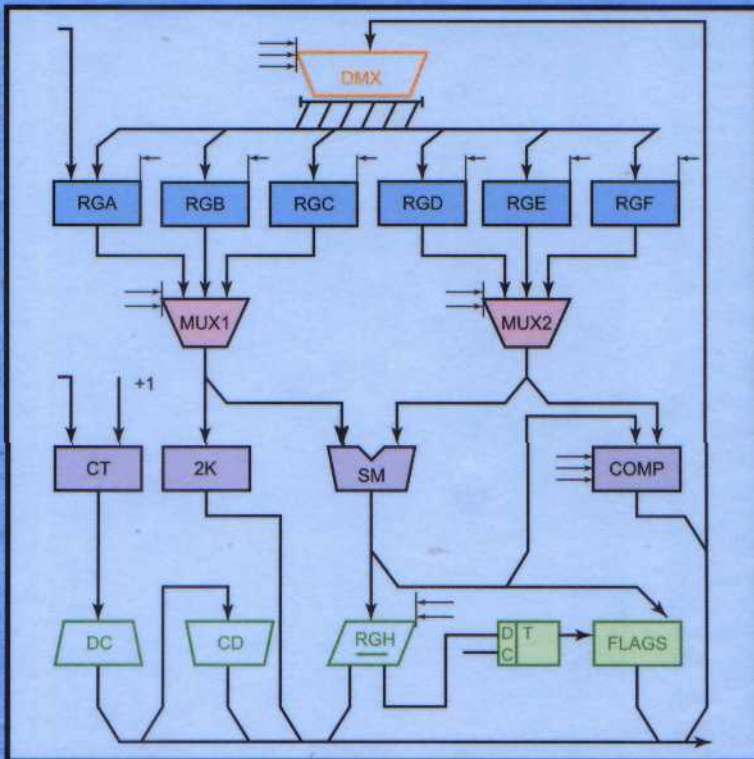


0 0 0 0 1 0 0 1 1 0 1 0 0 0 1 1 0
 0 0 0 1 1 0 0 0 1 0 0 1 0 1 0 0 0
 0 0 1 0 0 1 1 1 1 0 0 0 0 1 0 1 0
 0 0 1 1 0 1 1 0 0 1 1 1 0 1 1 0 0
 0 1 0 0 0 1 0 1 0 1 1 0 0 1 1 1 0

М.П.БАБИЧ, І.А.ЖУКОВ

КОМП'ЮТЕРНА СХЕМОТЕХНІКА



М. П. Бабич, І. А. Жуков

Комп'ютерна схемотехніка

*Рекомендовано Міністерством освіти і науки
України як навчальний посібник для студентів
вищих навчальних закладів*

**“МК-Прес”
Київ, 2004**

ББК з973.2–04я73–5

Б125

УДК 004.31 (075.8)

Рецензенти член-кореспондент НАН України, професор В.Ф. Євдокимов — директор Інституту проблем моделювання в енергетиці НАН України, доктор технічних наук, професор Г.М. Луцький — завдувач кафедри “Обчислювальна техніка” НТУУ “КПІ”

Бабич М. П., Жуков І. А.

Б125 Комп’ютерна схемотехніка: Навчальний посібник. — К.: “МК-Прес”, 2004. — 412 с., іл.

ISBN 966–96415-1-9

В цьому навчальному посібнику систематизовано викладені інформаційні, арифметичні і логічні основи мікроелектронних схем сучасних комп’ютерів. Розглянуто принципи побудови і функціонування логічних та запам’ятовуючих елементів, типових функціональних вузлів, аналого-цифрових і цифро-аналогових перетворювачів, електронної пам’яті, арифметико-логічних та керуючих пристроїв, мікропроцесорів, інтерфейсних контролерів.

Розраховано на студентів інженерно-технічних спеціальностей вищих навчальних закладів

ББК з973.2–04я73–5

Навчальний посібник

**Бабич Микола Павлович
Жуков Ігор Анатолійович**

Комп’ютерна схемотехніка

Рекомендовано Міністерством освіти і науки України як навчальний посібник для студентів вищих навчальних закладів, лист за №14/18.2–1242 від 13 червня 2002 року.

*Комп’ютерна верстка: Ю. О. Шлак
Дизайн обкладинки. М. В. Шашкова*

Підписано до друку 03.06 2004. Формат 70 × 100 1/16
Папір офсетний. Друк офсетний. Ум друк л 17. Обл.-вид л 25,8.
Тираж 1500 екз. Замовлення № 4-1175

ПП Савченко Л.О., Україна, м Київ, тел./ф.: (044) 517-73-77; e-mail: mc@symmetron.com.ua.
Свідоцтво про внесення суб’єкта видавничої справи до Державного реєстру видавників, виготовників та розповсюджувачів видавничої продукції. серія ДК №51582 від 28.11.2003 р.

Надруковано в ЗАТ “ВІПОЛ”. 03151, м Київ, вул Волинська, 60

ISBN 966–96415-1-9

© Бабич М.П., Жуков І.А., текст, ілюстрації, 2002
© “МК-Прес”, оформлення, дизайн обкладинки, 2004

Зміст

Передмова	5
Перелік вживаних скорочень	6
Розділ 1. Основи теорії комп'ютерної схемотехніки	8
1 1 Інформаційні основи комп'ютерної схемотехніки	8
1 2 Арифметичні основи комп'ютерної схемотехніки	16
1 3 Логічні основи комп'ютерної схемотехніки	37
1 4 Основні характеристики цифрових мікросхем	48
Розділ 2. Елементи комп'ютерної схемотехніки	60
2 1 Діодні транзисторні інтегрально-інжекційні та діодно-транзисторні логічні елементи	60
2 2 Транзисторно-транзисторні логічні елементи	68
2 3 Логічні елементи емітерно-зв'язної логіки	77
2 4 Логічні елементи на МОП-транзисторах	80
2 5 Імпульсна і потенціально-імпульсна системи елементів Магнітна схемотехніка	86
2 6 Асинхронні та синхронні RS-тригери	96
2 7 Тригери типів JK, T, D та DV	103
Розділ 3. Послідовнісні (накопичувальні) вузли комп'ютерної схемотехніки	111
3 1 Регистри	111
3 2 Лічильники	120
Розділ 4. Комбінаційні функціональні вузли комп'ютерної схемотехніки	132
4 1 Дешифратори	132
4 2 Шифратори	138
4 3 Мультиплексори і демультимплексори	141
4 4 Схеми порівняння і контролю	148
4 5 Перетворювачі кодів	156
4 6 Двійкові суматори	163
4 7 Двійково-десяткові суматори	176
Розділ 5. Цифро-аналогові і аналого-цифрові перетворювачі	179
5 1 Цифро-аналогові перетворювачі	179
5 2 Аналого-цифрові перетворювачі	185
Розділ 6. Основи комп'ютерної техніки	191
6 1 Загальна характеристика комп'ютерної техніки і коротка історія її розвитку	191
6 2 Архітектура і структура комп'ютерів	192
6 3 Принцип програмного керування	195

6 4	Основні характеристики комп'ютерів	197
6 5	Покоління комп'ютерів	198
Розділ 7. Пам'ять комп'ютерів		201
7 1	Загальна характеристика пам'яті	201
7 2	Основні структури напівпровідникової пам'яті	206
7 3	Кеш-пам'ять	211
7 4	Постійна пам'ять	214
7 5	Флеш-пам'ять	222
7 6	Статичні запам'ятовуючі пристрої	227
7 7	Динамічна пам'ять	230
Розділ 8. Арифметико-логічні пристрої та пристрої керування		240
8 1	Класифікація арифметико-логічних пристроїв	240
8 2	Мови опису операційних пристроїв	242
8 3	Пристрої керування	244
8 4	Керуючі автомати із схемною логікою	246
8 5	Структурний синтез керуючого автомата зі схемною логікою	249
8 6	Синтез мікропрограмного автомата з програмовною логікою	252
8 7	Центральний пристрій керування	258
Розділ 9. Проектування спеціалізованих арифметико-логічних пристроїв		261
9 1	Реалізація операції додавання	261
9 2	Реалізація операції віднімання	268
9 3	Реалізація операцій додавання і віднімання	274
9 4	Реалізація операції множення	281
9 5	Реалізація операції ділення	290
Розділ 10. Мікропроцесори		303
10 1	Загальна характеристика процесорів і мікропроцесорів	303
10 2	Однокристальні восьмирозрядні мікропроцесори	314
10 3	Однокристальні шістнадцятирозрядні мікропроцесори	333
10 4	Арифметичні співпроцесори	340
10 5	Суперскалярні 32-розрядні мікропроцесори із <i>CISC</i> -архітектурою	342
10 6	Суперскалярні мікропроцесори з <i>RISC</i> -архітектурою	354
Розділ 11. Інтерфейси мікропроцесорних систем		359
11 1	Загальна характеристика мікропроцесорних систем	359
11 2	Загальна характеристика інтерфейсів	364
11 3	Інтерфейсні мікросхеми	369
11 4	Програмовні інтерфейсні контролери	380
Висновки		397
	Словник загальних термінів	398
	Список літератури	410

Передмова

Комп'ютерна схемотехніка — це науково-технічна дисципліна, яка вивчає теоретичні методи аналізу і синтезу схем комп'ютерів (електронних обчислювальних машин) і засоби їхньої технічної реалізації. Розвиток комп'ютерної схемотехніки є основою удосконалення архітектури комп'ютерів, якісного підвищення їхньої продуктивності та надійності, істотного зменшення масових та габаритних показників. Комп'ютери широко використовують у різних галузях господарства.

Метою застосування комп'ютерів є інформатизація суспільства — забезпечення повсюдного використання автоматизованих методів збирання, передавання, оброблення і зберігання інформації. Вирішення проблеми інформатизації забезпечує вихід країни на новий рівень цивілізації

Даний навчальний посібник призначений для студентів, які навчаються за спеціальностями "Комп'ютерні системи та мережі", "Інформаційні управляючі системи та технології", "Програмне забезпечення автоматизованих систем". Тут знайшли відображення результати методологічних досліджень з оптимізації навчального процесу, одержані на кафедрі обчислювальної техніки Інституту інформатики Національного авіаційного університету.

У посібнику викладені інформаційні, арифметичні, логічні та схемотехнічні основи комп'ютерної схемотехніки. Розглянуті елементи і типові функціональні вузли, а також основні пристрої комп'ютерів: електронна пам'ять, арифметико-логічні, керування та системи введення/виведення інформації. У стислій формі наведені характеристики процесорів, мікропроцесорів та комп'ютерів, дані визначення, класифікація, основні області їхнього застосування.

Автори вдячні рецензентам — директору Інституту проблем моделювання в енергетиці НАН України, члену-кореспонденту НАН України, професору В. Ф. Євдокимову та завідувачу кафедри "Обчислювальна техніка" НТУУ "КПІ", доктору технічних наук, професору Г. М. Луцькому за слушні зауваження.

Автори висловлюють подяку кандидату технічних наук Г. А. Бабич за допомогу в підготовці книги до видання.

Перелік вживаних скорочень

A	акумулятор
AK	арифметичний канал
АЛБ	арифметико-логічний блок
АЛП	арифметико-логічний пристрій
АЦП	аналогово-цифровий перетворювач
АШ	арбітр шини
БАД	буфер адреси даних
БЕТ	багатомітерний транзистор
БМК	блок місцевого керування
БІФ	блок інтерфейса
ВІС	велика інтегральна схема
ВК	відкритий колектор
ГТІ	генератор тактових імпульсів
ДДНФ	досконала диз'юнктивна нормальна форма
ДКНФ	досконала кон'юнктивна нормальна форма
ЕЗЛ	емітерно-зв'язана логіка
ЕОМ	електронна обчислювальна машина
ЕП	елемент пам'яті; емітерний повторювач
ЗГ	задаючий генератор
ЗЗП	зовнішній ЗП
ЗП	запам'ятовуючий пристрій
І²Л	інтегральна інжекційна логіка
ІМС	інтегральна мікросхема
КА	керуючий автомат
КВВ	канал введення-виведення
КМОН	комплементарна МОН-структура
КОП	код операції
КПДП	контролер ПДП
КПР	контролер переривань
КПТО	комплекс програм технічного обслуговування
КШ	контролер шини
МЕН	структура "метал-напівпровідник"
МІС	мала інтегральна схема
МК	мікрокоманда
МНОН	структура "метал-нітрид-оксид-напівпровідник"
МОН	структура "метал-оксид-напівпровідник"
МП	мікропроцесор
МПА	мікропрограмний автомат
МПК	мікропроцесорний комплект
МПС	мікропроцесорна система
МЦ	машинний цикл
НВІС	надвелика інтегральна схема
НГМД	накопичувач на гнучких магнітних дисках
НЖМД	накопичувач на жорстких магнітних дисках

НМС	накопичувач на магнітних стрічках
НОД	накопичувач на оптичних дисках
ОА	операційний автомат
ОЗП	оперативний ЗП
ОП	оперативна пам'ять; операційний підсилювач
ОС	операційна система
ПВВ	пристрій введення-виведення інформації
ПВв	пристрій введення інформації
ПВив	пристрій виведення інформації
ПД	перемикач Джозефсона
ПДП	прямий доступ до пам'яті
ПЕОМ	персональна ЕОМ
ПЗП	постійний ЗП
ПК	пристрій керування
ПМК	пам'ять мікрокоманди
ПП	периферійний (зовнішній) пристрій
ППА	програмовний паралельний адаптер
ППЗП	програмовний ПЗП
ППі	програмовний послідовний інтерфейс
ППП	пакети прикладних програм
ПТ	програмовний таймер
РЗП	регістр загального призначення
РПЗП-ЕС	репрограмовний ПЗП з електричним стиранням
РПЗП-УФ	репрограмовний ПЗП із стиранням даних ультрафіолетовим променем
РТ	розподільник тактів
СІС	середня інтегральна схема
СПП	схема прискореного перенесення
СШ	системна шина
ТЛ	транзисторна логіка
ТТЛ	транзисторно-транзисторна логіка
ТТЛШ	транзисторно-транзисторна логіка з діодами Шотки
ФАМК	формувач адреси МК
ЦАП	цифро-аналоговий перетворювач
ЦП	центральний процесор
ЦПК	центральний пристрій керування
ША	шина адреси
ШВВ	шина введення-виведення
ШД	шина даних
ШК	шина керування

Основи теорії комп'ютерної схемотехніки

1.1. Інформаційні основи комп'ютерної схемотехніки

1.1.1. Інформатика, інформація, сигнали та їхнє представлення

Широке застосування комп'ютерів сприяє науково-технічному розвитку країни. Сфера використання комп'ютерів охоплює практично всі види людської діяльності.

Комп'ютер — це програмно керована фізична система, призначена для алгоритмічного опрацювання інформації, поданої сигналами.

У широкому значенні слова інформація є відбитком реального світу. Інформація — це єдиний ресурс життєзабезпечення, що не зменшується. Більше того: її обсяг у даний час подвоюється щорічно. Інформація, підготовлена для опрацювання на комп'ютерах, називається **даними**.

Інформаційний процес включає в себе такі етапи:

1. збирання інформації від різноманітних джерел і представлення її у формі, необхідній для введення у комп'ютер;
2. передачу (пересилку) інформації від джерела до приймача;
3. зберігання — процес передачі інформації у часі;
4. оброблення — систематичне виконання операцій над даними;
5. видачу результату опрацювання користувачеві.

На всіх цих етапах використовують засоби комп'ютерної схемотехніки. До інформації пред'являють такі вимоги:

- коректність (однозначність сприйняття);
- цінність (корисність) і оперативність (актуальність),
- точність, достовірність і стійкість (спроможність реагувати на зміни початкових даних);
- достатність (повнота) — наявність мінімально необхідного обсягу інформації для прийняття правильного рішення.

Структуру і загальні властивості інформаційних процесів вивчають в інформатиці, яка містить:

- теорію інформації;
- алгоритмічні, програмні й комп'ютерні засоби оброблення інформації;
- архітектуру комп'ютерів, системи штучного інтелекту, обчислювальні мережі та ін.

У теорії інформації вивчають процеси передавання, перетворення і зберігання інформації, куди входять:

- методи визначення кількості інформації в повідомленні;

- раціональні засоби представлення інформації за допомогою різноманітних символів (букв, цифр та ін.);
- засоби формування, виявлення й оцінки параметрів інформаційних процесів.

Упорядковану послідовність символів (букв, цифр, математичних знаків, призначених для передачі інформації), закодовану в матеріальній формі, називають **повідомленням**.

Інформаційне повідомлення завжди зв'язане з джерелом і приймачем інформації, сполученими каналом передачі (рис. 1.1).

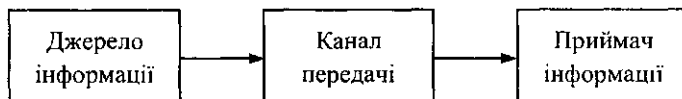


Рис. 1.1. Інформаційна модель каналу передачі

Джерелом і приймачем інформації можуть бути як люди, так і технічні пристрої (комп'ютери, датчики, індикатори та ін.). **Каналом передачі** (зв'язку) називається сукупність пристроїв, що мають один вхід і один вихід, та призначені для передачі інформації на відстані. Повідомлення можуть мати різноманітні форми: звуку, тексту, зображення, електричної напруги від датчиків (наприклад, від термопар).

Інформаційні повідомлення розміщують на машинних носіях інформації. **Носій інформації** — це будь-яке запам'ятовуюче предметне середовище, призначене для записування і зберігання інформації з метою її безпосереднього введення в комп'ютер. Носій інформації є проміжною ланкою між комп'ютером і первинним документом, який містить числові дані, текстові матеріали, схеми, графіки, різноманітні виміри.

Зберігання повідомлень — це, як правило, три види операцій: записування, власне зберігання і зчитування. Інформація записується до носія за допомогою зміни фізичних або механічних властивостей запам'ятовуючого середовища. Дані зчитуються за допомогою перетворення їх в електричні сигнали. Вважають, що в носії інформації сигнал зберігається в закодованій формі.

Носії інформації розрізняють за такими ознаками:

- середовищем накопичення: неперервні (магнітні стрічки і диски) та дискретні, де кожній збереженій одиниці даних виділяється своя дискретна ділянка або елемент (феритові сердечники, перфокарти, перфострічки, тригери, кріотрони та ін.);
- типом матеріалу: папір з текстом або малюнком; паперові перфострічки, перфокарти; магнітні плівки, магнітні стрічки і диски; напівпровідникові схеми пам'яті;
- засобом зчитування даних: механічні, оптичні, магнітні, електричні;
- конструктивним виконанням: стрічкові, дискові, електронні та ін.

Для передачі інформації від джерела до приймача повідомлення перетворюють у сигнали. Відповідно до ДСТУ 2938–94 "сигнал – це зміна фізичної величини, що використовується для пересилання даних". Таким чином, сигнал утворюється на основі деякої фізичної величини (електромагнітні або акустичні коливання, електрична напруга та ін.), яку традиційно називають енергетичним носієм, і зміни одного або декі-

льких її параметрів (амплітуди, частоти, фази, тривалості та ін.) за законом переданої інформації. Вважають, що сигнал – це матеріально-енергетичне втілення повідомлення. За допомогою сукупності сигналів можна подати будь-яке складне повідомлення. Сигнал може перетворюватися без зміни смислу інформації з однієї фізичної величини в іншу, зручнішу для передачі по каналу зв'язку й оброблення в схемах комп'ютера. Зміна параметрів фізичної величини за законом переданого повідомлення називається **модуляцією**, а змінювані параметри — **інформативними**.

Сигнали класифікують за такими ознаками:

- ступенем визначеності очікуваних значень — випадкові й детерміновані;
- структурою часової зміни — неперервні й дискретні;
- роллю переданої інформації в комп'ютері — адреси, дані й керування;
- особливостями спектрального представлення — низькочастотні й високочастотні, вузькосмужні й широкосмужні;
- способом перетворення — кодовані, декодовані, посилені, дискретизовані та ін.;
- приналежністю до виду зв'язку — телеграфні, телефонні, радіолокаційні, міжмашинні та внутрішньомашинні;
- характером зміни кодованих сигналів у синхронізовані моменти часу — потенціальні та імпульсні.

У процесі передачі сигналів від джерела до приймача фізичні величини і засоби їхньої модуляції можуть багаторазово змінюватися, але зміст повідомлення залишається незмінним, оскільки воно визначається тільки за законом модуляції.

Для інформаційного обміну використовують знаки різного рангу:

- перший ранг — символи. **Символ** — це елементарна одиниця повідомлення. Якщо число різноманітних символів обмежене, то їхню сукупність називають алфавітом (наприклад, букви латинського алфавіту, двійкові символи 0 і 1 — в технічних пристроях);
- другий ранг — **слова** (групи символів), з яких будуються фрази і вирази.

У загальному випадку спосіб формалізованого описування різноманітних сигналів (і відповідно повідомлень) називається **представленням інформації**. У теорії інформації розглядають не фізичне, а математичне представлення сигналів, тобто їхній опис за допомогою різноманітних функцій, формул, графіків, законів розподілу ймовірностей.

Найбільш поширеними способами представлення сигналів є часовий, спектральний, статистичний. У багатьох випадках інформація щодо протікання деякого фізичного процесу надходить від відповідних датчиків у вигляді електричних сигналів, які безупинно змінюються у часі. Розрізняють такі різновиди сигналів, що описуються часовою функцією $y(t)$.

1. Неперервна функція неперервного аргументу в інтервалі часу $0 \leq t \leq t_k$ (рис. 1.2, а).
2. Дискретна функція неперервного аргументу (рис. 1.2, б). Значення, набуті функцією $y(t)$, утворюють дискретний ряд чисел y_i , $i = 1, 2, \dots, k$. Значення аргументу може бути будь-яким у заданому інтервалі часу $0 \leq t \leq t_k$. Перетворення неперервної функції $y(t)$ в дискретну множину значень y_i називається **квантуванням за рівнем**.

3. Неперервна функція дискретного аргументу (рис. 1.2, в). Значення функції $y(t)$ визначаються тільки на дискретній множині $t_i, i = 1, 2, \dots, k$. Функція $y(t)$ може набувати будь-яких значень у заданому діапазоні. Перетворення функції $y(t)$ неперервного аргументу t в функцію $y(t_i)$ дискретного аргументу t_i називається **дискретизацією (квантуванням) у часі**.
4. Дискретна функція дискретного аргументу (рис.1.2, г). Значення, що приймаються функцією й аргументом, утворюють дискретні ряди чисел y_0, y_1, \dots, y_k і t_0, t_1, \dots, t_k .

