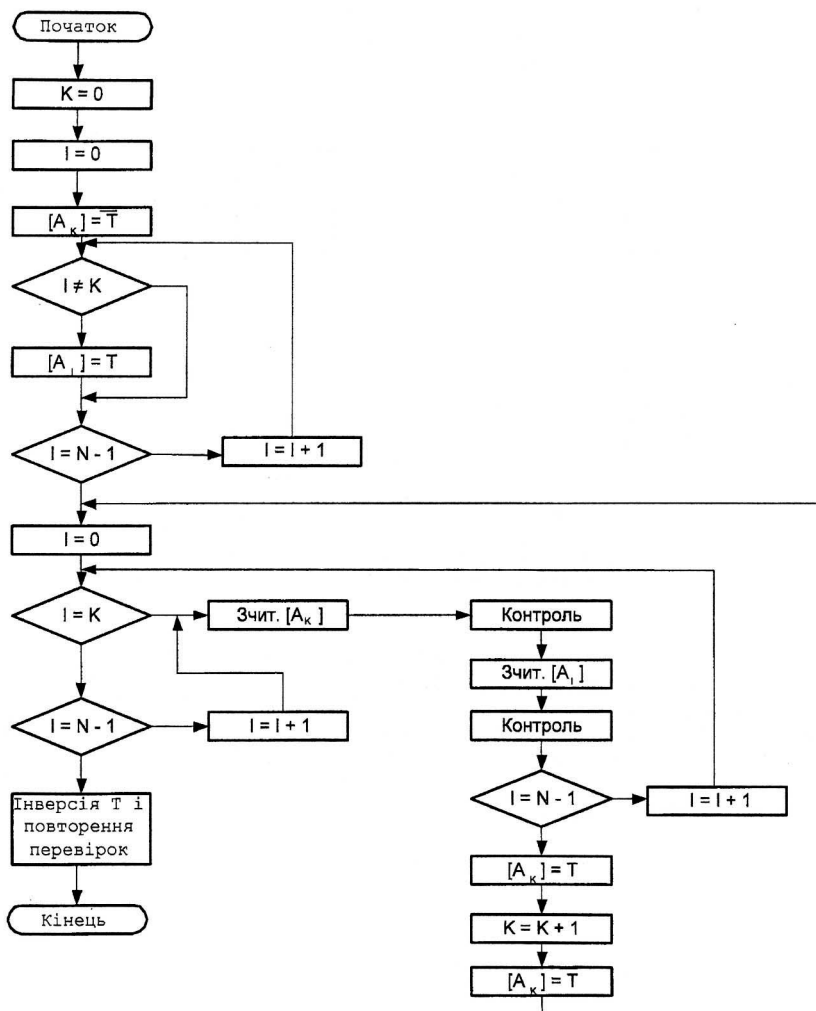


Рис.Д.9. Блок-схема алгоритму теста
"Пінг - понг"

Д2. Схеми вихідних ланцюгів ВІС елементів пам'яті

Вихідні каскади ВІС ЗП дозволяють поєднувати їх за схемою провідного АБО, якщо ВІС ЗП виконаний з відкритим колектором, або трьома станами (рис. Д10). При цьому об'єднанні повинні дотримуватися вимоги технічних умов на ВІС по вихідних навантаженнях.

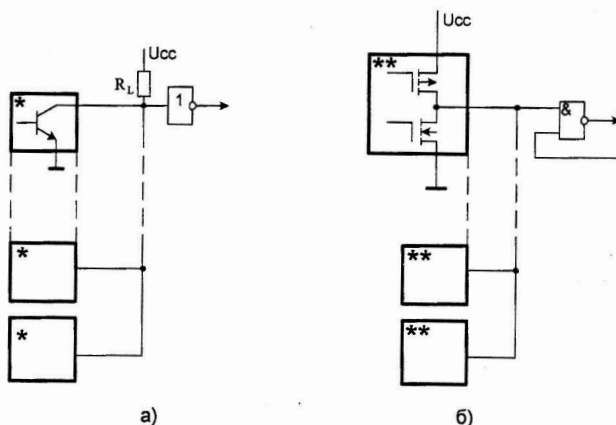


Рис. Д 10. Схеми об'єднання вихідних каскадів ВІС ЗП:
а - вихід з загальним колектором; б - вихід з трьома станами

Виходи ВІС ЗП для збільшення інформаційної ємності поєднують по провідному АБО чи по логічному АБО [1 - 5]. У кожен момент часу можливо вибрати тільки одну з об'єднаних мікросхем. При об'єднанні ВІС ЗП по логічному АБО розрахунок вихідних навантажувальних характеристик виробляється аналогічно розрахунку схем на логічних мікросхемах ТТЛ.