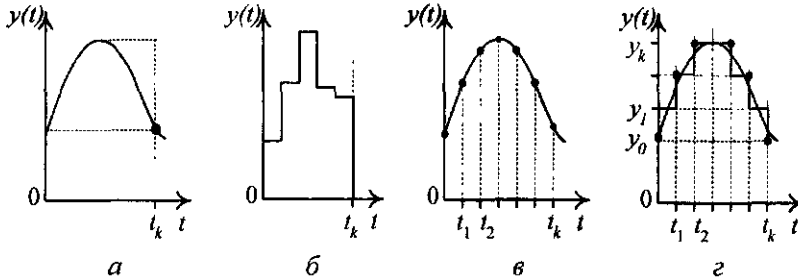


Рис. 1.2. Математичне представлення сигналів

Перший з розглянутих різновидів описує неперервні (аналогові) сигнали, другий і третій — дискретно-неперервні, а четвертий — чисто дискретні. Сумісне застосування дискретизації й квантування дозволяє перетворювати неперервну функцію в чисто дискретну.

Відповідно до теореми Котельникова сигнал, що описується функцією з обмеженим спектром, визначається своїми дискретними значеннями, які відраховуються через інтервали часу $t = 1/2F_c$, де F_c — ширина спектру. Таким чином, сигнал $y(t)$ можна передавати окремими миттєвими значеннями, які відраховуються через кінцевий інтервал часу. За цими значеннями комп'ютер повністю відновлює первинний неперервний сигнал.

До дискретно-неперервних функцій відносять також часо-імпульсне представлення первинного сигналу $y(t)$ прямокутними імпульсами з неперервним інформативним параметром t/T , де t — тривалість імпульсів, пропорційна значенню сигналу; T — період повторення імпульсів (рис. 1.3, а).

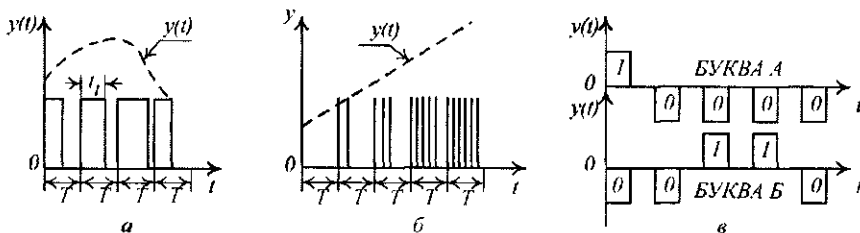


Рис. 1.3. Представлення сигналів:

а — часо-імпульсне; б — число-імпульсне; в — імпульсно-кодове

При число-імпульсному представленні (рис.1.3, б) інформативним параметром є число імпульсів за період. У телеграфному зв'язку використовують імпульсно-кодове представлення символів. Наприклад, у коді Бодо передачу символів реалі-

зують послідовною послілкою в часі п'яти імпульсів різної полярності, як показано на рис. 1.3, в для букв А і Б.

У комп'ютерній схемотехніці використовують розрядно-цифрове (або просто цифрове) кодування, у якому первинний сигнал представляється групою символів, які відображають значення цифр 0 і 1 двійкової системи числення електричними сигналами, наприклад, імпульсами. Наявність імпульсу відповідає цифрі 1, відсутність — 0. Розряди двійкового числа характеризуються вагою, кратною степеню двійки — 1, 2, 4, 8, ... (у напрямку від молодших до старших розрядів), наприклад, для чотирирозрядного двійкового числа маємо:

$$\begin{aligned}x &= x_3 \cdot x_2 \cdot x_1 \cdot x_0 = x_3 \cdot 2^3 + x_2 \cdot 2^2 + x_1 \cdot 2^1 + x_0 \cdot 2^0 = \\ &= x_3 \cdot 8 + x_2 \cdot 4 + x_1 \cdot 2 + x_0 \cdot 1\end{aligned}$$

Якщо двійкове число $x_2 = 1101$, то отримуємо такий десятковий еквівалент:

$$x_{10} = 1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 13$$

Цифровий код передають послідовно в часі (**послідовний код**) за допомогою одного каналу передачі (рис. 1.4, а) або одночасно (**паралельний код**) за допомогою багатоканальної передачі (рис. 1.4, б).

На практиці послідовний код використовують при передачі інформації на великій відстані (наприклад, між комп'ютерами), а паралельний код — при передачі інформації на малій відстані (наприклад, внутрішньомашинні).

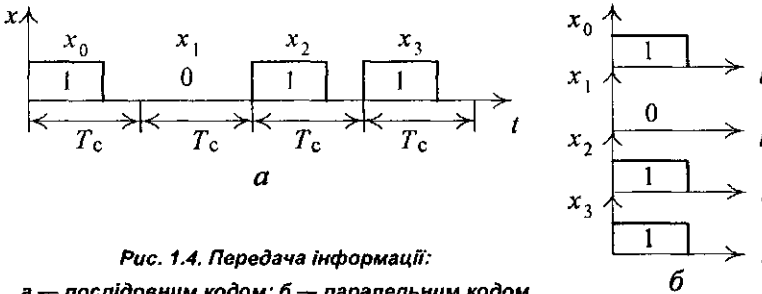


Рис. 1.4. Передача інформації:

а — послідовним кодом; б — паралельним кодом

1.1.2. Інформаційні міри

У теорії передачі та перетворення інформації встановлені інформаційні міри кількості та якості інформації — семантичні, структурні, статистичні.

Семантичний підхід дозволяє виділити корисність чи цінність інформаційного повідомлення. У структурному аспекті розглядають будову масивів інформації та їхній вимір простим підрахунком інформаційних елементів або комбінаторним методом. Структурний підхід використовують для оцінки можливостей інформаційних систем незалежно від умов їхнього застосування. При використанні структурних мір інформації враховують тільки дискретну будову повідомлення, кількість інформаційних елементів, що містяться в ньому, зв'язків між ними. При структурному підході розрізняють геометричну, комбінаторну й адитивну міри інформації.

Геометрична міра визначає параметри геометричної моделі інформаційного повідомлення (довжина, площа, об'єм) у дискретних одиницях. Цю міру застосовують

як для оцінки інформаційної ємності всієї моделі, так і для оцінки кількості інформації в одному повідомленні.

У комбінаторній мірі кількість інформації I визначають числом комбінацій елементів (символів), що збігається з числом:

$$I = \frac{q!}{n!(q-n)!},$$

- сполучень з q елементів по n :
наприклад, для множини цифр 1, 2, 3, 4 можна скласти шість сполучень по дві цифри: 12, 13, 14, 23, 24, 34;
- перестановок $I = q!$, наприклад, для множини букв a, b, c можна одержати шість перестановок: $abc, acb, bac, bca, cab, cba$;
- розміщень з повтореннями з q елементів по n :

$$I = q^n.$$

Наприклад, для $q = 0, 1$ та $n = 3$ маємо: 000, 001, 010, 011, 100, 101, 110, 111.

Велике поширення одержала адитивна міра. Нехай N — число рівномірних повідомлень, n — їхня довжина, q — число букв алфавіту, який використовується для передачі інформації. Кількість можливих повідомлень довжини n дорівнює числу розміщень з повтореннями

$$N = q^n. \quad (1.1)$$

Цю міру наділяють властивістю адитивності, щоб вона була пропорційною довжині повідомлення і дозволяла підсумовувати кількість інформації ряду джерел. Для цього Хартлі запропонував логарифмічну функцію як міру кількості інформації:

$$I = \log N = n \log q. \quad (1.2)$$

Кількість інформації, що припадає на один елемент повідомлення, називається **ентропією**:

$$H = \frac{I}{n} = \log q. \quad (1.3)$$

Основа логарифму залежить від вибору одиниці кількості інформації. Якщо для алфавіту використовують двійкові цифри 0 і 1, то за основу логарифму приймають $q = 2$, у результаті чого $I = n \log_2 2 = n$. При довжині $n = 1$ одержують $I = 1$ і цю кількість інформації називають **бітом**.

Передача повідомлення довжиною $n = 1$ еквівалентна вибору одного з двох можливих рівномірних повідомлень — одне з них дорівнює одиниці, інше — нулю. Двійкове повідомлення довжиною n вміщує n бітів інформації. Якщо основа логарифма дорівнює 10, то кількість інформації вимірюється в десяткових одиницях — **дїтах**, причому 1 дїт = 3,32 біта. Наприклад, текст складений з 32 букв алфавіту і передається послідовно по телетайпу в двійковому коді. При цьому кількість інформації $I = \log_2 N = \log_2 32 = 5$ бітів. Далі використовуються логарифми з основою два.

У загальному випадку повідомлення з'являються з різною ймовірністю. Статистична міра використовує ймовірнісний підхід до оцінки кількості інформації. Згідно із Шенноном кожне повідомлення характеризується ймовірністю появи і чим вона менша, тим більше в повідомленні інформації. Ймовірність конкретних типів повідомлень установлюють на основі статистичного аналізу.

Нехай повідомлення створюється послідовною передачею букв деякого алфавіту $x_1, \dots, x_n, \dots, x_q$ з ймовірністю появи кожної букви $p(x_1) = p_1, \dots, p(x_i) = p_i, \dots, p(x_q) = p_q$, при цьому виконується умова: $p_1 + \dots + p_i + \dots + p_q = 1$.

Множину з відомим розподілом елементів називають **ансамблем**. За Шенноном кількість інформації, яка міститься в повідомленні x_i , розраховують за формулою

$$I(x_i) = \log \frac{1}{p_i}. \quad (1.4)$$

Для абсолютно достовірних повідомлень $p_i = 1$, кількість інформації $I(x_i) = 0$; при зменшенні значення p_i кількість інформації збільшується.

Нехай в ансамблі всі букви алфавіту $x_1, \dots, x_n, \dots, x_q$ — рівномірні, тобто $p_1 = p_2 = \dots = p_q = 1/q$, і статистично незалежні. Тоді кількість інформації в повідомленні довжиною n букв з урахуванням виразу (1.4)

$$I = \sum_{i=1}^n I(x_i) = \sum_{i=1}^n \log \frac{1}{p_i} = \log \frac{1}{p_1} + \dots + \log \frac{1}{p_n} = n \log q,$$

що збігається з мірою Хартлі згідно з виразами (1.1) і (1.2).

За Шенноном інформація — це зняття невизначеності, яке розуміють таким чином. До дослідження подію (наприклад, появу букви x_i) характеризують малою початковою ймовірністю p_n , якій відповідає велика невизначеність. Після дослідження невизначеність зменшується, оскільки кінцева ймовірність $p_k > p_n$. Зменшення невизначеності розраховують як різницю між початковим I_n і кінцевим I_k значеннями кількості інформації. Наприклад, для $p_n = 0,1$ і $p_k = 1$ одержимо:

$$\Delta I = I_n - I_k = \log \frac{1}{p_n} - \log \frac{1}{p_k} = \log 10 - \log 1 = 3,32 \text{ біт}.$$

Нехай складне повідомлення характеризується алфавітом з букв x_1, x_2, \dots, x_q , їхніми ймовірностями p_1, p_2, \dots, p_q і частотою появи кожної букви m_1, m_2, \dots, m_q . Усі повідомлення статистично незалежні, при цьому $m_1 + m_2 + \dots + m_q = m$. Загальна кількість інформації для всіх q типів повідомлень з урахуванням виразу (1.4)

$$I_{\Sigma} = \sum_{i=1}^q m_i \log \frac{1}{p_i}.$$

Середнє значення кількості інформації на одне повідомлення (ентропія) за формулою Шеннона

$$H = \frac{I_{\Sigma}}{m} = \sum_{i=1}^q \frac{m_i}{m} \log \frac{1}{p_i} = \sum_{i=1}^q p_i \log \frac{1}{p_i}, \quad (1.5)$$

де при великому значенні m відношення m_i/m характеризує ймовірність p_i кожної букви. Вираз $\log 1/p_i$ розглядають як часткову ентропію, яка характеризує інформативність букви x_i , а ентропію H — як середнє значення часткових ентропій. При малих значеннях p_i часткова ентропія велика, а з наближенням p_i до одиниці вона наближається до нуля (рис. 1.5, а).

Функція $\eta = (p_i) = p_i \log 1/p_i$, відображає внесок букви x_i в ентропію H . Як бачимо, при $p_i = 1$ ця функція дорівнює нулю, потім зростає до свого максимуму і при змен-

шенні p_i наближається до нуля. Функція $\eta(p_i)$ при значенні $p_i = 0,37$ має максимум 0,531.

Інтерес викликають повідомлення з використанням двобуквеного алфавіту x_1 і x_2 (наприклад, цифри 0 і 1). Оскільки при $q = 2$ ймовірність літер алфавіту $p_1 + p_2 = 1$, то можна покласти, що $p_1 = p$ і $p_2 = 1 - p$. Тоді ентропію визначають співвідношенням

$$H = \eta(p_1) + \eta(p_2) = p \log \frac{1}{p} + (1 - p) \log \frac{1}{(1 - p)},$$

графік якої показаний на рис. 1.5, б. Він утворюється підсумовуванням двох графіків, які визначають ентропію кожної з двох букв. Із графіків видно, що при $p = 0$ або $p = 1$ ентропія дорівнює нулю і невизначеність цілком знімається. Це значить, що з ймовірністю, що дорівнює одиниці, можна знати, яким буде наступне повідомлення.

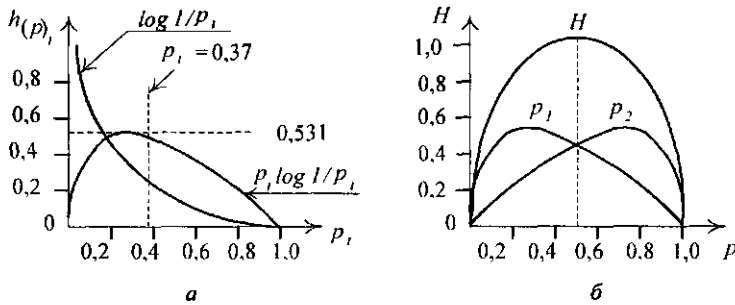


Рис. 1.5. Графіки функції H :

а — для часткової ентропії; б — для двійкових повідомлень

Ентропія двобуквених повідомлень досягає максимального значення, що дорівнює 1 біт при $p = 0,5$, і її графік симетричний щодо цього значення. Це той випадок, коли найважче вгадати, яке повідомлення буде наступним, — тобто ситуація найбільш невизначена.

У загальному випадку ентропія має такі властивості.

1. Ентропія є величина дійсна, неперервна, обмежена і невід'ємна.
2. Ентропія дорівнює нулю, якщо повідомлення заздалегідь відоме. У цьому випадку деяке повідомлення задане з ймовірністю $p_i = 1$, а ймовірність інших дорівнює нулю.
3. Ентропія максимальна, якщо всі повідомлення рівноймовірні: $p_1 = p_2 = \dots = p_q = 1/q$. Тоді на основі виразу (1.5) отримаємо:

$$H_{\max} = \sum_{i=1}^q p_i \log \frac{1}{p_i} = \sum_{i=1}^q \frac{1}{q} \log q = \log q,$$

що збігається з виразом (1.3). У цьому випадку оцінки кількості інформації за Хартлі і Шенноном збігаються.

4. При нерівних ймовірностях кількість інформації за Шенноном менша міри Хартлі. При об'єднанні ентропій двох незалежних джерел повідомлень їхні ентропії додаються.

В комп'ютері найменшою можливою одиницею об'ємної (геометричної) міри інформації є біт. Об'єм (або ємність) інформації підраховується за кількістю двійкових

символів 0 і 1, записаних у пам'яті комп'ютера. При цьому можливе тільки ціле число бітів на відміну від ймовірнісного підходу, де може бути і не ціле число.

Для зручності використання введені й більші за біт одиниці кількості інформації. Так, двійкове слово з восьми символів містить 1 байт інформації, 1024 байт створюють кілобайт (Кбайт), 1024 Кбайт — мегабайт (Мбайт) і 1024 Мбайт — гігабайт (Гбайт); при цьому $1024 = 2^{10}$

Між об'ємною і ймовірнісною кількостями інформації співвідношення неоднозначне. Якщо повідомлення допускає вимірювання кількості інформації і об'ємно і ймовірно, то вони не обов'язково збігаються. При цьому ймовірнісна кількість не може бути більша за об'ємну. В подальшому тексті кількість інформації розуміється в об'ємному значенні.

1.2. Арифметичні основи комп'ютерної схемотехніки

1.2.1. Принципи побудови системи числення

Числова інформація в комп'ютерах характеризується:

- системою числення (двійкова, десяткова та ін.);
- видом числа (числа дійсні, комплексні, масиви);
- типом числа (змішане, ціле, дробове);
- формою представлення числа (місце коми) — з природною (змінною), фіксованою, плаваючою комами;
- розрядною сіткою і форматом числа;
- діапазоном і точністю подання чисел;
- способом кодування від'ємних чисел — прямим, оберненим та доповняльним кодами;
- алгоритмами виконання арифметичних операцій.

Системою числення називається сукупність цифр і правил для записування чисел. Запис числа у деякій системі числення називається його **кодом**. Усі системи числення поділяють на позиційні й непозиційні. Для запису чисел у позиційній системі числення використовують певну кількість графічних знаків (цифр і букв), які відрізняються один від одного. Число таких знаків q називається **основою позиційної системи числення**. В комп'ютерах використовують позиційні системи з різною основою.

Система числення з основою два (цифри 0 і 1) називається двійковою, система числення з основою три (цифри 0, 1, 2) — трійковою і т.д. У системах числення з основою меншою десяти використовують десяткові цифри, а для основи більшої десяти додають букви латинського алфавіту — *A, B, C, D, E, F*. Далі в позначеннях при необхідності пишуть десятковий індекс, що дорівнює основі системи числення, яка застосована (табл. 1.1).

У позиційних системах числення значення кожної цифри визначається її зображенням і позицією в числі. Окремі позиції в записі числа називають **розрядами**, а номер позиції — номером розряду. Число розрядів у записі числа називається його **розрядністю** і збігається з довжиною числа.

Таблиця 1.1

Основа q	Система числення	Знаки
2	Двійкова	0, 1
3	Трійкова	0, 1, 2
5	П'ятіркова	0, 1, 2, 3, 4
8	Вісімкова	0, 1, 2, 3, 4, 5, 6, 7
10	Десяткова	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
16	Шістнадцяткова	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

У непозиційних системах числення значення кожної цифри не залежить від її позиції. Найвідомішою непозиційною системою є римська, в якій використовуються сім знаків — I, V, X, L, C, D, M, що відповідають таким значенням:

I	V	X	L	C	D	M
1	5	10	50	100	500	1000

Наприклад III — 3; LIX — 59, DLV — 555

Недоліком непозиційної системи є відсутність нуля та формальних правил запису чисел і відповідно арифметичних дій з ними (хоч за традицією римськими числами часто користуються при нумерації розділів у книгах, віків у історії та ін.). Система числення повинна забезпечувати

- можливість представлення будь-якого числа в заданому діапазоні,
- однозначність, стислість запису числа і простоту виконання арифметичних операцій,
- досягнення високої швидкодії машини в процесі оброблення інформації

Число в позиційній системі можна представити поліномом

$$A_q = a_k \cdot q^k + a_{k-1} \cdot q^{k-1} + \dots + a_0 \cdot q^0 + a_{-1} \cdot q^{-1} + \dots + a_{-m} \cdot q^{-m} = \sum_{i=-m}^k a_i \cdot q^i, \quad (1.6)$$

де q — основа системи числення, q^i — вага позиції, $a_i \in \{0, 1, \dots, (q-1)\}$ — цифри в позиціях числа, $0, 1, \dots, k$ — номери розрядів цілої частини числа, $-1, -2, \dots, -m$ — номери розрядів дробової частини числа

Позиційні системи з однаковою основою в кожному розряді називаються **однорідними**. Оскільки на значення q немає ніяких обмежень, то теоретично можлива нескінченна множина позиційних систем числення

На практиці застосовують скорочений запис полінома (1.6) у вигляді послідовності цифр із знаком залежно від типу числа

- для змішаного числа

$$A_q = \pm a_k a_{k-1} \dots a_1 a_0, a_{-1} \dots a_{-m}, \quad (1.6, a)$$

- для цілого числа

$$A_q = \pm a_k a_{k-1} \dots a_1 a_0, \quad (1.6, б)$$

- для правильного дробу

$$A_q = \pm 0, a_{-1} a_{-2} \dots a_{-m}. \quad (1.6, в)$$

Приклад 1.1

Ілюстрація запису чисел у вигляді послідовності цифр (1.6) і відповідного полінома (1.5):

- двійкова система: $q = 2; a_i \in \{0, 1\}$:

$$A_2 = 111,01 = 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} = 7,25_{10};$$

- вісімкова система: $q = 8; a_i \in \{0, 1, 2, \dots, 7\}$:

$$A_8 = 45,21 = 4 \cdot 8^1 + 5 \cdot 8^0 + 2 \cdot 8^{-1} + 1 \cdot 8^{-2} = 37,26510;$$

- десяткова система: $q = 10; a_i \in \{0, 1, 2, \dots, 9\}$:

$$A_{10} = 135,64 = 1 \cdot 10^2 + 3 \cdot 10^1 + 5 \cdot 10^0 + 6 \cdot 10^{-1} + 4 \cdot 10^{-2}.$$

- шістнадцяткова система: $q = 16; a_i = \{0, 1, \dots, F\}$:

$$\begin{aligned} A_{16} = DE,1B &= D \cdot 16^1 + E \cdot 16^0 + 1 \cdot 16^{-1} + B \cdot 16^{-2} = \\ &= 13 \cdot 16^1 + 14 \cdot 16^0 + 1 \cdot 16^{-1} + 11 \cdot 16^{-2} = 222,10510. \end{aligned}$$

Розглянуті записи чисел показують один із способів переведення недесяткових чисел у десяткові. При однаковій розрядності в системах числення з більшою основою можна записати більше різних чисел.

Достоїнством двійкової системи є: простота виконання арифметичних операцій; наявність надійних мікроелектронних схем з двома стійкими станами (тригерів), призначених для зберігання значень двійкового розряду — цифр 0 або 1. Двійкові цифри називають також **бітами**. У двійково-десятковій системі числення кожна десяткова цифра записується чотирма двійковими розрядами (**тетрадами**).

Приклад 1.2

Запис десяткового числа в двійково-десятковій системі:

$$A_{10} = 873,25 = \overbrace{1000}^8 \overbrace{0111}^7 \overbrace{0011}^3 \overbrace{0010}^2 \overbrace{0101}^5.$$

1.2.2. Переведення чисел з однієї системи в іншу

Для переведення цілого числа з однієї системи числення в іншу необхідно поділити перевідне число на нову основу за правилами початкової системи. Одержана перша остача є значенням молодшого розряду в новій системі, а першу частку необхідно знову ділити. Цей процес продовжується аж до появи неподільної частки. Результат записують у порядку, оберненому їхньому одержанню у вигляді формули (1.6, б).

Приклад 1.3

Переведення цілого десяткового числа $A = 118$ у двійкове.

Початкове число	Частка	Остача	
118/2	59	0	
59/2	29	1	
29/2	14	1	
14/2	7	0	
7/2	3	1	
3/2	1	1	
1/2	-	1	

Результат. $A = 118_{10} = 1110110_2$.

Для переведення правильного дробу з однієї системи числення в іншу необхідно, діючи за правилами початкової системи, помножити перевідне число на основу нової системи; від результату відокремити цілу частину, а дробову частину, що залишилася, знову помножити на цю основу. Процес такого множення повторюється до одержання заданого числа цифр. Результат записують як цілі частини добутку в порядку їхнього одержання, які розташовують у вигляді формули (1.6, в).

Приклад 1.4

Переведення правильного десяткового дробу $A = 0,625$ у двійкове число з точністю до четвертого знака:

0,	625	
	2	0;
1	250	
	2	$a_{-1} = 1;$
0	500	
	2	$a_{-2} = 0;$
1	000	
	2	$a_{-3} = 1;$
0	000	$a_{-4} = 0.$

Результат: $A = 0,625_{10} = 0,1010_2$.

Для переведення змішаних чисел у двійкову систему потрібно окремо переводити їхні цілу і дробову частини. У записі результату ціла частина переведення відокремлюється від дробової відповідно до формули (1.6, а).

Приклад 1.5

Переведення змішаного десяткового числа $A = 118,625_{10}$ у двійкове. Використовуючи результати переведення цілого числа 118 у прикладі 1.3 і дробового числа в прикладі 1.4, запишемо результат:

$A = 118,625_{10} = 1110110,1010_2$.

У вісімкових і шістнадцяткових числах основа системи числення кратна степеню двійки: $2^3 = 8$; $2^4 = 16$. Тому переведення цих чисел у двійкові реалізується дуже просто. кожна цифру записують трьома двійковими цифрами (тріадами) для вісімкових чисел і чотирма двійковими цифрами (тетрадами) для шістнадцяткового числа в напрямках вліво і вправо від коми. При цьому крайні незначущі нулі опускаються.

Приклад 1.6

Ілюстрація переведення вісімкових і шістнадцяткових чисел у двійкові:

$$A = 305,42_8 = \overbrace{011}^3 \overbrace{000}^0 \overbrace{101}^5, \overbrace{100}^4 \overbrace{010}^2_2;$$

$$A = 7AB,EF_{16} = \overbrace{0111}^7 \overbrace{1010}^A \overbrace{1011}^B, \overbrace{1110}^E \overbrace{1111}^F_2.$$

Для переведення двійкового числа у вісімкове початкове число розбивають на тріади вліво від коми; відсутні крайні цифри доповнюють нулями. Потім кожен тріаду записують вісімковою цифрою. Аналогічно здійснюють переведення двійкового числа в шістнадцяткове, при цьому виділяють тетради, які замінюються шістнадцятковими цифрами.

Приклад 1.7

Ілюстрація переведення двійкових чисел у вісімкові та шістнадцяткові:

$$A = 110011,100010_2 = \overbrace{110}^6 \overbrace{011}^3, \overbrace{100}^4 \overbrace{010}^2_2 = 63,42_8;$$

$$A = 110011,100010_2 = \overbrace{0011}^3 \overbrace{0011}^3, \overbrace{1000}^8 \overbrace{1000}^8_2 = 33,88_{16}.$$

Вкажемо на правила, які полегшують розрахунки з двійковими числами.

1. Необхідно знати представлення десяткових чисел від нуля до 15 у системах числення з основою $q = 2, 8, 16$ та $2-10$ (табл. 1.2).

Таблиця 1.2

$q = 10$	$q = 2$	$q = 8$	$q = 16$	$q = 2-10$
0	0	0	0	0000
1	1	1	1	0001
2	10	2	2	0010
3	11	3	3	0011
4	100	4	4	0100
5	101	5	5	0101
6	110	6	6	0110
7	111	7	7	0111
8	1000	10	8	1000
9	1001	11	9	1001
10	1010	12	A	0001 0000
11	1011	13	B	0001 0001

Таблиця 1.2. Продовження

q = 10	q = 2	q = 8	q = 16	q = 2-10
12	1100	14	C	0001 0010
13	1101	15	D	0001 0011
14	1110	16	E	0001 0101
15	1111	17	F	0001 0101

2. Необхідно знати десяткові значення чисел 2^k від $k = 0$ до $k = 10$ (табл. 1. 3).

Таблиця 1.3

k	0	1	2	3	4	5	6	7	8	9	10
2^k	1	2	4	8	16	32	64	128	256	512	1024

3. Корисно знати такі співвідношення для двійкових цілих і дробових чисел:

$$A = \underbrace{100\dots 00}_k = 2^k, \text{ наприклад, } A = 1000_2 = 2^3 = 8_{10};$$

$$A = \underbrace{11\dots 11}_k = 2^k - 1, \text{ наприклад, } A = 11111_2 = 2^5 - 1 = 31_{10};$$

$$A = 0, \underbrace{11\dots 11}_k = 1 - 2^{-k}, \text{ наприклад, } A = 0,1111_2 = 1 - 2^{-4} = \left(\frac{15}{16}\right)_{10};$$

$$A = 0, \underbrace{00\dots 01}_k = 2^{-k}, \text{ наприклад, } A = 0,00001_2 = 2^{-5} = \left(\frac{1}{32}\right)_{10}.$$

Двійкові дробові числа в десятковому виді записуються так: k розрядів справа від коми визначають ціле додатне число в чисельнику, а в знаменнику — число 2^k . Наприклад, $A = 0,1011_2 = (11/2^4)_{10} = (11/16)_{10}$, що читається як одинадцять шістнадцятих.

1.2.3. Спеціальні системи числення

Розглянуті позиційні системи числення відносять до класичних. Крім них, в комп'ютерах використовують ряд спеціальних позиційних двійкових систем, в тому числі:

- систему з використанням символів $0, -1$, або $1, \bar{1}$; для зручності символ -1 позначають як $\bar{1}$; систему з символами 1 та $\bar{1}$ називають **системою числення $(1, \bar{1})$** ;
- системи з від'ємною основою $q < -1$ і символами $0, 1, \dots, (q - 1)$; систему з основою $q = -2$ і символами $0, 1$ називають **мінус-двійковою**;
- надлишкову систему з основою q і кількістю символів більше q , наприклад, при $q = 2$ використовують символи $(1, 0, \bar{1})$, таку систему називають також **симетричною знакорозрядною**.

Перевагою спеціальних систем числення є спрощення і прискорення виконання ряду арифметичних операцій (додавання, віднімання, множення, ділення) та представлення єдиним кодом додатних і від'ємних чисел без додаткового знакового розряду. Недоліком спеціальних систем є складність правил переводу їх в класичні системи числення і навпаки, а також неоднозначне представлення ряду чисел.

Якщо у формулі (1.6) врахувати, що символи a , набувають значення 1 або $\bar{1}$, то

для основи $q = 2$ число $A = 99_{10}$ матиме вид 1111111 . В системі чисел $(1, \bar{1})$ немає символу "нуль", тому деякі цілі й дробові числа в ній не можуть бути представлені скінченною множиною символів. Наприклад, десяткове число $A = 20_{10}$ в системі чисел $(1, \bar{1})$ записується як нескінченне число $11111, \bar{1}\bar{1}\bar{1}\bar{1}\dots$, що обумовлює похибку в системі $(1, \bar{1})$. В той же час є числа, які не мають єдиного зображення.

У мінус-двійковій системі може бути представлено будь-яке додатне чи від'ємне число. Наприклад, для $q = -2$, $n = 4$ з урахуванням формули (1.6): $A = 5_{10} = 0101_{(-2)}$; $A = -5_{10} = 1111_{(-2)}$. Деякі числа представлені неоднозначно і тільки нескінченними дробами: $A = 1/3 = 0,010101\dots$ або $A = 1,101010\dots$

Надлишкова двійкова система пов'язана із звичайною співвідношенням:

$$\underbrace{11\dots11}_k {}_2 = \sum_{i=0}^{k-1} 2^i = 2^k - 1 = \underbrace{100\dots01}_k.$$

На основі виразу (1.6) здійснюється перехід від звичайної двійкової системи до надлишкової із символами 1, 0, $\bar{1}$ і навпаки. Надлишкова система має такі властивості:

- одне і те саме число не має єдиного запису: $A=12_{10} = 01100_2 = 10\bar{1}00_2 = 11100$;
- додатні і від'ємні числа зображуються без використання додаткового знакового розряду: $A = 13_{10} = 01101_2 = 10\bar{1}01$; $A = -13_{10} = -01101_2 = \bar{1}010\bar{1}$ (при зміні знака числа достатньо замінити в записі числа 1 на $\bar{1}$ і навпаки);
- можливе зменшення кількості одиниць в записі числа, що дозволяє спростити і прискорити виконання арифметичних операцій, особливо додавання і віднімання.

Перетворення двійкових чисел у надлишкову систему з мінімальним числом одиниць в розрядах виконують так. На першому етапі всі комбінації в записі числа виду $01\dots11$ замінюють рівнозначною комбінацією виду $10\dots01$ з тим же числом розрядів; потім виконують заміну сполучень 11 на 01 та $\bar{1}\bar{1}$ на $0\bar{1}$, наприклад:

Кількість одиниць у записі числа зменшилась з десяти до чотирьох.

$$A = 10 \underbrace{0111}_1 \underbrace{0011101101}_3 {}_2 = 10100\bar{1}0100\bar{1}0\bar{1}1 = 10100\bar{1}01000\bar{1}01.$$

1.2.4. Вибір системи числення

У комп'ютерах в основному використовують однорідні позиційні системи числення. При виборі основи q цих систем враховують такі показники.

1. Наявність фізичних елементів для зображення цифр системи у вигляді одного із q станів, наприклад, різниці напруг. Зменшення числа станів спрощує фізичний елемент, тому найбільш прийнятною є двійкова система.
2. Економічність системи числення. Система з більшою основою q забезпечує представлення певного числа меншою кількістю розрядів. Але при цьому ускладнюється побудова фізичного елемента з більшим числом станів.

Ефективність системи числення з основою q , оцінюється кількістю цифрових розрядів D , необхідних для зображення певного числа з довжиною n , тобто $D_i = q_i n_i$. При цьому враховується, що зменшення значення D_i при-

зводить до зменшення електронних схем для представлення чисел в комп'ютерах.

Для представлення будь-якого десяткового числа довжиною, наприклад, $n_{10} = 6$ використовують $D_{10} = q_{10} \cdot n_{10} = 10 \cdot 6 = 60$ цифророзрядів. У двійковій системі для представлення того ж числа (з врахуванням співвідношення $n_2 = 3 \cdot n_{10}$) потрібно мати $D_2 = q_2 \cdot 3 \cdot n_{10} = 2 \cdot 3 \cdot 6 = 36$ цифрових розрядів.

Найбільш економічною є система з основою $q = 2,73 \dots \approx 3$. Двійкова система економічно поступається трійковій на 5,8%, проте має надійніші фізичні елементи. Крім того, для запам'ятовування цифр трійкової системи 0, 1, 2 використовують два двійкових фізичних елементи. Із цього виходить, що найефективнішою є двійкова система числення.

3. Трудомісткість і швидкодія виконання арифметичних операцій. Чим менша основа q , тим менше цифр бере участь в обчисленні даних і тим вища швидкодія комп'ютера. Наприклад, швидкодія машини в двійковій системі перевищує швидкодію в трійковій на 26,2%, а в десятковій — у 2,7 раза.
4. Наявність формального математичного апарату для аналізу і синтезу цифрових схем. Таким апаратом для двійкових елементів є булева алгебра.

Таким чином, з перерахованих показників видно, що найприйнятнішою для застосування в комп'ютерах є однорідна позиційна двійкова система числення. Двійкові системи числення використовують у великих і середніх комп'ютерах, призначених для розв'язання науково-технічних задач з великим об'ємом обчислень і порівняно малою кількістю початкових даних.

Двійково-десяткову систему застосовують для розв'язання економічних задач, які характеризуються великим об'ємом вхідних і вихідних даних порівняно з малим об'ємом розрахунків.

Двійково-десяткова система має такі достоїнства:

- не потрібне переведення початкових даних з однієї системи в іншу;
- зручність контролю результатів зображення їх на екрані дисплея;
- зручність автоматичного контролю через наявність надлишкових кодів у зображенні цифр: 1010, 1011, ..., 1111.

1.2.5. **Форми, діапазон і точність представлення чисел**

Двійкові числа в комп'ютерах розміщуються в комірках пам'яті або в регістрах, які складаються із запам'ятовуючих елементів — тригерів. У комірці або тригері зберігається значення одного двійкового розряду — біт інформації. **Розрядною сіткою комп'ютера** називається сукупність запам'ятовуючих елементів для розміщення одного двійкового числа. Для різних класів комп'ютерів довжина розрядної сітки складає 8, 16, 32, 64 і більше розрядів. **Форматом** називається спосіб розміщення компонентів числа у розрядній сітці, тобто послідовність і позиції знака, мантиси, порядку та ін.

В комп'ютерах використовують дві форми представлення числа:

- з фіксованою комою перед старшим розрядом числа (для правильного дробу) або після молодшого (для цілого числа);
- з плаваючою комою, місце положення якої задається порядком числа.

Місце коми в обох форматах розуміється неявно, без використання додаткових

розрядів. За традицією нумерація розрядів у великих машинах проводиться зліва направо, а в міні- і мікрокомп'ютерах та мікропроцесорах — справа наліво.

У числах із знаком виділяють додатковий знаковий розряд (звичайно, крайній зліва). У ньому для знака плюс записують цифру 0, а для знака мінус — цифру 1. У числах без знака всі розряди числа визначають модуль. Формати двійкових дробових і цілих чисел із знаком, з нумерацією розрядів справа наліво та значенням їх ваг показані на рис. 1.6.

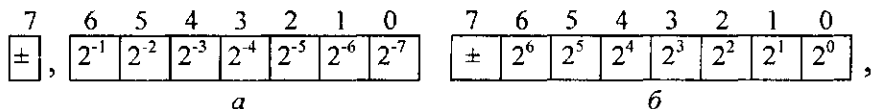


Рис. 1.6. Формат байта: *a* — для правильного дробу; *б* — для цілого числа

Числа в розрядну сітку записують відповідно до ваг позицій. Не зайняті розряди заповнюють нулями, як це показано на рис. 1.7 для чисел $+0,725_{10} = +0,10111_2$ та $-47_{10} = -101111_2$.

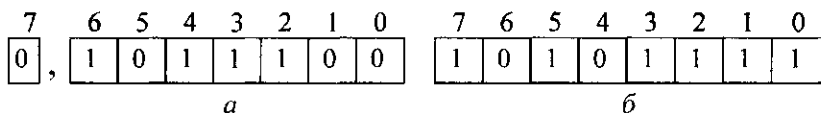


Рис. 1.7. Запис чисел: *a* — дробу $+0,10111_2$; *б* — цілого числа -101111_2

Інтервал числової осі, замкнутий симетрично між максимальним і мінімальним значенням числа, називається **діапазоном представлення**. Абсолютне значення цілого числа A змінюється в межах:

$$A_{\min} \leq |A| \leq A_{\max}; \quad A_{\min} = 1; \quad A_{\max} = 2^k - 1, \quad (1.7)$$

де k — число розрядів цифрової частини числа.

Із виразу (1.7) випливає, що числа, за абсолютним значенням менші одиниці, сприймаються як "машинний нуль"; числа, більші за A_{\max} , викликають переповнення розрядної сітки. Діапазон представлення цілих чисел з урахуванням симетрії відносно нуля числової осі

$$D_A = 2A_{\max} = 2(2^k - 1) \approx 2^{k+1}. \quad (1.8)$$

Приклад 1.8

Розрахувати максимальне значення і діапазон представлення цілого числа із знаком у 16-розрядній сітці. На основі виразів (1.7) і (1.8) для $k = 15$ одержуємо:

$$A_{\max} = 2^{15} - 1 = 32767; \quad D = 2A_{\max} = 65534$$

Абсолютне значення правильного дробу B змінюється в межах:

$$B_{\min} \leq |B| \leq B_{\max}; \quad B_{\min} = 2^{-k}; \quad B_{\max} = 1 - 2^{-k}. \quad (1.9)$$

Правильний дріб, за модулем менший за 2^{-k} , сприймається як "машинний нуль"; числа, більші за одиницю, викликають переповнення розрядної сітки. Діапазон представлення правильного дробу

$$D_B = 2B_{\max} = 2(1 - 2^{-k}) \approx 2,$$

тому що $1 \gg 2^{-k}$.

Точність представлення чисел у комп'ютері визначається абсолютною і відносною похибкою. Абсолютна похибка представлення чисел у комп'ютері складає для цілих чисел

$$\Delta A = 1/2 = 2^{-1}; \quad (1.10)$$

для правильного дробу

$$\Delta B = 2^{-k} / 2 = 2^{-(k+1)}. \quad (1.11)$$

Із виразу (1.10) випливає, що абсолютна похибка представлення цілих чисел у формі з фіксованою комою дорівнює половині молодшого розряду і не залежить від довжини розрядної сітки. Із виразу (1.11) видно, що абсолютна похибка представлення правильного дробу у формі з фіксованою комою зменшується при збільшенні довжини розрядної сітки.

Відносна похибка δ представлення числа задається відношенням абсолютної похибки до поточного значення числа і знаходиться в межах

$$\delta_{\min} \leq \delta \leq \delta_{\max}.$$

Для цілих чисел у формі з фіксованою комою мінімальне і максимальне значення відносної похибки

$$\begin{aligned} \delta_{A_{\min}} &= \Delta A / A_{\max} = 2^{-1} / (2^k - 1) \approx 2^{-(k+1)}; \\ \delta_{A_{\max}} &= \Delta A / A_{\min} = 2^{-1} / 1 = 0,5. \end{aligned}$$

Такими ж формулами визначаються мінімальне і максимальне значення відносної похибки для правильного дробу з фіксованою комою. Відносна похибка представлення чисел у формі з фіксованою комою істотно залежить від значення числа і значно зростає для малих чисел.

У формі з плаваючою комою числа представляються у вигляді добутків:

$$C = q^{\Pi} \cdot M = X \cdot M, \quad (1.12)$$

де q — основа системи числення (зазвичай цілий степінь числа 2); Π — порядок числа довжиною $k+1$ (ціле число зі знаком); M — мантиса числа довжиною $r+1$ (правильний дріб зі знаком); X — характеристика числа. Знак усього числа визначається знаком мантиси. Для міні-комп'ютерів основи порядку і мантиси збігаються (далі передбачається цей випадок), а для великих машин — вони різні. Представлення числа формулою (1.12) називають також напівлогарифмічним, тому що тільки частина числа — характеристика — подається в логарифмічній формі

Мантиса називається нормалізованою, якщо її значення визначається нерівністю виду

$$1/q \leq |M| < 1, \quad (1.13)$$

тобто в старшому розряді модуля мантиси має бути записана одиниця. Так, для основи $q = 2$ маємо: $2^{-1} \leq M < 1$.

Приклад 1.9

Ілюстрація запису числа у формі з плаваючою комою:

$$A_2 = 2^1 \cdot 110,111 = 2^2 \cdot 11,0111 = 2^4 \cdot 0,110111.$$

Значення порядку вказує на кількість позицій, на яку "плаває" кома.

Формат числа з плаваючою комою в 16-розрядній сітці показаний на рис. 1.8. Тут для модулів порядку і мантиси відведено відповідно п'ять і дев'ять розрядів. Кома в порядку розташовується (умовно) після молодшого розряду, а в мантиси — перед старшим. Знаки порядку і мантиси розміщуються перед їхніми старшими розрядами.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
±	Порядок					±	Мантиса								

Рис. 1.8. Формат чисел у формі з плаваючою комою

Абсолютне значення числа C у формі з плаваючою комою з урахуванням формул (1.7), (1.9) і (1.13) змінюється в межах

$$X_{\min} M_{\min} = C_{\min} \leq |C| \leq C_{\max} = X_{\max} M_{\max}, \quad (1.14)$$

де $X_{\min} = 2^{-\Pi_{\max}}$; $M_{\min} = 2^{-1}$; $X_{\max} = 2^{+\Pi_{\max}}$; $M_{\max} = 1 - 2^{-k}$; $\Pi_{\max} = 2^r - 1$; r і k — розрядність модулів порядку і мантиси.

Діапазон представлення чисел у формі з плаваючою комою

$$D_c = 2C_{\max} = 2 \cdot 2^{+\Pi_{\max}} (1 - 2^{-k}) \approx 2^{+\Pi_{\max}+1}. \quad (1.15)$$

Приклад 1.10

Розрахувати значення діапазону D_c для числа C при $r = 5$, $k = 9$. З урахуванням виразу (1.15) одержуємо:

$$\Pi_{\max} = 2^r - 1 = 2^5 - 1 = 31; \quad D_c = 2^{31+1} = 2^{32},$$

що приблизно відповідає десятковому числу $10^{32 \cdot 0.3} \approx 10^9$. Діапазон представлення чисел з плаваючою комою приблизно більший в Π_{\max} разів за діапазон представлення чисел у формі з фіксованою комою.

Абсолютна похибка представлення чисел у формі з плаваючою комою залежить від похибки мантиси і порядку числа:

$$\Delta C = \Delta M \cdot 2^{+\Pi}; \quad \Delta M = 2^{-(k+1)}, \quad (1.16)$$

де ΔM — похибка представлення мантиси.

Мінімальна і максимальна відносні похибки представлення чисел у формі з плаваючою комою не залежать від характеристики (вона записується у чисельнику і знаменнику виразу і тому скорочується). З урахуванням формул (1.14) і (1.16) відносні похибки розраховують із співвідношень:

$$\begin{aligned} \delta C_{\min} &= \frac{\Delta M}{M_{\max}} = \frac{2^{-(k+1)}}{1 - 2^{-k}} \cong 2^{-k+1}; \\ \delta C_{\max} &= \frac{\Delta M}{M_{\min}} = \frac{2^{-(k+1)}}{2^{-1}} = 2^{-k}. \end{aligned} \quad (1.17)$$

Із виразів (1.17) випливає, що відносні похибки представлення чисел у формі з плаваючою комою практично сталі у всьому діапазоні чисел.

1.2.6. Представлення інформації в мікропроцесорах класу Pentium

Розглянемо представлення операндів у 32-розрядних мікропроцесорах класу Pentium. У них використовуються такі типи даних: цілі числа, дійсні числа, двійково-десяткові числа і рядки бітів, байтів та слів. Цілі числа представляються зі знаком і без знаку в форматах байта, слова, подвійного і учетвереного слова довжиною відповідно 8, 16, 32 і 64 біт (рис. 1.9).