Навантаження на виході обраної ВІС ЗП з відкритим колектором (ОК) у крапці А (рис. Д11,б) визначається як

$$I_{ЛН} = I_{ОН} + (K_{СОІ} - 1) I_{ЛО} + z I_{ІН} + I_{R1};$$

$$C_L = (K_{СОС} - 1) C_0 + z C_1 + C_M,$$

де $I_{ЛН}$ - струм у режимі логічної одиниці через резистор R_1 ;

$I_{ОН}$ - струм через обрану ВІС ЗП;

$I_{ІН}$ - вхідний струм логічної одиниці одного ланцюга навантаження;

I_{R1} - струм через резистор R_1 ;

$K_{СОІ}$ - коефіцієнт об'єднання при токовому навантаженні;

C_0 - вихідна ємність один ВІС ЗП;

C_1 - вхідна ємність одного ланцюга навантаження;

z - кількість вхідних ланцюгів навантаження;

C_M - монтажна ємність;

$K_{СОС}$ - коефіцієнт об'єднання при ємнісному навантаженні;

$I_{ЛО}$ - вихідний струм витоку необраної ВІС ЗП.

Опір резистора R_L у схемі з ОК визначається мінімальним і максимальним значеннями припустимих рівнів узгодження на вході ВІС ЗП. Максимальне значення визначається з умови, коли усі виходи мікросхеми мають значення логічної одиниці (високий рівень):

$$R_{L \max} = \frac{U_{CC \min} - U_{OH \min}}{I_{OH} + z I_{IH} + (K_{CO} - 1) I_{LD}}, \quad (Д1)$$

а мінімальне значення $R_{L \min}$ - коли один з вихідів мікросхем має логічний нуль (низький рівень):

$$R_{L \min} = \frac{U_{CC \max} - U_{OL \max}}{I_{OL} - z I_{IL}}, \quad (Д2)$$

де $U_{CC \max}$ і $U_{CC \min}$ - напруга харчування мікросхем (максимальне і мінімальне значення);

$U_{OH \min}$ - мінімальний рівень логічної одиниці;

$U_{OL \max}$ - максимальний рівень логічного нуля;

I_{OH} , I_{OL} - вихідний струм логічних 1 і 0 мікросхеми ЗП;

I_{IH} , I_{IL} - вхідний струм логічних 1 і 0 схеми навантаження;

I_{LO} - струм витоку невідбраної схеми;

K_{CO} - кількість об'єднаних виходів мікросхеми ЗП;

z - кількість керованих входів схем навантажень.

Опір резистора R_L повинне бути в межах

$$R_{L \min} \leq R_L \leq R_{L \max}. \quad (Д3)$$

Якщо при розрахунку отримане $R_L < 1 \text{ кОм}$, то щоб виконати вимоги технічних умов на інтегральні схеми ТТЛ (R_L - навантаження ВІС ЗП, дивися рис. Д11), необхідно між вихідом ВІС і загальною шиною включити додатковий резистор R_1 (формулу для його визначення вивести самостійно).

Припустимий коефіцієнт об'єднання по вихіді при ємнісному навантаженні для схем з ОК визначається з формули:

$$K_{COC \text{ доп}} \leq \frac{C_{L \text{ доп}} + C_O - z C_1 - C_M}{C_O}, \quad (Д4)$$

де $C_{L \text{ доп}}$ - припустима ємність навантаження ВІС ЗП.

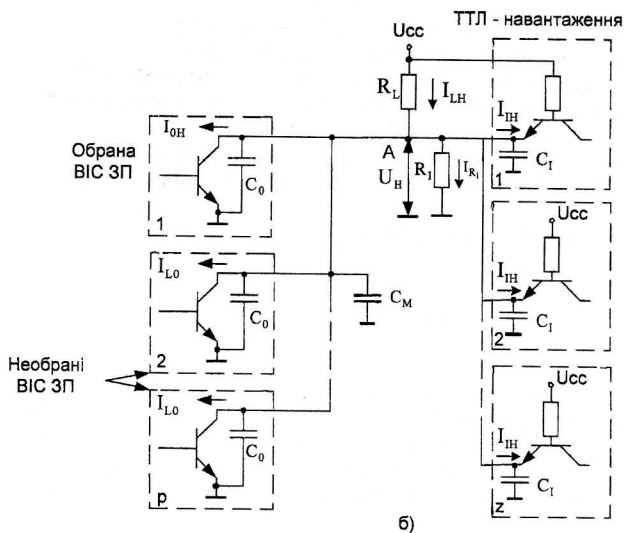
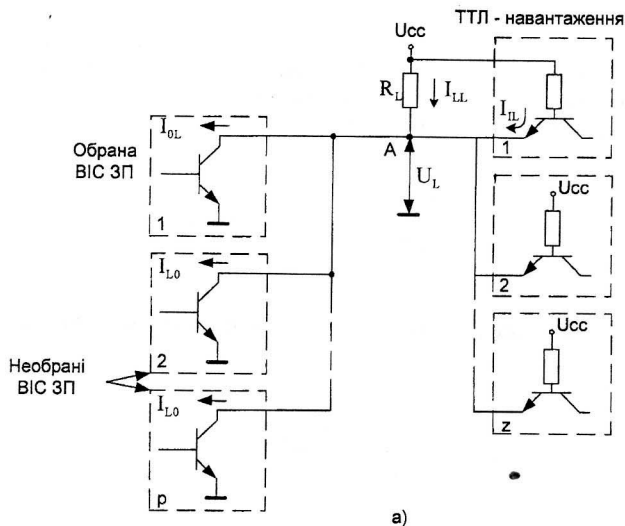