Рис. 1.9. Формати цілих чисел: а, в, д, е — зі знаком; б, г, в — без знаку

Над числами у цих форматах виконують операції додавання, віднімання, множення, ділення та ін. Діапазони представлення знакових і беззнакових значень в даних форматах наведені в табл. 1.4.

Таблиця 1.4

Формат числа	Діапазон представлення чисел	
	без знаку	зі знаком
Байт	0...255	-128...+127
Слово	0...65535	-32768...+32767
Подвійне слово	0... $4 \cdot 10^9$	$-2 \cdot 10^9$... $+2 \cdot 10^9$
Учетверенне слово	0... $5 \cdot 10^{19}$	$-2,5 \cdot 10^{19}$... $+2,5 \cdot 10^{19}$

Дійсні числа представляються у форматі з плаваючою комою в короткому (32 біт), довгому (64 біт) і розширеному (80 біт) форматах (рис. 1.10). Числа з плаваючою комою довжиною 32 і 64 біт, які використовуються в багатьох комп'ютерах, наприклад, в ЄС EОМ, СМ EОМ, звичайно називаються числами з одинарною і подвійною точністю. Розширений формат характерний тільки для процесорів класу Pentium.

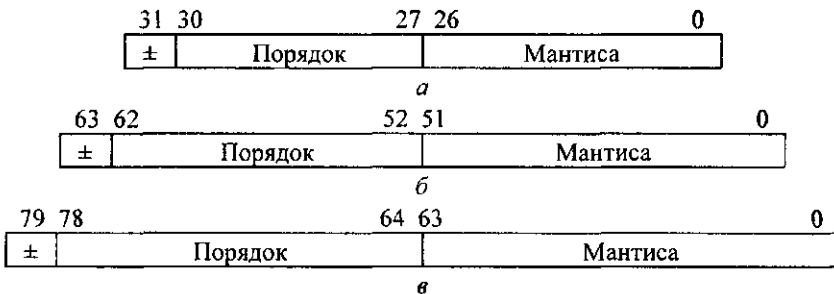


Рис. 1.10. Формати чисел з плаваючою комою: а — короткий; б — довгий; в — розширений

У даних форматах чисел з плаваючою комою використовуються зміщені порядки Π_{3M}

$$\Pi_{3M} = \Pi - \Delta\Pi = \Pi + 2^{r-1} - 1 = \begin{cases} \Pi + 127 & \text{для } r = 8; \\ \Pi + 1023 & \text{для } r = 11; \\ \Pi + 16383 & \text{для } r = 15. \end{cases}$$

де Π — значення істинного порядку; $\Delta\Pi = 2^{r-1} - 1$ — зміщення; r — довжина порядку, яка дорівнює 8, 11 або 15 біт відповідно для короткого, довгого і розширеного формату. Значення Π_{3M} завжди додатне, тому знаковий розряд не потрібний. Представлення порядку зі зміщенням спрощує операції порівняння чисел з плаваючою комою, що особливо важливо для алгоритмів сортування.

Значення числа з плаваючою комою і зміщеним порядком визначається за формулою

$$C = (-1)^S 2^{n - \Delta\Pi} (F_0, F_1, \dots, F_i, \dots, F_n),$$

де S — знак числа; n — число, яке для різних форматів дорівнює 23, 52 чи 63.

В машині мантиса представляється в нормалізованій формі, яка складається із цілої частини $F_0 = 1$ та дробу в такому вигляді:

$$M = 1, F_1 F_2, \dots, F_i, \dots, F_n.$$

У короткому і довгому форматах біт F_0 при передачі чисел і зберіганні їх у пам'яті не фігурує. Це — прихований (неявний) біт, який в нормалізованому числі завжди дорівнює одиниці.

Приклад 1.11

Представити десяткове число $-247,375$ в короткому форматі. Двійковий код цього числа дорівнює -11110111.011 ; істинний порядок буде $+7$ (кома зсувається вліво на сім розрядів), а зміщення досягне значення $\Pi_{3M} = 127 + 7 = 134$. Із врахуванням прихованого біта $F_0 = 1$ маємо.

Знак	Порядок	Мантиса
1	1000 0110	, 1110 1110 1100 0000 0000 0000.

Параметри форматів дійсних чисел наведені в табл. 1.5.

Таблиця 1.5

Параметри	Формат		
	короткий	довгий	розширений
Довжина формату, біт	32	64	80
Довжина мантиси, біт	24	53	64
Довжина порядку, біт	8	11	15
Зміщення порядку	+127	+1023	+16383
Діапазон	$10^{\pm 38}$	$10^{\pm 308}$	$10^{\pm 4932}$

Числа в короткому і довгому форматах існують тільки в пам'яті. При завантаженні чисел в одному з цих форматів у мікропроцесор вони автоматично перетворюються в 80-бітний формат, який використовується тільки для внутрішніх операцій. Аналогічно дані з процесора перетворюються в короткий або довгий формати для зберігання їх у пам'яті.

Точність обчислень чисел з плаваючою комою зростає зі збільшенням довжини мантиси. Діапазон представлення чисел з плаваючою комою залежить від довжини порядку і основи числення q . В машинах ЕС ЕОМ значення $q = 16$. В процесорах Pentium діапазон представлення чисел в короткому форматі для $q = 2$ знаходиться в границях $10^{\pm 38}$. Якщо ж для цього формату взяти основу $q = 16$, то одержимо значення діапазону в границях $10^{\pm 152}$.

У мікропроцесорах Pentium використовуються двійково-десяткові цифри в таких форматах:

- восьмирозрядні упаковані, що містять в одному байті дві десяткові цифри в кодї 8421, наприклад, $39_{10} = 0011\ 1001_{2-10}$;
- восьмирозрядні неупаковані, які містять одну десяткову цифру в байті (молдша тетрада) разом з ознакою (зоною) 0011_2 в міжнародному кодї ASCII, наприклад, $39_{10} = 0011\ 0011\ 0011\ 1001_{2-10}$.

Представлення десяткового числа 136492 в неупакованому і упакованому форматах показано на рис. 1.11;

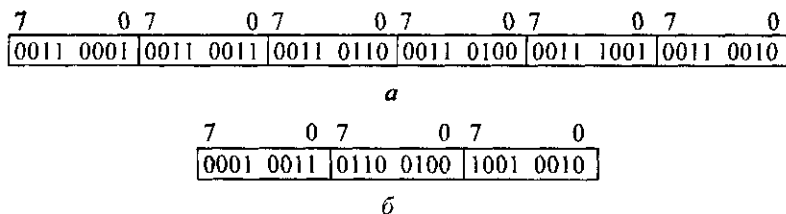


Рис. 1.11. Представлення десяткових цифр в форматах:

a — неупакованому; *б* — упакованому.

- 80-розрядні упаковані, в які записується 19 десяткових цифр, і в старшу тетраду записується знак числа.

1.2.7. Кодування від'ємних чисел

Для записування знака числа, заміни операції віднімання чисел додаванням їхніх кодів, а також для визначення переповнення розрядної сітки використовують прямий, обернений і доповняльний коди, де для представлення знака числа відводиться знаковий розряд, який розташовується зліва від числа і відокремлюється комою. У знаковий розряд записують нуль (для додатного числа) або одиницю (для від'ємного). Кома в машині у явному вигляді не зображується, а тільки припускається.

Числа, представленні в прямому, оберненому і доповняльному кодах, називають **машинними зображеннями**. Вони складаються зі знакового розряду і цифрової частини (модуля числа). Додатні числа у всіх кодах записуються однаково. Якщо позначити машинні зображення числа A в прямому кодї $[A]_{\text{пр}}$, в оберненому кодї $[A]_{\text{об}}$, в доповняльному кодї $[A]_{\text{д}}$, то для додатних чисел маємо:

$$[A]_{\text{пр}} = [A]_{\text{об}} = [A]_{\text{д}}$$

Для від'ємних чисел з довжиною мантиси k бітів кодї визначають за такими правилами:

прямий код.

$$[A]_{\text{пр}} = \begin{cases} 1 + |A| & \text{-- для дробу;} \\ 2^k + |A| & \text{-- для цілого числа;} \end{cases}$$

обернений код:

$$[A]_{\text{об}} = \begin{cases} 2 - |A| - 2^{-k} & \text{-- для дробу} \\ 2^{k-1} - |A| & \text{-- для цілого числа} \end{cases}$$

доповняльний код.

$$[A]_{\text{д}} = \begin{cases} 2 - |A| & \text{-- для дробу} \\ 2^{k-1} - |A| & \text{-- для цілого числа} \end{cases}$$

Приклад 1.12

Запис двійкових чисел $A = 1010_2$ і $B = -1010_2$ у прямих кодах:

$$[A]_{\text{пр}} = 0,1010; [B]_{\text{пр}} = 1,1010.$$

Обернений код від'ємного числа утворюється з його прямого коду після інвертування значень розрядів цифрової частини, тобто заміною нуля на одиницю і одиниці — на нуль; значення знакового розряду не змінюється. Доповняльний код від'ємного двійкового числа утворюється з його оберненого коду додаванням одиниці до молодшого розряду.

Приклад 1.13

Представлення в оберненому і доповняльному кодах двійкових чисел: цілого $A = -1001_2$ і дробового $B = -1001_2$:

$$[A]_{\text{об}} = 1,0110; [A]_{\text{д}} = 1,0111;$$

$$[B]_{\text{об}} = 1,0110; [B]_{\text{д}} = 1,0111.$$

Із цих прикладів видно, що обернений і доповняльний коди цілих і дробових чисел за видом запису збігаються, розходження між ними відображені в алгоритмах обробки інформації

У модифікованих кодах знак числа дублюється в двох знакових розрядах

Приклад 1.14

Представлення в модифікованих кодах чисел $A = 1101_2$ і $B = -1101_2$:

$$[A]_{\text{пр}} = [A]_{\text{об}} = [A]_{\text{д}} = 00,1101;$$

$$[B]_{\text{пр}} = 11,1101; [B]_{\text{об}} = 11,0010; [B]_{\text{д}} = 11,0011.$$

Для переходу від оберненого коду від'ємного числа до прямого коду треба інвертувати значення розрядів цифрової частини, не змінюючи значення знакового розряду. Для переходу від доповняльного коду від'ємного числа до прямого спочатку одержують його обернений код, а потім додають одиницю до молодшого розряду.

1.2.8. Арифметичні операції додавання і віднімання

Всі операції в комп'ютері виконуються в арифметико-логічному пристрої (АЛП). Числа, які беруть участь в операціях, називаються **операндами**. Основною операцією в АЛП є додавання. Операція віднімання замінюється додаванням операндів в оберненому або доповняльному кодах. Операції множення і ділення зводяться до багатократних додавань і зсувів.

Правила виконання операцій додавання, віднімання, множення і додавання за модулем 2 у двійковій арифметиці наведені в табл. 1.6. При додаванні двох одиниць виникає перенесення у старший розряд; при відніманні від нуля одиниці потрібна позика із старшого розряду.

Таблиця 1.6

Додавання	Віднімання	Множення	Модуль 2
$0+0=0$	$0-0=0$	$0\cdot0=0$	$0+0=0$
$0+1=1$	$1-0=1$	$0\cdot1=0$	$0+1=1$
$1+0=1$	$1-1=0$	$1\cdot0=0$	$1+0=1$
$1+1=10$	$0-1=11$	$1\cdot1=1$	$1+1=0$
↑	↑		
Перенесення	Позика		

Операція додавання в АЛП виконується типовим функціональним вузлом — **суматором**. Він складається з окремих логічних схем, які називаються однорозрядними суматорами. Зв'язок між окремими однорозрядними суматорами реалізується за допомогою ланцюгів переносу в напрямку від молодших розрядів до старших.

Операнди A і B надходять з виходів регістрів операндів. Комбінаційний суматор будується на логічних елементах і не має властивості запам'ятовування інформації, тому результат додавання записується в регістр результату.

У всіх комп'ютерах є команди додавання і віднімання чисел. Проте в суматорах реалізуються тільки операції додавання умовно додатних машинних зображень. Машинні зображення додатних операндів у всіх кодах збігаються. Машинні зображення від'ємних операндів одержують за правилами представлення чисел в оберненому і доповняльному кодах. В операціях віднімання знак другого операнда (який віднімається) автоматично змінюється на протилежний і після цього одержують його машинне зображення. Тому в наступних прикладах розглядаються тільки операції додавання.

У машинних зображеннях (для оберненого і доповняльного кодів) знаковий розряд і цифрова частина числа розглядаються як єдине ціле. Вони однаково беруть участь в операції додавання. При додаванні в обернених кодах перенесення із старшого знакового розряду результату подається на вхід перенесення молодшого розряду (циклічне перенесення). При додаванні в доповняльних кодах перенесення із старшого знакового розряду результату не враховується, тому в суматорі ланцюг циклічного перенесення розривається. Знак результату при додаванні машинних зображень утворюється автоматично.

При додаванні двійкових n -розрядних чисел $A = a_n, \dots, a_i, \dots, a_1$ та $B = b_n, \dots, b_i, \dots, b_1$ результат в кожному розряді визначають за формулами:

$$a_i + b_i + Z_i = S_i + 2 \cdot P_{i+1}$$

$$P_{i+1} = \begin{cases} 0 & \text{при } a_i + b_i \leq 1; \\ 1 & \text{при } a_i + b_i \geq 2; \end{cases}$$

де a_i, b_i — значення i -х розрядів; Z_i — перенесення з молодшого розряду; S_i — результат; P_{i+1} — перенесення у старший розряд

Приклад 1.15

Додати двійкові числа $A = 1010$ і $B = 0011$ в оберненому коді:

$$\begin{array}{r} [A]_{\text{об}} = 0,1010 \\ + [B]_{\text{об}} = 0,0011 \\ \hline [C]_{\text{об}} = 0,1101; \quad C = 1101_2 = 13_{10}. \end{array}$$

Приклад 1.16

Додати двійкові числа $A = -1010$ і $B = -0011$ в оберненому і доповняльному кодах:

$$\begin{array}{r} [A]_{\text{об}} = 1,0101 \\ + [B]_{\text{об}} = 1,1100 \\ \hline 1,0001 \\ \quad \downarrow +1 \\ [C]_{\text{об}} = 1,0010; \quad C = -1101_2 = -13_{10}. \end{array} \qquad \begin{array}{r} [A]_{\text{д}} = 1,0110 \\ + [B]_{\text{д}} = 1,1101 \\ \hline [C]_{\text{д}} = 1,0011; \\ C = -1101_2 = -13_{10}. \end{array}$$

При додаванні чисел одного знаку можливе переповнення розрядної сітки, ознакою чого є розбіжність знака результату зі знаками операндів. В АЛП є спеціальні логічні схеми, які автоматично формують ознаку переповнення.

Приклад 1.17

Додати двійкові числа $A = 1011$ і $B = 0111$ в оберненому і доповняльному кодах:

$$\begin{array}{r} [A]_{\text{об}} = 0,1011 \\ + [B]_{\text{об}} = 0,0111 \\ \hline [C]_{\text{об}} \neq 1,0010; \end{array} \qquad \begin{array}{r} [A]_{\text{д}} = 0,1011 \\ + [B]_{\text{д}} = 0,0111 \\ \hline [C]_{\text{д}} \neq 1,0010. \end{array}$$

У цьому прикладі додавання додатних чисел призвело до додатного переповнення: операнди — додатні, результат — від'ємний.

Приклад 1.18

Додати двійкові числа $A = -1011$ і $B = -0111$ в оберненому і доповняльному кодах:

$$\begin{array}{r} [A]_{\text{об}} = 1,0100 \\ + [B]_{\text{об}} = 1,1000 \\ \hline [C]_{\text{об}} \neq 0,1101; \end{array} \qquad \begin{array}{r} [A]_{\text{д}} = 1,0101 \\ + [B]_{\text{д}} = 1,1001 \\ \hline [C]_{\text{д}} \neq 0,1110. \end{array}$$

У цьому прикладі додавання від'ємних чисел призвело до від'ємного переповнення; операнди — від'ємні, результат — додатний.

Для спрощення виявлення переповнення розрядної сітки використовуються модифіковані коди, для яких знаковий розряд у суматорі дублюється. Додатному переповненню в знакових розрядах відповідають цифри 01, а від'ємному — 10. Значення знакових розрядів 00 відповідає правильному додатному результату, а цифри 11 — від'ємному.

Приклад 1.19

Додати двійкові числа $A = -1010$ і $B = 1100$ в оберненому і доповняльному модифікованих кодах:

$$\begin{array}{r} [A]_{\text{об}}^M = 11,0101 \\ + [B]_{\text{об}}^M = 00,1100 \\ \hline 100,0001 \\ \quad \downarrow +1 \\ [C]_{\text{об}}^M = 00,0010; \end{array}$$

$$C = 0010_2 = 2_{10}.$$

$$\begin{array}{r} [A]_{\text{д}}^M = 11,0110 \\ + [B]_{\text{д}}^M = 00,1100 \\ \hline [C]_{\text{д}}^M = 00,0010; \end{array}$$

$$C = 0010_2 = 2_{10}.$$

Приклад 1.20

Складання чисел $A = -1101$ і $B = -1010$ в оберненому і доповняльному модифікованих кодах:

$$\begin{array}{r} [A]_{\text{об}}^M = 11,0010 \\ + [B]_{\text{об}}^M = 11,0101 \\ \hline [C]_{\text{об}}^M \neq 10,1000; \end{array}$$

$$\begin{array}{r} [A]_{\text{д}}^M = 11,0011 \\ + [B]_{\text{д}}^M = 11,0110 \\ \hline [C]_{\text{д}}^M \neq 10,1001. \end{array}$$

У цих прикладах відбулося від'ємне переповнення в результаті.

1.2.9. Арифметичні операції множення і ділення

Операція множення чисел складається з k циклів, де k — число цифрових розрядів множника. Результат множення i -го розряду множника на множене називається **частковим добутком**, а їхнє послідовне додавання — сумою часткових добутків (СЧД). У кожному циклі аналізується наступна цифра множника: якщо це 1, то до СЧД додається множене, якщо 0, то додавання не відбувається. Цикл завершується зсувом на один розряд множеного відносно СЧД або зсувом СЧД відносно нерухомого множника. Множене і множник розміщуються в розрядній сітці на основі спеціальних схем-регістрів, а СЧД — в суматорі-регістрі.

Множення можна виконувати одним з чотирьох способів:

- з молодших розрядів множника із зсувом його та СЧД вправо при нерухомому множеному (рис. 1.12, а). Старші розряди добутку знаходяться в суматорі, а молодші — в регістрі множника;
- з молодших розрядів множника при зсуві множеного і нерухомій СЧД;
- із старших розрядів множника при зсуві СЧД вліво і нерухомому множеному;
- із старших розрядів множника при зсуві його та СЧД вліво (рис. 1.12, б).

Якщо хоч один із співмножників дорівнює нулю, то добутку зразу присвоюється нульове значення.

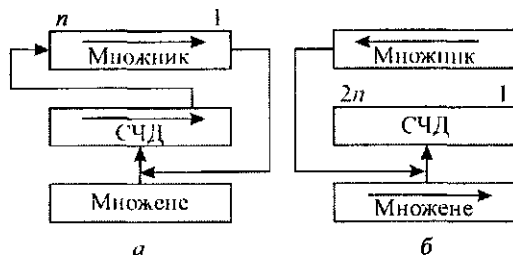


Рис. 1.12. Способи множення: а — перший; б — четвертий

Алгоритм множення цілих чисел, представлених у прямому коді, на основі першого способу реалізується у такій послідовності:

1. взяти модулі співмножників і обнулити початкове значення СЧД;
2. якщо молодша цифра множника дорівнює 1, то до СЧД додається множене, якщо це 0, то додавання не відбувається,
3. зсув вправо на один розряд СЧД та множника;
4. пп 2 і 3 послідовно виконуються k раз, після цього добутку присвоюється знак, який визначається додаванням за модулем 2 знаків співмножників

Приклад 1.21

Перемножити першим способом числа $A = 13_{10} = 1101_2$ і $B = 11_{10} = 1011_2$, представлені прямими кодами

1101	— модуль множеного
1011	— модуль множника
<u>0000</u>	— початкова СЧД
+1101	— перший частковий добуток (ЧД)
1101	— перша СЧД
01101	— зсув вправо СЧД на 1 розряд
<u>+1101</u>	— другий ЧД
100111	— друга СЧД
0100111	— зсув вправо СЧД на 1 розряд
<u>+0000</u>	— третій ЧД
100111	— третя СЧД
0100111	— зсув вправо СЧД на 1 розряд
<u>+1101</u>	— четвертий ЧД
10001111	— четверта СЧД
10001111	— зсув вправо СЧД на 1 розряд.

Добуток $C = A \cdot B = 10001111_2 = 143_{10}$.

Розроблені також методи прискореного множення та множення в доповняльних кодах

Операція ділення числа $C = A / B$ звичайно зводиться до послідовності віднімань дільника B спочатку від діленого A , а потім від створюваних в процесі ділення залишків R_i . Числа A і B подаються в прямому коді. Залежно від способу віднімання дільника розрізняють два алгоритми ділення: без відновлення залишку та з відновленням.

Для обох алгоритмів спільними є:

- ділення проводиться над модулями чисел A і B (при діленні цілих чисел в машинах довжина діленого A вдвоє більша довжини дільника B);
- знак частки визначається додаванням за модулем 2 знаків A і B ;
- перед діленням аналізуються числа A і B на рівність нулю: якщо $A = 0$, то частці присвоюється значення 0 і операція закінчується; при $B = 0$ виконання програми переривається;
- множення частки на два здійснюється зсувом її вліво на один розряд, а операція віднімання дільника B замінюється його додаванням у доповняльному коді;
- перший залишок R_1 визначається за формулою

$$R_1 = 2 \cdot |A| - B = 2 \cdot |A| + [-B]_д,$$

причому, якщо $R_1 < 0$, то ділення можливе, якщо $R_1 > 0$, то виникає переповнення розрядної сітки;

- цифри частки визначаються, починаючи із старших розрядів:

$$C_i = \begin{cases} 0 & \text{при } R_i < 0; \\ 1 & \text{при } R_i \geq 0, \end{cases}$$

$i = 1, 2, \dots, k$, де k — задана кількість цифр частки. Звичайно перший розряд C_i відводиться для записування знака.

Алгоритм ділення без відновлення залишку визначають за таким рекурентним співвідношенням:

$$R_{i+1} = \begin{cases} 2R_i + |B| & \text{при } R_i < 0; \\ 2R_i + [B]_д & \text{при } R_i \geq 0. \end{cases}$$

Алгоритм ділення з відновленням залишку визначають за таким рекурентним співвідношенням:

$$R_{i+1} = \begin{cases} 4R_{i-1} + [-B]_д & \text{при } R_i < 0; \\ 2R_i + [B]_д & \text{при } R_i \geq 0. \end{cases}$$

Приклад 1.22

Поділити числа $A = 49_{10}$ і $B = -7_{10}$ на основі алгоритму без відновлення залишку. Прямий код операндів: $[A]_{пр} = 0,0110001$, $[B]_{пр} = 1,111$; $[-B]_д = 1,001$:

Приклад 1.22. Продовження

$$\begin{array}{r}
 [A]_{\text{ПР}} = 0,0110001 \\
 + [B]_{\text{ПР}} = 1,111 \\
 \hline
 2|A| = 0,1100010 \\
 + \\
 [-B]_{\text{Д}} = 1,001 \\
 \hline
 R_1 = 1,1110010 < 0 \quad C_1 = 0 \\
 2R_1 = 1,1100100 \\
 + \\
 |B| = 0,111 \\
 \hline
 R_2 = 0,1010100 > 0 \quad C_2 = 1; \\
 2R_2 = 1,0101000 \\
 + \\
 [-B]_{\text{Д}} = 1,001 \\
 \hline
 R_3 = 0,0111000 > 0 \quad C_3 = 1; \\
 2R_3 = 0,1110000 \\
 + \\
 [-B]_{\text{Д}} = 1,001 \\
 \hline
 R_4 = 0,0000000 > 0 \quad C_4 = 1;
 \end{array}$$

частка $[C]_{\text{ПР}} = 1,111$, $C = -111 = -7_{10}$;
 залишок дорівнює нулю
 Знак частки $A[7] \oplus B[7] = 0 \oplus 1 = 1$

Приклад 1.23

Поділити числа $A = 49_{10}$ і $B = -7_{10}$ на основі алгоритму з відновленням залишків:

$$\begin{array}{r}
 [A]_{\text{ПР}} = 0,0110001 \\
 + [B]_{\text{ПР}} = 1,111 \\
 \hline
 2|A| = 0,1100010 \\
 + \\
 [-B]_{\text{Д}} = 1,001 \\
 \hline
 R_1 = 1,1110010 < 0 \quad C_1 = 0 \\
 4R_0 = 1,1000100 \\
 + \\
 [-B]_{\text{Д}} = 1,001 \\
 \hline
 R_2 = 0,1010100 > 0 \quad C_2 = 1; \\
 2R_2 = 1,0101000 \\
 + \\
 [-B]_{\text{Д}} = 1,001 \\
 \hline
 R_3 = 0,0111000 > 0 \quad C_3 = 1; \\
 2R_3 = 0,1110000 \\
 + \\
 [-B]_{\text{Д}} = 1,001 \\
 \hline
 R_4 = 0,0000000 > 0 \quad C_4 = 1;
 \end{array}$$

частка $[A]_{\text{ПР}} = 1,111$; $C = -111 = -7_{10}$.

Описані способи виконання арифметичних операцій над числами з фіксованою комою застосовують також для виконання їх над мантисами чисел з плаваючою комою. Додавання і віднімання чисел A і B з плаваючою комою може здійснюватися тільки при рівності їхніх порядків P_A і P_B . Для цього вони попередньо вирівнюються зсувом одного з них. При множенні чи діленні чисел A і B з плаваючою комою порядок результату P_P дорівнює відповідно сумі або різниці порядків цих чисел, тобто $P_P = P_A \pm P_B$.

1.3. Логічні основи комп'ютерної схемотехніки

1.3.1. Булева алгебра

Теоретичною основою комп'ютерної схемотехніки є алгебра логіки — наука, яка використовує математичні методи для розв'язання логічних задач. Алгебру логіки називають булевою на честь англійського математика Дж. Буля, який вніс найбільший вклад у розвиток цієї науки.

Основним предметом булевої алгебри є висловлювання — просте твердження, про яке можна стверджувати: істинне воно (позначають символом 1) або хибне (позначають символом 0). Зазвичай прості висловлювання позначають буквами, наприклад, X_1, X_2, \dots, X_n , які у комп'ютерній схемотехніці називають змінними (аргументами). За допомогою логічних зв'язок НЕ, ЧИ, І, ЯКЩО... ТО... будують складні висловлювання, які називають булевими (логічними) функціями і позначають буквами F, L, K, M, P та ін.

У даний час головна задача алгебри логіки — аналіз, синтез і структурне моделювання будь-яких дискретних скінченних систем. Апарат булевої алгебри поширюється на об'єкти найрізноманітнішої природи безвідносно до їхньої суті, тільки б вони характеризувалися двома значеннями або станами: контакт увімкнений або вимкнений, наявність високого чи низького рівня електричної напруги, виконання або невиконання деякої умови роботи та ін.

Використання апарата алгебри логіки в комп'ютерній схемотехніці засноване на тому, що цифрові елементи характеризуються двома станами і через це можуть бути описані булевими функціями. Стандарт ДСТУ 2533–94 "Арифметичні і логічні операції. Терміни і визначення" конкретизував основні поняття булевої алгебри в системах оброблення інформації.

Змінну із скінченним числом значень (станів) називають **перемикальною**, а з двома значеннями — **булевою**. Функція, яка має як і кожна її змінна скінченне число значень, називається перемикальною (логічною). Логічна функція, число можливих значень якої і кожної її незалежної змінної дорівнює двом, є булевою. Таким чином, булева функція — це окремий випадок перемикальної.

Операція — це чітко визначена дія над одним або декількома операндами, яка створює новий об'єкт (результат). У булевій операції операнди і результат набувають "булевого значення 1" (далі просто значення 1) і "булевого значення 0" (далі просто значення 0). Булеву операцію над одним операндом називають одномісною, над двома — двомісною і т.д.

Булеві функції можуть залежати від однієї, двох і в цілому від n змінних. Запис

$F(X_1, X_2, \dots, X_n)$ означає, що деяка булева функція F залежить від змінних X_1, X_2, \dots, X_n . Основними булевими операціями є заперечення (операція НЕ, інверсія), диз'юнкція (операція ЧИ, логічне додавання, об'єднання) і кон'юнкція (операція І, логічне множення).

Заперечення — це одномісна булева операція $F = \overline{X}$ (читається "не X "), результатом якої є значення, протилежне значенню операнда.

Диз'юнкція — це булева операція $F = X_1 \vee X_2$ (читається " X_1 чи X_2 "), результатом якої є значення нуль тоді і тільки тоді, коли обидва операнди мають значення нуль.

Кон'юнкція — це булева операція $F = X_1 \cdot X_2$ (читається " X_1 і X_2 "), результатом якої є значення одиниця тоді і тільки тоді, коли значення кожного операнда дорівнює одиниці. У виразі $X_1 \cdot X_2$ крапку можна опускати; часто застосовують запис $X_1 \wedge X_2$ або $X_1 \& X_2$.

Операції заперечення, диз'юнкції і кон'юнкції можна задати за допомогою таблиць істинності (табл. 1.7 і 1.8), у яких зліва подані значення операндів, а справа — значення булевої функції.

Таблиця 1.7

X	$F = \overline{X}$
0	1
1	0

Таблиця 1.8

X_1	X_2	$F = X_1 \vee X_2$	$F = X_1 \cdot X_2$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

В табл. 1.8. булеві функції ЧИ, І задані для двох змінних X_1, X_2 .

Для булевих операцій заперечення, диз'юнкції і кон'юнкції справедливі такі закони, властивості й тотожності:

- комутативність (переміщувальний закон):

$$X_1 \vee X_2 = X_2 \vee X_1; \quad X_1 \cdot X_2 = X_2 \cdot X_1;$$

- асоціативність (сполучний закон):

$$X_1 \vee X_2 \vee X_3 = (X_1 \vee X_2) \vee X_3; \quad X_1 \cdot X_2 \cdot X_3 = (X_1 \cdot X_2) \cdot X_3;$$

- дистрибутивність (розподільний закон):

$$X_1 (X_2 \vee X_3) = X_1 \cdot X_2 \vee X_1 \cdot X_3; \quad X_1 \vee X_2 \cdot X_3 = (X_1 \vee X_2) (X_1 \vee X_3);$$

- ідемпотентність (виключення повторення):

$$X \vee X \vee X = X; \quad X \cdot X \cdot X = X;$$

- закон поглинання:

$$X_1 \vee X_1 \cdot X_2 = X_1; \quad X_1 (X_1 \vee X_2) = X_1;$$

- закон склеювання:

$$X_1 \cdot X_2 \vee X_1 \cdot \overline{X_2} = X_1; \quad (X_1 \vee X_2) (X_1 \vee \overline{X_2}) = X_1;$$

- закон де Моргана:

$$\overline{X_1 \vee X_2} = \overline{X_1} \cdot \overline{X_2}; \quad \overline{X_1 \cdot X_2} = \overline{X_1} \vee \overline{X_2};$$

- властивості заперечення і констант:

$$X \vee \overline{X} = 1; \quad X \cdot \overline{X} = 0; \quad \overline{\overline{X}} = X; \quad \overline{1} = 0; \quad \overline{0} = 1;$$

$$X \vee 0 = X; \quad X \vee 1 = 1; \quad X \cdot 1 = X; \quad X \cdot 0 = 0;$$

- тотожності:

$$X_1 \vee \overline{X_1} \cdot X_2 = X_1 \vee X_2; \quad X_1 (\overline{X_1} \vee X_2) = X_1 \cdot X_2.$$

Справедливість наведених законів булевої алгебри перевіряється підстановкою в логічний вираз нуля і одиниці, як показано в табл. 1.9 для формули $X_1 X_2 = X_2 \vee X_2$.

Таблиця 1.9

X_1	X_2	$X_1 \cdot X_2$	$\overline{X_1 X_2}$	$\overline{X_1}$	$\overline{X_2}$	$\overline{X_1} \vee \overline{X_2}$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

Областю визначення булевої функції $F(X_1, X_2, \dots, X_n)$ є скінченна множина різних двійкових наборів довжиною n , на кожному з яких вказується значення функції нуля або одиниця. Кількість різноманітних двійкових наборів дорівнює множині n -розрядних двійкових чисел $m = 2^n$. Наприклад, для функції двох змінних X_1 і X_2 є чотири двійкових набори: $\langle 0, 0 \rangle$; $\langle 0, 1 \rangle$; $\langle 1, 0 \rangle$; $\langle 1, 1 \rangle$.

Часто набори нумеруються десятковими еквівалентами двійкових чисел від нуля до $2^n - 1$. Наприклад, для $n = 4$, набори $\langle 0, 1, 0, 1 \rangle$ і $\langle 1, 0, 0, 1 \rangle$ мають відповідно номери 5 і 9. Дві функції відрізняються одна від одної, якщо їхні значення будуть різними хоч би на одному наборі. Число різноманітних булевих функцій від n змінних дорівнює 2^m , де $m = 2^n$.

Довільну булеву функцію можна задавати різними способами: словесним описом, часовими діаграмами, геометричними фігурами, графами, таблицями істинності та аналітичними виразами.

Словесний опис деякої булевої функції $F(X_1, X_2)$ можна представити так: $F = 1$ при $X_1 X_2 = 1$ і $F = 0$, якщо $X_1 X_2 = 0$. Таку функцію можна зобразити часовою діаграмою (рис. 1.13, а) або геометрично за допомогою двовимірного куба (рис. 1.13, б), у якому точками виділені одиничні вершини (дана функція набуває значення одиниці на наборі $\langle 1, 1 \rangle$), а також графом, де вершини відображають значення нуля і одиниці, а на орієнтованих дугах змінні вказують на умови переходів (рис. 1.13, в).

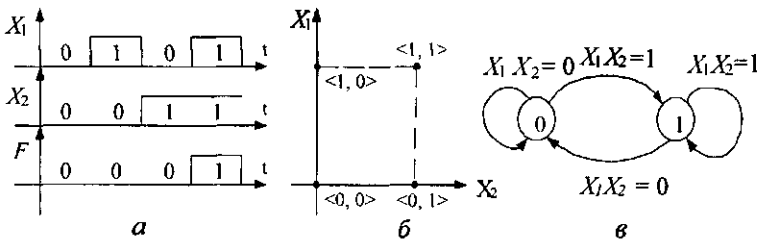


Рис. 1.13. Способи задавання булевих функцій

За допомогою таблиць істинності показують усі можливі функції однієї змінної (усього чотири функції) і двох змінних (усього 16 функцій). Для $n = 3$ число можливих булевих функцій дорівнює 256, для $n = 4$ їхня кількість – $2^{16} = 65536$.

Булеві функції однієї змінної подані в табл. 1.10. Як бачимо, з чотирьох булевих функцій практичний інтерес викликає тільки операція заперечення $L_2 = \overline{X}$.

Таблиця 1.10

L_i	X_i		Вираз	Назва операції
	0	1		
L_0	0	0	$L_0 = 0$	Константа 0
L_1	0	1	$L_1 = X$	Повторення X
L_2	1	0	$L_2 = \overline{X}$	Заперечення X
L_3	1	1	$L_3 = 1$	Константа 1

Всі 16 булевих функцій $F_0 - F_{15}$ двох змінних X_1, X_2 наведені в табл. 1.11.

Таблиця 1.11

X_1	0 0 1 1	Вираз	Назва операції
X_2	0 1 0 1		
F_0	0 0 0 0	$F_0 = 0$	Константа 0
F_1	0 0 0 1	$F_1 = X_1 X_2$	Кон'юнкція
F_2	0 0 1 0	$F_2 = X_1 \overline{X_2}$	Заборона по X_2
F_3	0 0 1 1	$F_3 = X_1$	Повторення X_1
F_4	0 1 0 0	$F_4 = \overline{X_1} X_2$	Заборона по X_1
F_5	0 1 0 1	$F_5 = X_2$	Повторення X_2
F_6	0 1 1 0	$F_6 = X_1 \oplus X_2$	Сума за модулем 2
F_7	0 1 1 1	$F_7 = X_1 \vee X_2$	Диз'юнкція
F_8	1 0 0 0	$F_8 = X_1 \downarrow X_2$	Заперечення диз'юнкції
F_9	1 0 0 1	$F_9 = X_1 \sim X_2$	Еквівалентність
F_{10}	1 0 1 0	$F_{10} = \overline{X_2}$	Заперечення X_2
F_{11}	1 0 1 1	$F_{11} = X_1 \leftarrow X_2$	Імплікація від X_2 до X_1
F_{12}	1 1 0 0	$F_{12} = \overline{X_1}$	Заперечення X_1
F_{13}	1 1 0 1	$F_{13} = X_1 \rightarrow X_2$	Імплікація від X_1 до X_2
F_{14}	1 1 1 0	$F_{14} = X_1 / X_2$	Заперечення кон'юнкції
F_{15}	1 1 1 1	$F_{15} = 1$	Константа 1

Як впливає з табл. 1.11, функції F_0 і F_{15} — константи, F_3 і F_5 — повторюють, а F_{10} і F_{12} — заперечують одну із змінних, F_1 і F_7 — кон'юнкція і диз'юнкція, які розглянуті раніше. До нових булевих функцій (операцій) відносяться такі.

Виключення (заборона) — двомісна булева операція, результатом якої є значення одиниця тоді і тільки тоді, коли значення одного операнда дорівнює одиниці, а іншого — нулю. Записується у вигляді:

$$F_2 = X_1 \overline{X_2} \text{ або } F_4 = \overline{X_1} X_2.$$

Сума за модулем два (виключальне ЧИ, заперечення еквівалентності) — двомісна булева операція, результатом якої є значення одиниця тоді і тільки тоді, коли операнди мають різні значення. Позначається у вигляді:

$$F_6 = X_1 \oplus X_2 = \overline{X_1} X_2 \vee X_1 \overline{X_2}.$$

Заперечення диз'юнкції (операція НЕ ЧИ, стрілка Пірса) — булева операція, результатом якої є значення одиниця тоді і тільки тоді, коли обидва операнди дорівнюють нулю. Позначається у вигляді:

$$F_8 = X_1 \downarrow X_2 = \overline{X_1 \vee X_2}.$$

Узагальнюючи для n змінних, маємо:

$$X_1 \downarrow X_2 \downarrow X_3 \dots \downarrow X_n = \overline{X_1 X_2 \dots X_n} = \overline{X_1 \vee X_2 \vee X_3 \vee \dots \vee X_n}.$$

Еквівалентність (рівнозначність) — двомісна булева операція, результатом якої є одиниця тоді і тільки тоді, коли операнди набувають однакових значень. Позначається у вигляді:

$$F_9 = X_1 \sim X_2 = X_1 X_2 \vee \overline{X_1} \overline{X_2}.$$

Імплікація (включення) — двомісна булева операція, результатом якої є значення нуль тоді і тільки тоді, коли значення одного з операндів дорівнює нулю, а іншого — одиниці. Позначається у вигляді:

$$F_{11} = X_1 \leftarrow X_2 = X_1 \vee \overline{X_2}; \quad F_{13} = X_1 \rightarrow X_2 = \overline{X_1} \vee X_2.$$

Заперечення кон'юнкції (операція НЕ І, штрих Шеффера, заперечення перетину) — булева операція, результат якої дорівнює нулю тоді і тільки тоді, коли обидва операнди дорівнюють одиниці. Позначається у вигляді:

$$F_{14} = X_1 / X_2 = \overline{X_1 X_2}.$$

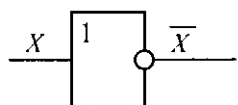
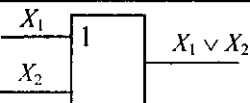
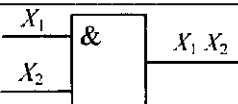
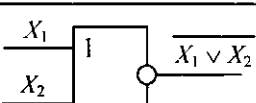
Узагальнюючи для n змінних, маємо:

$$X_1 / X_2 / \dots X_n = \overline{X_1 \vee X_2 \vee \dots X_n} = \overline{X_1 X_2 \dots X_n}.$$

Булеві функції одного і двох аргументів називають **елементарними**. Схему, яка здійснює елементарну логічну операцію, називають **логічним елементом (вентилем)**. Сукупність взаємозалежних логічних елементів з формальними методами опису називається **логічною схемою**.

Назви і умовні графічні позначення основних логічних елементів, які застосовуються в комп'ютерній схемотехніці, наведені в табл. 1.12. Значення змінних (операндів) відображаються електричними сигналами з двома чітко вираженими рівнями значень.

Таблиця 1.12

Назва операції	Назва елемента	Умовне графічне позначення
Заперечення	НЕ	
Диз'юнкція	ЧИ	
Кон'юнкція	І	
Заперечення диз'юнкції	НЕ ЧИ	

Таблиця 1.12. Продовження

Назва операції	Назва елемента	Умовне графічне позначення
Заперечення кон'юнкції	НЕ І	
Заперечення еквівалентності	Виключальне ЧИ	
Еквівалентність	Еквівалентність	
Імплікація	ЯКЩО, ТО	
Заборона	ЗАБОРОНА	

За допомогою суперпозицій, тобто підстановки в логічні формули замість змінних деяких інших булевих виразів, можна одержати складніші функції будь-якого числа змінних, наприклад,

$$Y = X_1 \vee X_2; \quad X_1 = Z_1 Z_2; \quad X_2 = Z_3 \vee Z_4, \quad \text{тоді } Y = Z_1 Z_2 \vee Z_3 \vee Z_4.$$

Однією з інтерпретацій булевих операцій є схеми, які складаються з ключів, джерела напруги E і лампочки L . Для реалізації операції диз'юнкції двох змінних X_1 і X_2 використовують два паралельно з'єднаних нормально розімкннутих ключі (рис. 1.14, а).

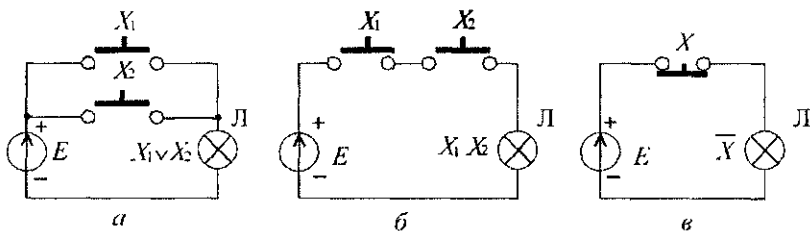


Рис. 1.14. Інтерпретація булевих операцій:
а — диз'юнкція; б — кон'юнкція; в — заперечення

При натисканні будь-якого ключа ($X_1 = 1$ або $X_2 = 1$) або обох разом лампочка горить (значення 1). Для реалізації операції кон'юнкції двох змінних X_1 і X_2 застосовують два послідовно з'єднаних нормально розімкннутих ключі (рис. 1.14, б). При натисканні одночасно обох ключів ($X_1 = X_2 = 1$) лампочка горить (значення 1). Для реалізації операції заперечення застосовують нормально замкнутий ключ (рис. 1.14, в). При $X = 0$ ключ замкнутий і лампочка горить; при $X = 1$ ключ розмикається і лампочка не горить.

1.3.2. Аналітичне представлення булевих функцій

Розроблено універсальні (канонічні) форми представлення булевих функцій, які дають можливість одержати аналітичну форму довільної функції безпосередньо з таблиці істинності. Ця форма надалі може бути мінімізована або спрощена. Оскільки між множиною аналітичних представлень і множиною схем, які реалізують цю функцію, є взаємно однозначна відповідність, то пошук канонічної форми запису є початковим етапом синтезу логічних схем. Найбільше поширення одержали досконала диз'юнктивна нормальна форма (ДДНФ) і досконала кон'юнктивна нормальна форма (ДКНФ). Для одержання цих форм вводяться поняття мінтермів (конституента 1) і макстермів (конституента 0).

Мінтерм — це функція n змінних, яка дорівнює одиниці тільки на одному наборі. Мінтерм одержують як кон'юнкцію n змінних, що входять до нього у прямому виді, якщо значення даної змінної в наборі $X_i = 1$, i — із запереченням, якщо $X_i = 0$. При n змінних є 2^n мінтермів m_0, m_1, \dots, m_R , де $R = 2^n - 1$. Всі мінтерми двох змінних наведені в табл. 1.13.