Рис. Д11. Вихідні ланцюги ВІС ЗП з загальним колектором:
а - режим логічного "0"; б - режим логічної "1"

Крім того, коефіцієнт об'єднання при токовому навантаженні для схем з ОК

$$K_{COI_{\text{доп}}} \leq \frac{I_{LH} - I_{OH} + I_{LO} - z I_{IH} - I_{R1}}{I_{LO}}. \quad (\text{Д5})$$

Д3. Режими роботи і часова діаграма одновібратора АГЗ

Мікросхема **K155АГЗ** складається з двох незалежних схем одновібраторів [7, 8]. Кожний з них містить тригер - формувач Шмітта і керуючі елементи. Кожен одновібратор має два інформаційних входи **A** і **B**, вхід установки в "0" **R**, прямий **Q** та інверсний $\bar{Q}$ виходи, а також виводи **C** і **RC** для підключення зовнішніх елементів (рис. Д13).

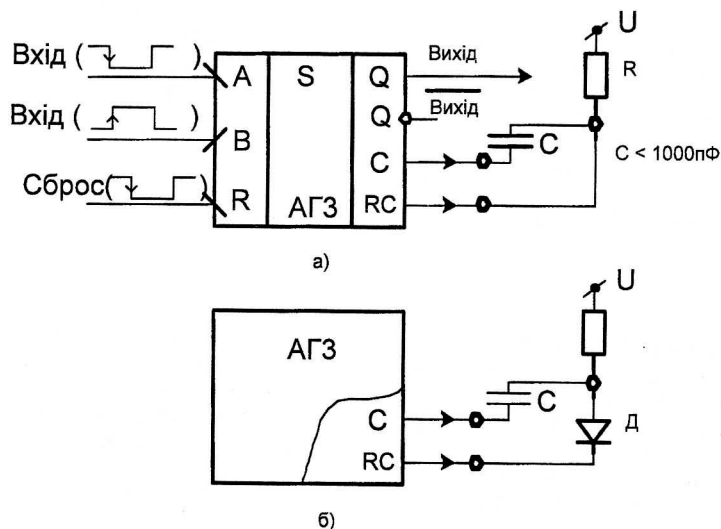


Рис. Д13. Варіанти підключення зовнішніх елементів

Запуск одновібратора здійснюється негативним перепадом на вході **A** або позитивним перепадом на вході **B**. При подачі на один з інформаційних входів імпульсу запуску, відбувається перезаряд ємності **C**. Час перезаряду ємності залежить від зовнішніх компонентів **R** і **C** і визначає тривалість імпульсу на виході. Даний цикл повторюється щораз при наявності на вході імпульсу запуску. Підключення зовнішніх елементів може бути виконане в залежності від конкретного випадку в двох варіантах, як це наведено на рис. Д13.

Схема на рис. Д13,а при використанні конденсатора з малими струмами витоку, відрізняється високою стабільністю. Для формування імпульсу широкого часового інтервалу (при $C > 1000$ пф) і при використанні електролітичного конденсатора доцільно використовувати схему рис. Д13,б.

Тривалість вихідного імпульсу визначається з формули:

$$\tau_{\text{имп}} = K \cdot R \cdot C \cdot \left(1 + \frac{0,7}{R}\right),$$

де $K = 0,28$ - для схеми (а) і $K = 0,25$ - для схеми (б).

Розмірність C - у пф, R - у кОм, $\tau_{\text{имп}}$ - у нс.

Режими роботи одновібратора АГЗ відбиті в табл. Д1.