Таблиця 1.13

X_1	X_2	F_9	f_i	Мінтерми	Макстерми
0	0	1	$f_0=1$	$m_0 = \overline{X_1} \overline{X_2}$	$M_0 = X_1 \vee X_2$
0	1	0	$f_1=0$	$m_1 = \overline{X_1} X_2$	$M_1 = X_1 \vee \overline{X_2}$
1	0	0	$f_2=0$	$m_2 = X_1 \overline{X_2}$	$M_2 = \overline{X_1} \vee X_2$
1	1	1	$f_3=1$	$m_3 = X_1 X_2$	$M_3 = \overline{X_1} \vee \overline{X_2}$

Значення функції F_9 , які відповідають, згідно з таблицею істинності, кожному i -му наборові, позначені через f_0, f_1, f_2, f_3 . Представлення функції F_9 у ДДНФ є диз'юнктивною сумою мінтермів, які відповідають наборам змінних, для яких $f_i = 1$:

$$\begin{aligned} F_9 &= f_0 \cdot m_0 \vee f_1 \cdot m_1 \vee f_2 \cdot m_2 \vee f_3 \cdot m_3 = 1 \cdot m_0 \vee 0 \cdot m_1 \vee 0 \cdot m_2 \vee 1 \cdot m_3 = \\ &= m_0 \vee m_3 = \overline{X_1} \overline{X_2} \vee X_1 X_2. \end{aligned}$$

Макстерм — це функція n змінних, яка дорівнює нулю тільки на одному наборі. Макстерм одержують як диз'юнкцію усіх змінних, що входять до нього у прямому вигляді, коли значення $X_i = 0$, або в інверсному вигляді, якщо значення $X_i = 1$. Число макстермів дорівнює 2^n , для функції двох змінних вони наведені в табл. 1.13. Представлення функції F_9 у ДКНФ записується у вигляді:

$$\begin{aligned} F_9 &= (f_0 \vee M_0) (f_1 \vee M_1) (f_2 \vee M_2) (f_3 \vee M_3) = \\ &= (1 \vee M_0) (0 \vee M_1) (0 \vee M_2) (1 \vee M_3) = M_1 M_2 = (X_1 \vee \overline{X_2}) (\overline{X_1} \vee X_2). \end{aligned}$$

Пояснимо на прикладі табл. 1.14 аналітичний запис функції трьох змінних у ДДНФ і ДКНФ. Для записування функції P в ДДНФ потрібно диз'юнктивно скласти ті мінтерми, для яких функція дорівнює одиниці:

$$P = \overline{X_1} \overline{X_2} X_3 \vee \overline{X_1} X_2 X_3 \vee X_1 \overline{X_2} \overline{X_3} \vee X_1 X_2 \overline{X_3} \vee X_1 X_2 X_3.$$

Таблиця 1.14

X_1	X_2	X_3	P
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Для записування функції P в ДДКФ необхідно записати кон'юнкцію макстермів, для яких функція дорівнює нулю:

$$P = (X_1 \vee X_2 \vee X_3) (X_1 \vee \overline{X_2} \vee X_3) (\overline{X_1} \vee X_2 \vee \overline{X_3}).$$

За цим способом виконують запис у ДДНФ і ДКНФ функцій з довільним числом змінних.

Система функцій, суперпозицією яких може бути представлена будь-яка булева функція, називається **функціонально повною**; вона утворює базис у логічному просторі.

Систему функцій називають **мінімально повним базисом**, якщо видалення з неї будь-якої функції перетворює цю систему в неповну. В теорії алгебри логіки доведено, що функціонально повні системи утворюють такі набори функцій:

1. $\overline{X}, X_1 \vee X_2, X_1 X_2$ (булевий базис, надмірний);
2. $\overline{X}, X_1 \vee X_2$;
3. $\overline{X}, X_1 X_2$;
4. $\overline{X_1 X_2}$;
5. $\overline{X_1 \vee X_2}$ та ін.

Інша алгебра логіки будується на основі функції суми за модулем два і кон'юнкції (алгебра Жегалкіна). Через операції алгебри Жегалкіна можна висловити усі інші булеві функції:

$$\begin{aligned} X_1 \rightarrow X_2 &= \overline{X_1} \vee X_2 = 1 \oplus X_1 \oplus X_1 X_2; \\ X_1 \sim X_2 &= (\overline{X_1} \vee X_2)(X_1 \vee \overline{X_2}) = 1 \oplus X_1 \oplus X_2; \\ X_1 \leftarrow X_2 &= \overline{X_1} \rightarrow X_2 = X_1 \oplus X_1 X_2. \end{aligned}$$

У булевій алгебрі широко використовується розкладання Шенона — формула, яка дозволяє переходити до представлення функції n змінних через функції від $(n-1)$ змінних:

$$F(X_1, X_2, \dots, X_n) = X_1 f(1, X_2, \dots, X_n) \vee \overline{X_1} f(0, X_2, \dots, X_n).$$

Вираз легко узагальнюється для будь-якого числа змінних, якщо обидві функції його правої частини піддати такому ж розкладанню за іншими змінними. Зазначимо, якщо зробити розкладання за усіма змінними, утвориться ДДНФ.

1.3.3. Мінімізація булевих функцій

Важливим етапом проектування комп'ютерних схем є мінімізація булевих функцій, тобто знаходження їхніх виражень з мінімальним числом букв. Мінімізація забезпечує побудову економічних схем комп'ютерів. Для мінімізації функцій із числом букв $n \leq 6$ застосовують карти Карно. Їх будують у вигляді таблиць з 2^n клітинок з розміткою рядків і стовпчиків змінними. Карти Карно для функції трьох змінних $F(X_1, X_2, X_3)$ показані на рис. 1.15. Рядки карти позначені значеннями змінної X_1 , а стовпчики — значеннями змінних X_2, X_3 . Кожна клітинка карти Карно однозначно відповідає одному наборові таблиці істинності для функції трьох змінних (рис. 1.15, а) або мінтермам цієї функції (рис. 1.15, б). Клітинки карти Карно часто нумерують десятковими цифрами — номерами наборів (рис. 1.15, в).

При мінімізації для кожного мінтерму, який входить у ДДНФ функції, ставиться одиниця, а інші клітинки не заповнюються. Наприклад, заповнення карти Карно для функції, заданої табл. 1.14, показано на рис. 1.15, г.

Мінтерми в сусідніх клітинках карти Карно в рядку (з врахуванням верхніх і нижніх) або в стовпчику (з врахуванням крайніх) розрізняються значенням однієї змінної, що дозволяє виконувати операцію склеювання по цій змінній.

Наприклад, на рис. 1.15, г мінтерми $\overline{X_1}X_2X_3$ і $X_1X_2X_3$ (клітинки з номерами 1 і 3) відрізняються значенням змінної X_1 , тому вони склеюються по ній і представляються кон'юнкцією двох змінних X_2X_3 . Аналогічно для мінтермів $X_1\overline{X_2}\overline{X_3}$ і $X_1X_2\overline{X_3}$ (номери клітинок 4 і 6) склеювання відбувається по змінній X_2 і одержують кон'юнкцію $X_1\overline{X_3}$. У результаті мінімізації функції $P(X_1, X_2, X_3)$ одержують її мінімальне вираження $P = \overline{X_1}X_3 \vee X_2X_3 \vee X_1\overline{X_3}$.

$X_1 \backslash X_2 X_3$	00	01	11	10
0	000	001	011	010
1	100	101	111	110

а

$X_1 \backslash X_2 X_3$	00	01	11	10
0	$\overline{X_1}\overline{X_2}\overline{X_3}$	$\overline{X_1}\overline{X_2}X_3$	$\overline{X_1}X_2X_3$	$\overline{X_1}X_2\overline{X_3}$
1	$X_1\overline{X_2}\overline{X_3}$	$X_1\overline{X_2}X_3$	$X_1X_2X_3$	$X_1X_2\overline{X_3}$

б

$X_1 \backslash X_2 X_3$	00	01	11	10
0	0	1	3	2
1	4	5	7	6

в

$X_1 \backslash X_2 X_3$	00	01	11	10
0		1	1	
1	1		1	1

г

Рис. 1.15. Карти Карно для функції трьох змінних

Наведемо загальні правила мінімізації.

1. Зображають карту Карно для n змінних і роблять розмітку її рядків і стовпчиків. У клітинки таблиці, які відповідають мінтермам (одичинним наборам) функції, що мінімізується, записують одиницю.
2. Склеюванню підлягають прямокутні конфігурації, які заповнені одиницями і містять 2, 4 або 8 клітинок. Верхні й нижні рядки, крайні ліві і праві стовпчики карти ніби склеюються, створюючи поверхню циліндра.

- Множина прямокутників, які покривають усі одиниці, називається **покриттям**. Чим менше прямокутників і чим більше клітинок у прямокутниках, тим краще покриття. З декількох варіантів вибирають той, у якого менший коефіцієнт покриття $z = r / s$, де r — загальне число прямокутників, s — їхня сумарна площа в клітинках. Наприклад, для зображеного покриття (рис. 1.15,а) маємо $z = 3/5$.
- Формули, отримані в результаті мінімізації, містять r елементарних кон'юнкцій (за числом прямокутників у покритті). Кожна кон'юнкція містить тільки ті змінні, які не змінюють свого значення в наборах, що склеюються у відповідному прямокутнику. Число змінних у кон'юнкції називається її **рангом**. При склеюванні двох сусідніх клітинок одержують ранг кон'юнкції $n-1$, чотирьох клітинок — $n-2$, восьми клітинок — $n-3$ і т.д.

Розмітка карт Карно для функцій чотирьох змінних показана на рис. 1.16. В клітинки карт записані значення мінтермів відповідно у двійковому і десятковому еквівалентах.

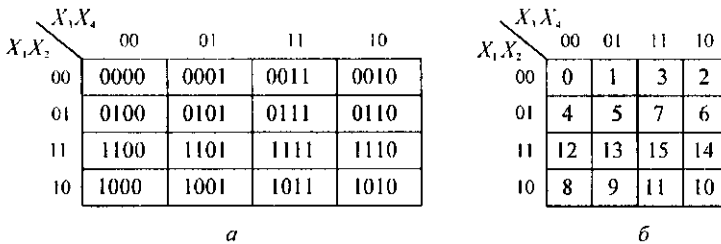


Рис. 1.16. Карти Карно для функцій чотирьох змінних:
а — двійкове значення мінтермів; б — десяткове значення мінтермів

Приклад 1.25

Мінімізувати булеві функції L_1, L_2 і L_3 чотирьох змінних. Функція L_1 задана двома способами — в ДДНФ і з десятковими еквівалентами мінтермів:

$$L_1 = \bar{X}_1\bar{X}_2\bar{X}_3\bar{X}_4 \vee \bar{X}_1\bar{X}_2\bar{X}_3X_4 \vee \bar{X}_1\bar{X}_2X_3\bar{X}_4 \vee \bar{X}_1\bar{X}_2X_3X_4 \vee \bar{X}_1X_2\bar{X}_3\bar{X}_4 \vee X_1\bar{X}_2\bar{X}_3\bar{X}_4 \vee X_1X_2\bar{X}_3\bar{X}_4 \vee X_1X_2X_3X_4 = 0 \vee 1 \vee 2 \vee 3 \vee 4 \vee 8 \vee 12 \vee 15.$$

Функції L_2 і L_3 для спрощення задані десятковими еквівалентами мінтермів:

$$L_2 = 2 \vee 3 \vee 6 \vee 7 \vee 8 \vee 9 \vee 12 \vee 13; \quad L_3 = 1 \vee 3 \vee 5 \vee 7 \vee 9 \vee 11 \vee 13 \vee 15.$$

Мінімізація функцій L_1, L_2 і L_3 проводиться на основі карт Карно, показаних на рис. 1.17. Результати мінімізації:

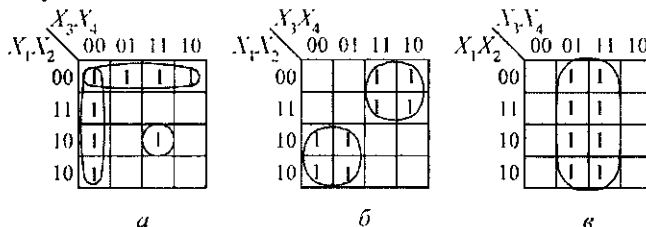


Рис. 1.17. Карти Карно для мінімізації булевих функцій чотирьох змінних:
а — L_1 ; б — L_2 ; в — L_3

Приклад 1.25. Продовження

$$L_1 = \overline{X_1} \overline{X_2} \vee \overline{X_3} \overline{X_4} \vee X_1 X_2 X_3 X_4; \quad z_1 = 3/8;$$

$$L_2 = \overline{X_1} X_3 \vee X_1 \overline{X_3}; \quad z_2 = 2/8;$$

$$L_3 = X_4; \quad z_3 = 1/8.$$

Для мінімізації булевих функцій використовують також діаграми Вейча, які аналогічні картам Карно і відрізняються від них способом розмітки: замість символів 0 і 1 використовують булеві позначення аргументів — $X_1, \overline{X_1}, X_2$ та ін. (рис. 1.18).

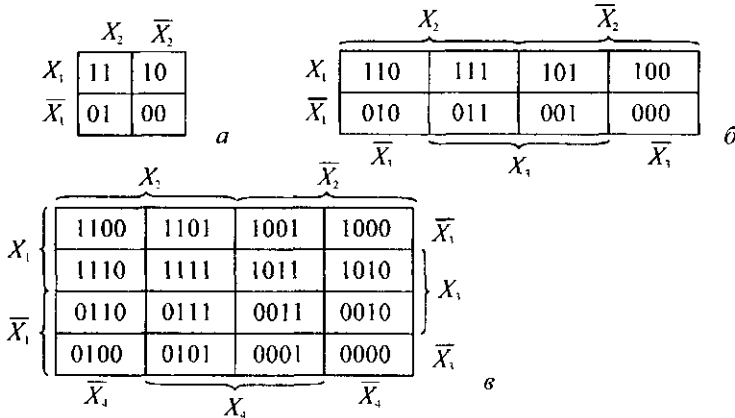


Рис. 1.18. Діаграми Вейча:

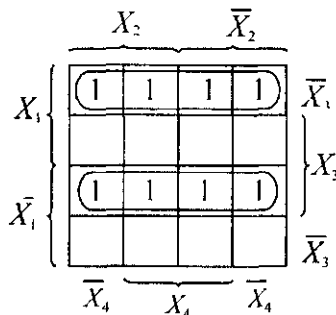
а — для двох змінних; б — для трьох змінних; в — для чотирьох змінних

Приклад 1.26

Мінімізувати за допомогою діаграм Вейча функцію L_2 з прикладу 1.25. Запис значення мінтермів функції L_2 в клітини Вейча показаний на рис. 1.19. Результат мінімізації:

$$L_2 = \overline{X_1} X_3 \vee X_1 \overline{X_3}, \quad z_2 = 1/4$$

збігається з даними прикладу 1.25.

Рис. 1.19. Діаграма Вейча для мінімізації функції L_2

1.4. Основні характеристики цифрових мікросхем

1.4.1. Поняття елементів, вузлів і пристроїв комп'ютерної схемотехніки

Технічні засоби комп'ютерної схемотехніки залежно від функцій, які вони виконують, поділяють на елементи функціональні вузли і пристрої, а також мікропроцесори та комп'ютери (рис. 1.20). Вони призначені для оброблення дискретної інформації і тому називаються цифровими.

Технічні засоби комп'ютерної схемотехніки в даний час основані на інтегральних мікросхемах (ІМС) різного ступеня складності.

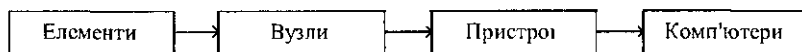


Рис. 1.20 Склад технічних засобів комп'ютерної схемотехніки

Елементами в комп'ютерній схемотехніці називаються найменші неподільні мікроелектронні схеми (вироби), призначені для виконання логічних операцій або зберігання біта інформації. До елементів умовно відносяться і допоміжні схеми — підсилювачі, повторювачі, формувачі та ін. Елементи будуються на основі двопозиційних ключів, що технічно реалізуються найпростіше. Елементи з двома станами називаються **двійковими**.

На входах і виходах двійкового елемента діють напруги, які набувають у сталому режимі двох значень — високого U_H і низького U_L рівнів (індекси від англійських слів *High* і *Low*). Ці напруги відображають електричні сигнали. Сигнал з двома станами називається двійковим. Перехід елемента з одного стану в інший називається його **перемиканням**. На основі елементів будують типові функціональні вузли.

Елементарні дії, які виконуються в комп'ютерах за один машинний такт, називаються **мікроопераціями**. Наприклад, інкремент або декремент слова, зсув, інверсія, додавання та ін. У комп'ютерах команди виконують послідовністю мікрооперацій над двійковими словами (числами). Типовими функціональними вузлами комп'ютерів називаються мікроелектронні схеми, призначені для виконання однієї або декількох мікрооперацій. За логікою роботи функціональні вузли розподіляються на комбінаційні та послідовнісні схеми.

У **комбінаційних схемах** логічний стан виходів елементів залежить тільки від комбінації вхідних сигналів у даний момент часу. До функціональних вузлів комбінаційного типу відносяться суматори, дешифратори, шифратори, мультиплексори і демультіплексори, схеми порівняння (компаратори) і контролю за парністю, кодоперетворювачі.

У **послідовнісних схемах** логічне значення виходів визначають як комбінацією вхідних сигналів, так і станом пам'яті схеми в даний момент часу. До функціональних вузлів послідовнісного типу відносяться регістри, лічильники, генератори чисел і керуючі автомати. На основі типових функціональних вузлів будують різноманітні пристрої комп'ютерів.

Універсальність комп'ютерів забезпечує можливість приймання і видавання інформації, її зберігання та арифметико-логічне опрацювання, а також керування

усім обчислювальним процесом. Ці функції реалізуються відповідними пристроями введення, виведення, запам'ятовування, арифметико-логічними і керування.

В усіх комп'ютерах використовують генератор тактових імпульсів (ГТІ), що виробляє періодичну послідовність прямокутних імпульсів, які називаються тактовими (C). Початок кожного імпульсу C називається **тактовим моментом**. Часовий інтервал між двома сусідніми імпульсами C називається **машинним тактом** T_C . На початку кожного імпульсу C відбувається зміна інформації на входах елементів і вузлів машини (рис. 1.21).

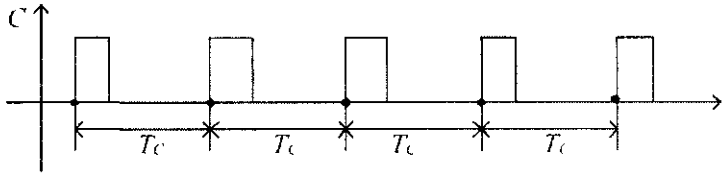


Рис. 1.21. Часова діаграма ГТІ

Частота ГТІ вимірюється десятками і сотнями мегагерц. У літературі ГТІ часто називають генераторами синхронізуючих імпульсів, а самі імпульси називають синхронізуючими або **синхроімпульсами**. Амплітуда і полярність імпульсу C залежить від фізичних принципів побудови машини. Принцип подачі інформації на входи елементів і вузлів у тактові моменти називається **дискретизацією сигналів у часі**.

У комп'ютерній схемотехніці застосовуються два основних види двійкових сигналів: потенціальні й імпульсні (рис. 1.22).

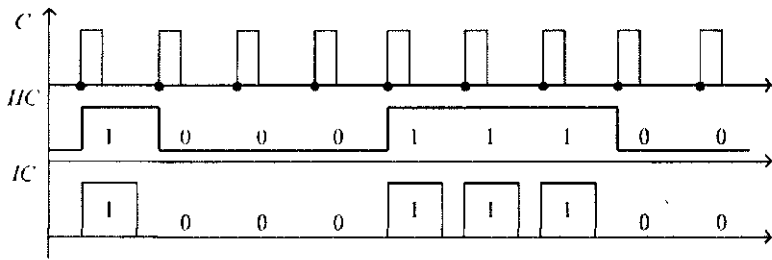


Рис. 1.22. Діаграми потенціальних (PC) та імпульсних (IC) сигналів

Сигнал, який змінюється тільки в тактові моменти часу, називається **потенціальним**. Сигнал, що наростає в тактовий момент, а спадає в границях даного такту, називається **імпульсним**. Тривалість потенціального сигналу дорівнює або кратна тривалості машинного такту.

У логіці значення двійкового сигналу і відповідної змінної X кодуються символами 0 (лог. 0) і 1 (лог. 1). Напрягу, що відображає символ 1, позначимо через U^1 , а символ 0 — через U^0 . Розрізняють два способи кодування логічних сигналів X_i потенціальними сигналами — позитивний та негативний. При позитивному кодуванні (позитивна логіка чи угода) більший рівень напруги U_H з урахуванням знака відображає лог. 1, а менший U_l — лог. 0, тобто $X = 1$, якщо $U^1 = U_H$, та $X = 0$ при $U^0 = U_l$ (рис. 1.23, а). При негативному кодуванні (негативна логіка чи угода) більший рівень напруги U_H з урахуванням знака відображає лог. 0, а менший U_l — лог. 1, тобто $X = 1$, якщо $U^1 = U_l$, та $X = 0$ при $U^0 = U_H$ (рис. 1.23, б).

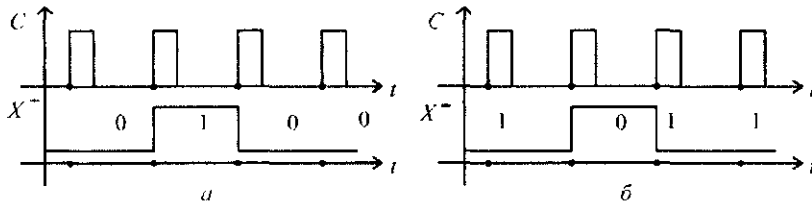


Рис. 1.23. Логічні угоди: а — позитивна X^+ ; б — негативна X^-

Для імпульсних сигналів розрізняють два роди кодування (рис. 1.24): перший — наявність імпульсу відображає лог. 1, відсутність — лог. 0; другий — наявність імпульсу однієї полярності відображає лог. 1, а іншої полярності — лог. 0.

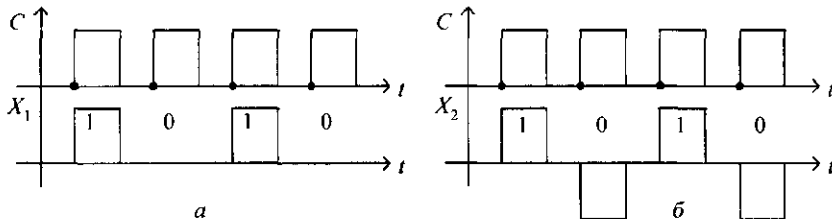


Рис. 1.24. Імпульсне кодування: а — першого роду; б — другого роду

Елементи комп'ютерів класифікуються за такими ознаками:

- використовуваними фізичними приладами;
- видом інформаційних сигналів;
- функціональним призначенням;
- конструкційно-технологічним виготовленням;
- рівнем і ступенем складності.

За типом фізичних приладів розрізняють такі елементи: побудовані на електронних лампах — перше покоління; на транзисторах — друге покоління; на ІМС малого і середнього ступеня інтеграції — третє покоління; на великих і надвеликих ІМС — четверте покоління.

За видом інформаційних сигналів виділяють:

- потенціальні елементи — використовуються тільки потенціальні сигнали;
- імпульсні елементи — використовуються тільки імпульсні сигнали;
- потенціально-імпульсні елементи — використовуються потенціальні й імпульсні сигнали.

За функціональним призначенням елементи комп'ютерної схемотехніки поділяють на такі класи:

- логічні елементи, призначені для виконання логічних операцій — НЕ, І, ЧИ, НЕ І, НЕ ЧИ, НЕ І ЧИ та ін.;
- елементи, які запам'ятовують — тригери, призначені для зберігання значення однієї двійкової змінної — нуля або одиниці, тобто одного біта інформації;
- допоміжні елементи (підсилювачі, формувачі, перетворювачі сигналів, схеми часового узгодження, генератори імпульсів та ін.), призначені для забезпечення роботи елементів перших двох класів.

За конструкційно-технологічним виготовленням елементна база сучасної комп'ютерної схемотехніки складається з **інтегральних мікросхем**. Це — мікроелект-

ронні вироби з високою щільністю упакування електрорадіоелементів (резисторів, діодів, транзисторів) і з'єднань між ними. З погляду специфікації, випробування, постачання та експлуатації ІМС розглядаються як єдине ціле.

В ІМС електрорадіоелементи називаються елементами, якщо вони невіддільні від схеми, і компонентами, якщо їх можна використовувати самостійно. Надалі, щоб уникнути плутанини з елементами комп'ютерів, електрорадіоелементи називають просто **компонентами**.

Мікросхеми класифікують за такими головними ознаками:

- технологією виготовлення — напівпровідникові, гібридні, плівкові,
- конструкційним оформленням — корпусні та безкорпусні
- формою оброблення інформації — аналогові, цифрові й аналого-цифрові
- ступенем інтеграції (складності) — малі, середні, великі, надвеликі й ультра-великі,
- типом активних елементів — побудованих на біполярних і МОН-транзисторах,
- областю застосування — широкого застосування, спеціалізовані, у тому числі замовлені і напівзамовлені,
- використовуваними матеріалами — кремнієві, арсенід-галієві,
- перспективними напрямками — кріомікроелектронні, акустоелектронні, оптикоелектронні, молекулярної електроніки та ін.

Набір цифрових мікросхем із спільними конструкційно-технологічними і схемотехнічними ознаками утворює **серію ІМС**. У комп'ютерній схемотехніці широко застосовуються цифрові напівпровідникові корпусні ІМС на основі кремнію і арсеніду галію. У напівпровідникових ІМС усі компоненти і з'єднання між ними виконані в об'ємі і на поверхні кристала площею від 4 до 100 мм². У гібридних ІМС навісні компоненти кріпляться на поверхні діелектричної підкладки. У плівкових ІМС усі компоненти і з'єднання між ними виконані у виді тонких плівок на діелектричній підкладці.

Складність мікросхем характеризується рівнем інтеграції N , ступенем інтеграції $K = \lg N$ і ступенем функціональної складності $F = \lg L$, де N — число компонентів комп, L — число двохходових логічних елементів (вентилів), значення десятичного логарифма округляється до більшого цілого числа. Промисловість виготовляє ІМС від першого (менше 10 комп) до шостого (менше 1 млн комп) і вище ступеня інтеграції.

Можливості інтегральної технології визначає щільність упакування, відношення числа компонентів до об'єму (іноді до площі) кристала. Щільність упакування в напівпровідникових ІМС складає 10⁷ комп /см³, а для гібридних — 100–200 комп /см³.

Мала інтегральна схема (МІС) вміщує до 100 комп включно, середня мікросхема (СІС) — 100–1000 комп, велика інтегральна схема (ВІС) — до 100000 комп, надвелика інтегральна мікросхема (НВІС) — до 1 млн комп, а ультравелика (ультра-ВІС) — до 10 млн комп і більше. На МІС будують елементи, на СІС реалізують типові вузли, на ВІС, НВІС і ультра-ВІС забезпечують побудову мікропроцесорів і мікрокомп'ютерів.

Кожна елементарна логічна функція реалізується відповідно логічним елементом ЧІ (диз'юнктор), І (кон'юнктор), НЕ (інвертор). Для реалізації складних функцій логічні елементи об'єднуються у логічну схему. Функціонально повна система логічних елементів дозволяє побудувати будь-яку складну логічну схему. Такі системи

утворюються такими наборами логічних елементів 1) ЧИ, НЕ, 2) І, НЕ, 3) НЕ ЧИ, 4) НЕ І та іншими. У технічно повній системі елементів забезпечується значення електричних параметрів двійкових сигналів, для цього використовуються допоміжні елементи — підсилювачі, повторювачі, формувачі та ін.

З урахуванням вищевикладеного можна сказати, що система елементів являє собою функціонально і технічно повний набір елементів, який використовує однакові способи представлення інформації, а також має спільні конструктивно-технологічні характеристики.

1.4.2. Характеристики логічних елементів

Логічні, схемотехнічні й експлуатаційні властивості логічних елементів визначаються сукупністю характеристик і параметрів, до яких відносяться:

- функції логічних елементів,
- логічні угоди,
- коефіцієнти об'єднання за входом і виходом,
- коефіцієнт розгалуження,
- швидкодія,
- потужність споживання,
- робота перемикачів,
- вхідні й вихідні напруги і струми,
- статична і динамічна стійкість до перешкод,
- надійність елементів
- допустимі розміри механічних впливів: діапазони тиску і температури навколишнього середовища, стійкість до радіаційних впливів,
- маса, вартість і конструктивне оформлення.

У більшості випадків зазначені характеристики і параметри відносяться і до ІМС, на яких реалізовані логічні елементи.

Коефіцієнт об'єднання за входом N_i характеризує число логічних входів логічного елемента — зазвичай 1, 2, 3, 4 або 8 (рис. 1.25).

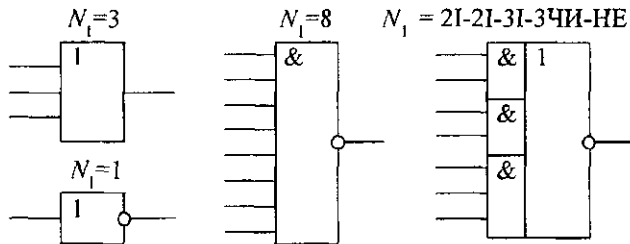


Рис. 1.25. Приклади значень коефіцієнта N_i

Коефіцієнт об'єднання за виходом N_o характеризує допустиму кількість з'єднаних між собою виходів логічних елементів з метою утворення нових функцій.

Коефіцієнт розгалуження за виходом N_p характеризує навантажувальну спроможність логічного елемента, тобто максимальне число входів ідентичних схем, яке може бути одночасно залучене до виходу даного елемента без порушення його працездатності (рис. 1.26). До складу серій ІМС зазвичай входять елементи з малою навантажувальною спроможністю ($N_p = 3 \dots 15$) та з великою ($N_p = 30 \dots 50$).

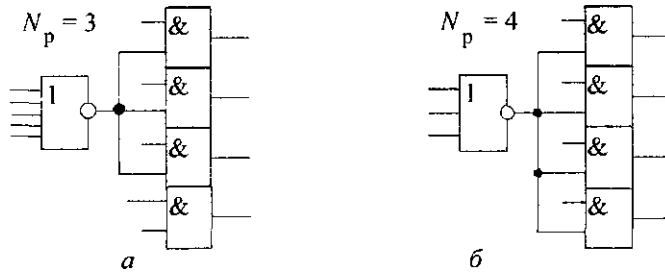


Рис. 1.26. Підключення навантаження: а — $N_p = 3$; б — $N_p = 4$

Прийняті такі визначення і буквені позначення електричних параметрів цифрових мікросхем (ДСТУ 2883-94):

- вхідні U_I і вихідні U_O рівні напруг (індекси — від англійських слів *Input* і *Output*);
- вхідні напруги низького U_{IL} і високого U_{IH} рівнів; для них установлюються максимальне значення низького рівня $U_{IL\max}$ та мінімальне значення високого рівня $U_{IH\min}$ (рис. 1.27, а);
- вихідні напруги низького U_{OL} і високого U_{OH} рівнів; для них установлені максимальне значення низького рівня $U_{OL\max}$ та мінімальне значення високого рівня $U_{OH\min}$ (рис. 1.27, б);
- вхідний I_I і вихідний I_O струми;
- вхідний струм I_{IH} — при низькому рівні напруги на вході, I_{IH} — при високому;
- вихідний струм I_{OL} — при низькому рівні напруги на виході, а I_{OH} — при високому;
- U_{CC} — значення напруги джерела живлення;
- I_{CC} — струм, споживаний ІМС від джерела живлення;
- P_{CC} — потужність, споживана ІМС від джерела живлення;
- вхідні граничні напруги, при яких відбувається перемикання елемента: U_{IHL} — найменше значення для високого рівня і U_{IHL} — найбільше значення для низького рівня.

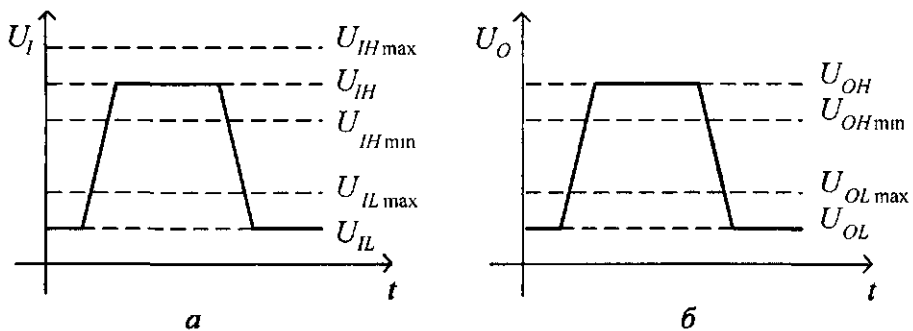


Рис. 1.27. Позначення рівнів напруги: а — вхідних; б — вихідних

Основні параметри логічних елементів визначають за допомогою вхідної, вихідної і передатної характеристик. Типові графіки цих характеристик для інвертувальних елементів транзисторно-транзисторної логіки наведені на рис. 1.28.

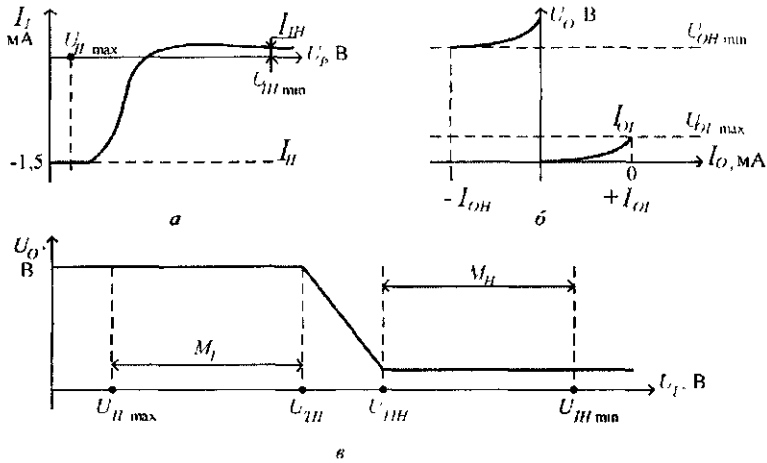


Рис. 1.28. Характеристики логічного транзисторно-транзисторного елемента: а — вхідна; б — вихідна; в — передатна

Вхідна характеристика логічного елемента $I_I = f(U_I)$ — це залежність вхідного струму від зміни вхідної напруги. Струми, що втікають у схему елемента, вважають додатними, а ті, що витікають — від'ємними (рис.1.28, а). З цієї характеристики визначають вхідні струми I_{IH} для напруги $U_{IH, \max}$ і струм I_{IL} для напруги $U_{IH, \min}$. Вихідна характеристика логічного елемента $U_O = f(I_O)$ визначає залежність вихідної напруги від струму навантаження для станів високого і низького рівнів (рис. 1.28, б). Із цієї характеристики визначають допустимі значення струмів: $+I_{OL}$ — при низькому рівні вихідної напруги $U_{OL, \max}$ і $-I_{OH}$ — при високому рівні напруги $U_{OH, \min}$ (рис. 1.28, б).

Передатна характеристика $U_O = f(U_I)$ — це залежність вихідної напруги від вхідної (рис. 1.28, в). З цієї характеристики визначають значення завадостійкості для низького рівня на вході M_L (перешкода, що відкриває) і для високого рівня на вході M_H (перешкода, що закриває):

$$M_L = U_{TH} - U_{IH, \max}; M_H = U_{IH, \min} - U_{TH}$$

Середня споживана потужність P_{CC}^* елементом від джерела живлення обчислюється за формулою

$$P_{CC}^* = U_{CC} (I_{CCL} + I_{CCH}) / 2 = U_{CC} I_{CC}^*$$

де I_{CCL} , I_{CCH} — струми споживання при низькому і високому рівнях напруги на виході відповідно; I_{CC}^* — середній струм споживання. Сучасні елементи споживають потужність від мікровоат до десятків міліватів.

Потенціальні сигнали характеризуються значенням логічного перепаду (амплітудою) $U_M = U_H - U_L$ і тривалістю позитивного t_{WH} та негативного t_{WL} перепадів (рис. 1.29). Перепади напруг часто називають позитивними і негативними імпульсами.

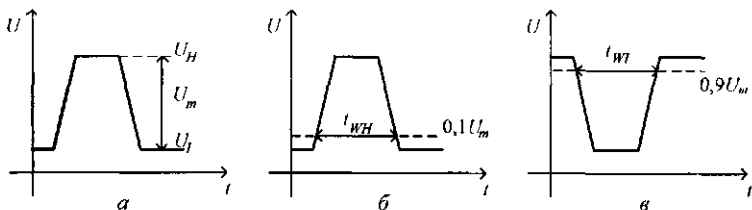


Рис. 1.29. Вимір параметрів сигналу: а — амплітуди; б, в — тривалості перепадів

Для вимірювання часових параметрів сигналу встановлюють умовні рівні в частках від амплітуди — 0,1; 0,5 і 0,9.

Швидкодію мікросхем визначають за значеннями таких тривалостей:

- фронту t_{1H} і спаду t_{HL} (рис. 1.30, а);
- власне вмикання t_{THL} і вимикання t_{TLH} (рис. 1.30, б); та їх затримки відповідно t_{DHL} та t_{DLH} ;
- затримок поширення сигналу при вмиканні t_{PHL} і вимиканні t_{PLH} (рис. 1.30, в).

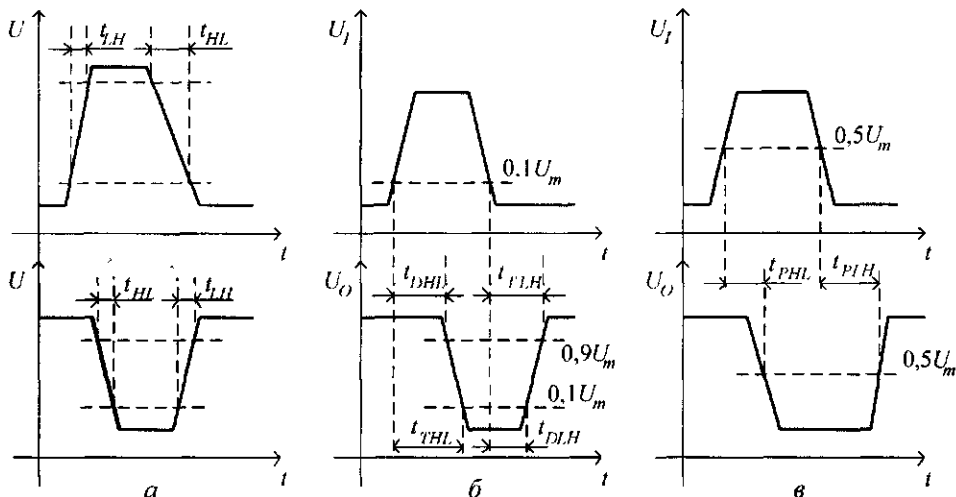


Рис. 1.30. Вимір часових параметрів сигналів:

а — фронту t_{1H} і спаду t_{HL} ; б — часу вмикання t_{THL} і вимикання t_{TLH} ;

в — часу затримок розповсюдження сигналу при вмиканні t_{PHL} та вимиканні t_{PLH}

Для практичних розрахунків використовують середній час затримки поширення сигналу

$$t_p = (t_{PHL} + t_{PLH}) / 2.$$

Для оцінки якості елемента широко використовують узагальнений параметр — роботу перемикання

$$A_{\Pi} = P_{CC}^* \cdot t_p.$$

Якщо потужність P_{CC}^* вимірюється в міліватах, а час затримки — в наносекундах, то робота перемикання A_{Π} виражається в пікоджоулях (пДж). Значення узагальненого параметра A_{Π} знаходиться в межах 0,1–200 пДж. Чим менше значення A_{Π} , тим кращі характеристики має логічний елемент.

Надійність ІМС характеризується трьома взаємозалежними показниками:

- інтенсивністю відмов $\lambda = n / (mt)$, де n — число відмов за час випробування, год; m — загальна кількість випробуваних мікросхем;
- напрацюванням на відмову $T = 1/\lambda$;
- можливістю безвідмовної роботи протягом заданого інтервалу часу $P = \exp(-\lambda t)$.

Для сучасних ІМС інтенсивність відмов $\lambda = (10^{-7} \dots 10^{-8})$. Приймавши, що $\lambda = 10^{-8}$, $t = 15000$, одержимо значення ймовірності безвідмовної роботи $P(t) = 0,998$ або 99,8%.

За конструктивно-технологічним виконанням мікросхеми поділяються на п'ять груп, яким присвоєні такі позначення (ДСТУ 3212–95)

- напівпровідниковим на біполярних транзисторах — 1, 6,
- напівпровідниковим на польових транзисторах — 5, 7,
- пбридним — 2, 4,
- іншим (плівковим, вакуумним, керамічним тощо) — 3,
- резервним — 0 8, 9

Зазначені групи мікросхем за конструктивно-технологічним виконанням поділяються на підгрупи, яким надаються такі позначення

- для першої групи комбіновані структури з біполярними та польовими транзисторами — 0 структура на біполярних транзисторах з ізоляцією *p-n* переходами чи діелектриком — відповідно 1 2, на транзисторах з інжекційною інтегрованою логікою — 3, резерв — 4–9,
- для другої групи структура на польових транзисторах *n-* або *p-* типу — відповідно 5, 6, із симетричною комплементарною структурою — 7, структури із зарядовим зв'язком — 8, резерв — 1–4, 9,
- для третьої групи (пбридні мікросхеми) товстоплівкові — 1, тонкоплівкові — 2, комбіновані — 3, резерв — 4–9,
- для четвертої групи товстоплівкові — 4, тонкоплівкові — 2, комбіновані — 3, резерв — 4–9

За функціональним призначенням мікросхеми поділяються на групи, яким надаються такі позначення

- генератори — Г,
- комутатори та ключі — К,
- логічні елементи — Л,
- багатофункціональні схеми — Х,
- набори елементів — Н,
- перетворювачі сигналів — П,
- схеми джерел вторинного електроживлення — Е,
- схеми затримки — Б,
- схеми порівняння — С,
- тригери — Т,
- підсилювачі — У,
- формувачі — А,
- схеми запам'ятовуючих пристроїв — Р,
- схеми цифрових пристроїв — И,
- схеми обчислювальних засобів — В

У кожній функціональній групі розрізняють види, наприклад

- логічних елементів ЛИ — елемент І, ЛН — елемент НЕ, ЛЛ — елемент ЧИ, ЛА — елемент НЕ І, ЛЕ — елемент НЕ ЧИ, ЛР — елемент НЕ І ЧИ, ЛД — розширювачі, ЛП — інші,
- тригерів ТВ — універсальні (тип *JK*), ТР — з роздільним записом (тип *RS*), ТМ — із затримкою (тип *D*), ТК — комбіновані, ТП — інші,
- схем обчислювальних засобів ВЕ — мікро-ЕОМ, ВМ — мікропроцесори, ВС — мікропроцесорні секції, ВУ — схеми мікропрограмного керування,

ВБ — схеми синхронізації, ВВ — схеми інтерфейсу, ВН — часозадавальні схеми, ВП — інші

Для характеристики матеріалу і типу корпусу за ГОСТ 174–67 перед цифровим позначенням серії додаються такі букви,

- Р — для пластмасового корпусу типу 2,
- М — для керамічного, металокерамічного корпусу типу 2,
- Е — для металополімерного корпусу типу 2,
- С — для склокерамічного корпусу типу 2 та ін

Для деяких мікросхем буквени позначення типу корпусу не застосовують

Присвоєння позначень мікросхемам здійснює у централізованому порядку головна організація із стандартизації виробів електронної техніки

Позначення мікросхем має складатися з таких елементів

- перший і другий елементи — дві цифри, які характеризують відповідно групу і підгрупу мікросхеми за конструктивно-технологічним виконанням,
- третій елемент — дві цифри, що позначають порядковий номер розробки серії мікросхеми,
- четвертий елемент — дві букви, що характеризують відповідно групу та вид мікросхеми,
- п'ятий елемент — дві цифри, що позначають порядковий номер розробки мікросхеми

Три перших елементи визначають серію мікросхем. У разі необхідності після позначення порядкового номера розробки мікросхеми за функціональним призначенням додатково проставляються букви від А до Я, що характеризують відмінність мікросхем одного типу за електричними параметрами. Така буква під час маркування може бути замінена кольоровою точкою. Буква чи колір маркувальної точки зазначаються у технічних умовах до мікросхем конкретних типів.