Таблиця Д1

Режим	Вхід			Вихід	
	A	B	R	Q	$\bar{Q}$
Запуск	$1 \rightarrow 0$	1	1		
Запуск	0	$0 \rightarrow 1$	1		
Скидання	*	*	$1 \rightarrow 0$	0	1
Стійкий стан	*	*	0	0	1
Запуск	0	1	$0 \rightarrow 1$		

Особливістю схеми АГЗ є можливість її роботи в режимі повторного запуску. Це дозволяє утримувати одновібратор в активному стані наступним імпульсом запуску, не чекаючи повернення його у вихідний стан. Такий режим дозволяє збільшити тривалість вихідного імпульсу. Часова діаграма роботи АГЗ наведена на рис. Д14.

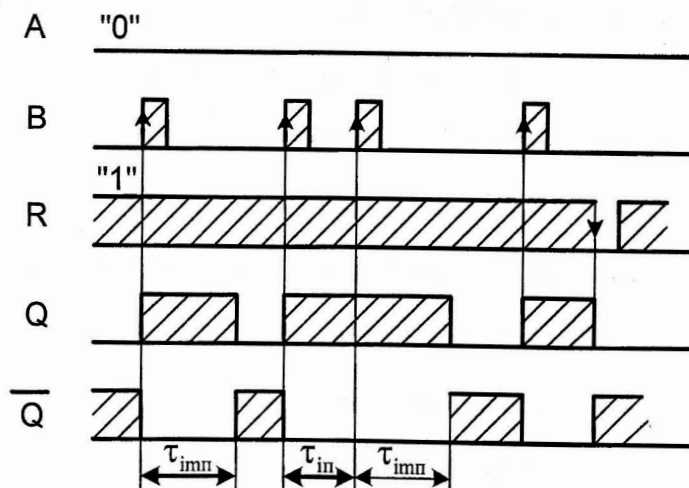


Рис. Д.14. Часова діаграма роботи одновібратора АГЗ

Фронт імпульсу повторного запуску стосовно фронту попереднього імпульсу запуску $t_{\text{п}}$ повинний знаходитися в наступному співвідношенні [8]:

$$0,22 C \leq t_{\text{п}} \leq \tau_{\text{имп}} + t^{01},$$

де розмірність C - у пф, $\tau_{\text{имп}}$, $t_{\text{п}}$, t^{01} - у нс. $t^{01} = 40$ нс.

Зміст

	Стр.
Лабораторна робота №1. РОЗРОБКА ТЕСТУ ФУНКЦІОНАЛЬНОГО КОНТРОЛЮ МОДУЛЯ ОПЕРАТИВНОГО ЗАПАМ'ЯТОВУЮЧОГО ПРИСТРОЮ.....	4
Лабораторна робота №2. ПРОЕКТУВАННЯ МОДУЛЯ ОЗП СТАТИЧНОГО ТИПУ....	12
Лабораторна робота №3. ПРОЕКТУВАННЯ МОДУЛЯ ОЗП ДИНАМІЧНОГО ТИПУ ЗІ СХЕМОЮ РЕГЕНЕРАЦІЇ ІНФОРМАЦІЇ.....	27
Лабораторна робота №4. РЕГЕНЕРАЦІЯ DRAM ЗА ДОПОМОГОЮ КОНТРОЛЕРА DMA	35
Лабораторна робота №5. МОДЕРНІЗАЦІЯ ОПЕРАТИВНОЇ ПАМ'ЯТІ КОМП'ЮТЕРА З ЗАСТОСУВАННЯМ МОДУЛІВ SIMM's	44
Лабораторна робота №6. ПАМ'ЯТЬ ТИПУ ЧЕРГИ І СТЕКА.....	49
Лабораторна робота №7. БУФЕРНІ ЗАПАМ'ЯТОВУЮЧІ ПРИСТРОЇ.....	53
Лабораторна робота №8. АНАЛІЗ АЛГОРИТМІВ І ПРОЕКТУВАННЯ МОДУЛІВ КЕШ-ПАМ'ЯТІ.	58
Додаток	
Д1. Опис тестів функціонального контролю ОЗП.....	64
Д2. Схеми вихідних ланцюгів ВІС елементів пам'яті	76
Д3. Режим роботи і часова діаграма одновібратора АГЗ..	80

МЕТОДИЧНІ ВКАЗІВКИ

**ДО ЛАБОРАТОРНОГО ПРАКТИКУМУ
з курсу "Проектування запам'ятовуючих пристроїв" для підготовки
бакалаврів та спеціалістів
на основі напрямку "Комп'ютерна інженерія"**

УКЛАДАЧ:

Юрій Володимирович Губарь

Віддруковано на різнографі
ТОВ фірма «Друк-Інфо»
Ум. друк. арк. 5,2. Обл.-вид. Арк. 4,8
Тираж 200 прим.. Замовлення № 1861
83000, м. Донецьк, вул. Артема, 58, к. 113
тел. 335-64-55