Приклади умовного позначення мікросхем

- 5704ВГ03 — напівпровідниковий програмований контролер керування динамічною пам'яттю з симетричною комплементарною структурою серії 5704, номер розробки серії — 04, номер розробки мікросхеми у даній серії за функціональним призначенням — 03 (рис. 1.31),

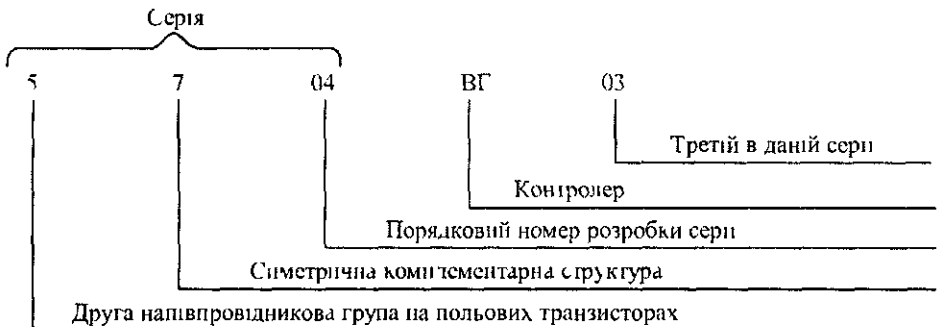


Рис. 1.31. Структура умовного позначення програмованого контролера пам'яті із симетричною комплементарною структурою 5704ВГ03

- 1101УД06 — напівпровідникова мікросхема серії 1101, порядковий номер розробки серії — 01, структура на біполярних транзисторах з ізоляцією $p-n$ переходом, операційний підсилювач постійного струму, порядковий номер розробки мікросхеми в даній серії за функціональним призначенням — 06.

Перед умовним позначенням мікросхем наводиться скорочена назва держави-розробника — У (Україна). Для мікросхем із кроком 1,27 чи 2,54 мм між выводами корпусу, які поставляються на експорт, на початку умовного позначення після літери У проставляється літера Е, наприклад:

- УЕ1217УД06 — мікросхеми серії 1217, вироблені в Україні в експортному виконанні (крок выводів 1,27 або 2,54 мм) в пластмасовому корпусі типу 2.

У раніше прийнятих позначеннях ІМС широкого застосування на початку ставили букву К.

Примітка

Умовні позначення ІМС, які випускались до 1991 року в СРСР, встановлювались ОСТ 11073.915–80. У відповідності з галузевим стандартом, умовне позначення мікросхем складалось із чотирьох цифро-буквених елементів.

Перший елемент — цифра, яка позначає конструктивно-технологічну групу: напівпровідникових — 1, 5, 6, 7; гібридних — 2, 4 8; інших (плівкових, керамічних) — 3. Другий елемент, який складається з двох або трьох цифр, позначає порядковий номер серії. Третій двобуквений елемент визначає функціональне призначення ІМС: перша буква — підгрупу; друга — вид в підгрупі, що співпадає з ДСТУ 3212-95. Четвертий елемент — це порядковий номер розробки ІМС даного типу в серії. Перший та другий елементи разом позначають серію ІМС, перед якою можуть проставлятися букви, що характеризують тип корпусу.

В подальшому умовні позначення ІМС подаються за галузевим стандартом. Для ІМС комерційного призначення умовне позначення починається з букви К, а в експортному варіанті — з букв ЭК. Приклад умовного позначення D-тригера ТТЛШ серії КР1333ТМ2 наведено на рис. 1.32.

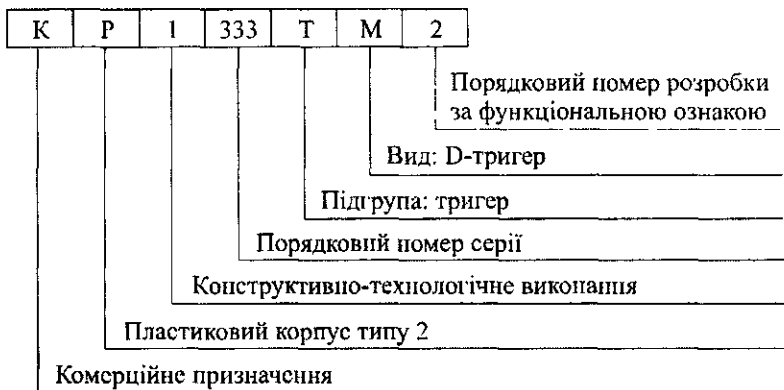


Рис. 1.32. Приклад умовного позначення ІМС за ОСТ 11073.915–80

Корпуси мікросхем складаються з трьох головних частин: кристала, корпусу для захисту кристала від кліматичних впливів і для зручності монтажу, а також провідни-

ків для електричного зв'язку між кристалом і виводами корпусу. Є чотири головних конструкційно-технологічних варіанти корпусів (рис. 1.33): пластмасовий, металокерамічний, металоскляний і склокерамічний.

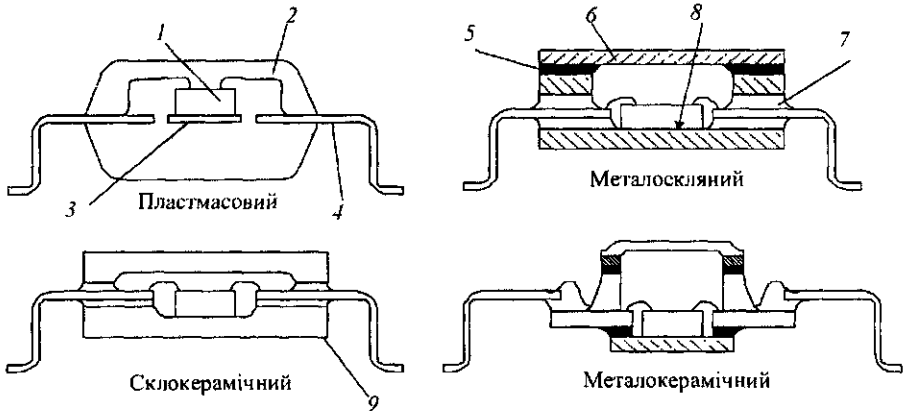


Рис. 1.33. Конструкції корпусів ІМС: 1 — кристал ІМС; 2 — проводові провідники; 3 — кристалотримач; 4 — виводи; 5 — низько температурний припій; 6 — кришка корпусу; 7 — скло (склоприпій); 8 — монтажна площадка; 9 — основа корпусу

Контрольні запитання

1. Які методи і технічні засоби вивчає дисципліна "Комп'ютерна схемотехніка"?
2. Що таке інформатика?
3. Що таке сигнал?
4. Охарактеризуйте основні системи числення, які використовуються в комп'ютерах.
5. Дайте визначення логічних функцій інверсії, диз'юнкції і кон'юнкції.
6. Назвіть основні технічні характеристики цифрових мікросхем.
7. Назвіть буквено-цифрові елементи, які визначають умовні позначення мікросхеми.
8. Поясніть кількісні міри інформації — біт, байт, кілобайт.
9. Запишіть число $A = -0101$ в прямому, оберненому і доповняльному кодах та їх модифікаціях.
10. Поясніть, для чого використовують мінімізацію булевих виразів.
11. Назвіть відміни між картами Карно і діаграмами Вейча.
12. Дайте визначення потенціального інформаційного сигналу.
13. Поясніть відміну між позитивним і негативним кодуванням логічних змінних потенціальними сигналами.
14. Що таке імпульсний інформаційний сигнал?
15. Поясніть способи кодування логічних змінних імпульсними сигналами.
16. Назвіть функціонально повні системи логічних елементів.
17. Охарактеризуйте класифікаційні ознаки елементів комп'ютерної схемотехніки.
18. Що таке серія ІМС?

Розділ 2

Елементи комп'ютерної схемотехніки**2.1. Діодні, транзисторні, інтегрально-інжекційні та діодно-транзисторні логічні елементи**

У комп'ютерній схемотехніці в основному використовують потенціальну систему елементів. Вона має такі особливості:

- на входах і виходах логічних елементів діють тільки потенціальні сигнали,
- з виходу одного елемента на вхід іншого передаються як перехідні, так і встановлені значення сигналів,
- на шляху потенціального сигналу не дозволяється вмикати конденсатори, обмотки трансформаторів, оскільки тривалість потенціального сигналу взагалі не обмежена (такий вид електричного зв'язку називається гальванічним або безпосереднім),
- відсутність конденсаторів і трансформаторів у колах зв'язку сприяє спрощенню технології виготовлення мікросхем,
- реалізується обмежений набір булевих функцій НЕ, ЧИ, І, НЕ ЧИ, НЕ І, НЕ І-ЧИ, "виключальне ЧИ", що полегшує застосування автоматизованих методів проектування.

Потенціальні елементи розрізняють за схемотехнічною ознакою — способом з'єднання транзисторів, діодів і резисторів між собою в межах однієї схеми типового базового елемента. Прийнято вважати, що сукупність елементів із загальною ознакою побудови утворюють вид схемної логіки або просто логіку.

Розрізняють такі види логіки потенціальних елементів:

- діодну (ДЛ) і діодно-транзисторну логіку (ДТЛ),
- транзисторну логіку (ТЛ), у якій виділяють схеми з безпосередніми зв'язками (ТЛБЗ), з резисторними зв'язками (ТЛРЗ) і резисторно-конденсаторними зв'язками (ТЛРКЗ),
- інтегральну інжекційну логіку (ІІЛ або І²Л),
- транзисторно-транзисторну логіку (ТТЛ) та її модифікації з діодами Шотки (ТТЛШ),
- емітерно-зв'язану логіку (ЕЗЛ),
- МОН-транзисторну логіку (р-МОН, n-МОН, КМОН),
- логіку на основі арсеніду галію (AsGa).

2.1.1. Діодні логічні елементи

Діодні логічні елементи є історично першими і найпростішими схемами, які реалізують булеві функції ЧИ, І, І-ЧИ, ЧИ-І. Діодні елементи не посилюють вхідних сигналів і не можуть виконувати операцію НЕ.

При розгляді роботи логічних елементів припускають (якщо не оговорено окре-

мо) позитивне кодування високий рівень напруги U_H відображає лог 1, а низький рівень U_H — лог 0

Елемент ЧИ

Логіка роботи логічного елемента ЧИ на два входи X_1 і X_2 подана в табл 2.1, на основі якої одержують вираз для вихідної булевої функції елемента $F = X_1 \vee X_2$

Використовуючи принцип суперпозиції, функцію F можна записати для довільного числа вхідних змінних

$$F = X_1 \vee X_2 \vee \dots \vee X_n$$

Схема двовходового елемента ЧИ, його умовне графічне зображення і часові діаграми роботи показані на рис 2.1

Таблиця 2.1

X_1	X_2	F
0	0	0
0	1	1
1	0	1
1	1	1

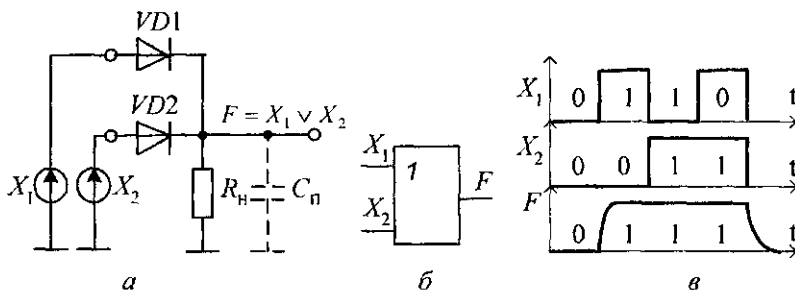


Рис. 2.1. Діодний елемент ЧИ:

а — схема; б — умовне графічне зображення; в — часові діаграми роботи

Високий рівень напруги U_{OH} на виході діодного елемента ЧИ встановлюється при подачі на один або обидва входи високих рівнів напруги U_{IH} , при яких відкриваються відповідні кремнієві діоди $VD1$ чи $VD2$, або обидва разом. В резистор навантаження R_H втікає вихідний струм I_{OH} , визначений за формулою

$$I_{OH} = U_{OH} / R_H$$

Значення вихідного високого рівня U_{OH} залежить від вхідних напруг

$$U_{OH} = U_{IH} - U^*$$

де $U^* = 0,8$ В — пряме падіння напруги на кремнієвому діоді

Для $U_{IH} = 5$ В, $R_H = 1$ кОм одержимо

$$U_{OH} = U_{IH} - U^* = 5 - 0,8 = 4,2$$
 В,

$$I_{OH} = U_{OH} / R_H = 4,2 / 103 = 4,2$$
 мА.

Під час подачі одночасно на обидва входи низьких рівнів напруги $U_{IL} \leq 0,4$ В, діоди закриті, струм у колі навантаження не протікає і вихідна напруга U_{OL} майже дорівнює нулю

На виході елемента звичайно є паразитна ємність $C_{П} = 25.. 100$ пФ, внаслідок чого тривалість фронту t_{IH} дуже мала (ємність швидко заряджається від джерела вхідних сигналів через малий прямий опір відкритого діода), а тривалість спаду t_{HI} велика (діоди закриті і ємність розряджається через резистор R_H). Тому для діодних схем ЧИ виконується нерівність $t_{IH} \ll t_{HI}$ (рис 2.1, в)

Елемент І

Логіка роботи логічного елемента І на два входи X_1 і X_2 подана в табл. 2.2, на основі якої одержують вираз для вихідної булевої функції елемента $F = X_1 X_2$.

Таблиця 2.2

X_1	X_2	F
0	0	0
0	1	0
1	0	0
1	1	1

Використовуючи принцип суперпозиції, функцію F можна записати для довільного числа вхідних змінних:

$$F = X_1 X_2 \dots X_n.$$

Схема двовходового елемента І, його умовне графічне зображення і часові діаграми роботи, показані на рис. 2.2.

Високий рівень напруги U_{OH} на виході діодного елемента І встановлюється тільки при одночасній подачі на обидва входи високих рівнів напруги U_{IH} , при яких закриваються кремнієві діоди $VD1$ і $VD2$. При цьому від джерела живлення U_{CC} через резистори R і R_H протікає струм навантаження

$$I_{OH} = U_{CC} / (R + R_H),$$

що визначає значення високого рівня вихідної напруги

$$U_{OH} = I_{OH} R_H = U_{CC} R_H / (R + R_H).$$

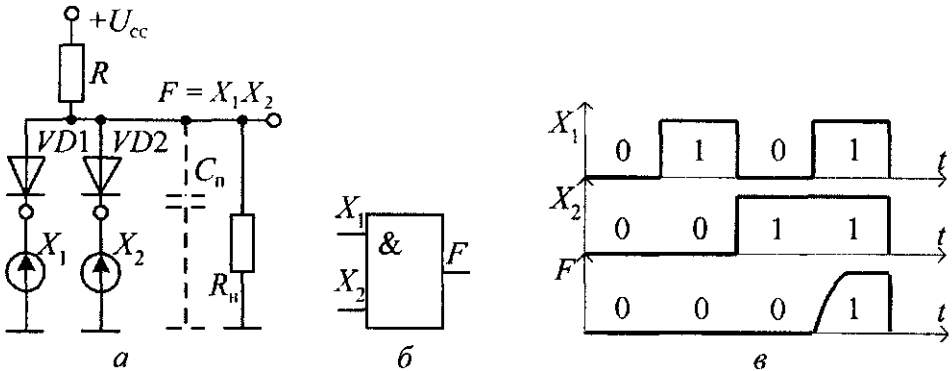


Рис. 2.2. Діодний елемент І:

а — схема; б — умовне графічне зображення; в — часові діаграми роботи

Як правило, використовують значення $R = 1 \dots 2$ кОм і $R_H > R$. Тривалість фронту вихідного сигналу t_{LH} визначається часом заряду паразитної ємності C_{Π} через великий опір резистора R . Якщо на один із входів, наприклад X_1 , поданий низький рівень напруги U_{IL} , то діод $VD1$ відкривається. При цьому від джерела живлення U_{CC} у колі резистор R , відкритий діод $VD1$ і джерело вхідного сигналу X_1 протікає струм, значення якого визначають із виразу

$$I_{IL} = [U_{CC} - (U^* + U_{IL})] / R,$$

і на виході встановлюється низький рівень напруги

$$U_{OL} = U_{CC} - I_{IL} R = U_{IL} + U^*,$$

де $U^* = 0,8$ В — пряме падіння напруги на відкритому діоді $VD1$. Джерела вхідних сигналів будують так, щоб вони пропускали струм I_{IL} , який в них втікає.

Тривалість спаду вихідного сигналу визначається часом розряду паразитної ємності C_{Π} через малий прямий опір відкритого діода. Тому в діодних схемах І тривалість фронту вихідного сигналу значно більша за тривалість спаду: $t_{LH} \gg t_{HL}$.

2.1.2. Елемент НЕ

Логіка роботи логічного елемента НЕ (інвертора) подана в табл. 2.3, на основі якої одержують вираз для вихідної булевої функції $F = \overline{X}$.

Схема елемента НЕ, його умовне графічне зображення і часові діаграми роботи показані на рис. 2.3. Схема елемента НЕ включає: $VТ1$ — кремнієвий транзистор $n-p-n$ типу; резистори в колі колекторного навантаження R_K і бази R_B ; U_{CC} — джерело живлення. Таку схему часто називають транзисторним ключем.

Таблиця 2.3

X	F
0	1
1	0

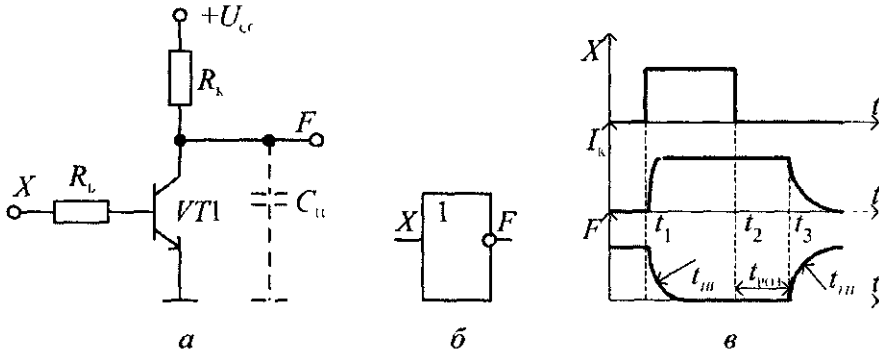


Рис. 2.3. Елемент НЕ:

а — схема; б — умовне графічне позначення; в — часові діаграми роботи

Транзистор $VТ1$ може знаходитися в трьох основних режимах: відсікання колекторного струму (закритий стан), насичення (відкритий стан) і активної роботи (посилення). У режимі відсікання колекторний і базовий переходи закриті (на вхід поданий низький рівень напруги U_{IL}), у колі колектора протікає дуже малий обернений струм колекторного переходу $I_{KO} \leq 1 \text{ мкА}$ і на колекторі транзистора встановлюється високий рівень напруги, визначений за виразом

$$U_{OH} = U_{CC} - I_{KO} R_K \approx U_{CC}.$$

У режимі насичення (на вхід поданий високий рівень напруги U_{OH}) на кремнієвому переході база-емітер пряме падіння напруги дорівнює $U_{BH} = U^* = 0,8 \text{ В}$, через колектор протікає максимально можливий струм, який називається струмом насичення колектора I_{KH} . Значення цього струму знаходять із співвідношення

$$I_{KH} = (U_{CC} - U_{KH})/R_K,$$

де $U_{KH} = U_{OL} \leq 0,4 \text{ В}$ — напруга на колекторі насиченого транзистора. Наприклад, для типових значень $U_{CC} = 5 \text{ В}$, $R_K = 1 \text{ кОм}$ одержимо для струму $I_{KH} \approx 5 \text{ мА}$. Для одержання цього струму потрібний мінімальний базовий струм насичення I_{BH} , значення якого обчислюють за співвідношенням

$$I_{BH} = I_{KH}/\beta,$$

де β — коефіцієнт передачі базового струму в транзисторі, увімкненому за схемою із спільним емітером. Для забезпечення надійного насичення транзистора значення базового струму насичення розраховують за виразом

$$I_{B1} = S I_{BH},$$

де $S = 3 \dots 10$ — ступінь насичення. Наприклад, якщо $\beta = 50$, то $I_{BH} = 0,1 \text{ мА}$ і прямий струм бази, що вмикає транзистор, $I_{B1} = 0,5 \text{ мА}$ для $S = 5$.

Значення опору резистора R_B одержують з умови забезпечення необхідного значення вмикаючого струму I_{B1}

$$R_B = (U_{IH} - U^*) / I_{B1}$$

Для $U_{IH} = 5$ В $U^* = 0,8$ В одержимо $R_B = 8,4$ кОм

У режимі насичення в базі транзистора накопичується надлишковий заряд значення якого пропорційне ступеню насичення. При подачі низького рівня входного сигналу транзистор закривається. Проте колекторний струм залишається постійним в інтервалі часу $t_2 - t_3$, який називається часом розсмоктування $t_{роз}$ надлишкового заряду в базі (рис. 2.3 в). Після закінчення розсмоктування колекторний струм спадає і формується фронт вихідного сигналу t_{1H} . Таким чином наявність насичення викликає затримку вимикання інвертора, що є недоліком даної схеми.

Зменшення тривалості перехідних процесів забезпечується схемою елемента НЕ (рис. 2.4 а) у якій замість резистора R_B включені два послідовно увімкнені діоди $VD1$ і $VD2$.

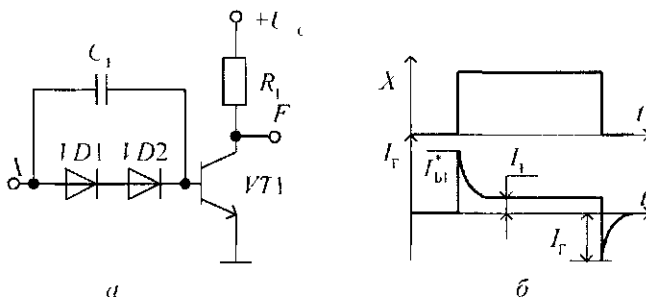


Рис. 2.4 Елемент НЕ з діодами в колі бази
а — схема, б — часові діаграми базових струмів

Діоди $VD1$ і $VD2$ називаються зміщувальними, оскільки вони зміщують граничний рівень входної відчиняючої напруги у більший бік на значення $2U^*$. Ємність C_B називається прискорюючою (форсуючою), оскільки вона при вмиканні швидко заряджається збільшуючи при цьому прямиий струм бази $I_{B1}^* > I_{BH}$, а при вимиканні швидко розряджається створюючи зворотний вмикаючий струм з амплітудою I_{B2} . Цей струм прискорює розсмоктування надлишкового заряду в базі транзистора.

2.1.3. Елементи ДТЛ

У цих елементах операції диз'юнкції і кон'юнкції реалізуються за допомогою діодних схем, а операцію заперечення виконує інвертор на основі транзисторного ключа. Логіка роботи двовходового елемента НЕ ЧИ подана в табл. 2.4 на основі якої одержують вираз для булевої функції

$$F = \overline{X_1 \vee X_2}$$

Використовуючи принцип суперпозиції функцію F можна записати для довільного числа змінних

$$F = \overline{X_1 \vee X_2 \vee \dots \vee X_n}$$

Схема ДТЛ-елемента, що реалізує булеву операцію НЕ ЧИ для двох змінних і його умовне графічне зображення показані на рис. 2.5

Таблиця 2.4

X_1	X_2	F
0	0	1
0	1	0
1	0	0
1	1	0

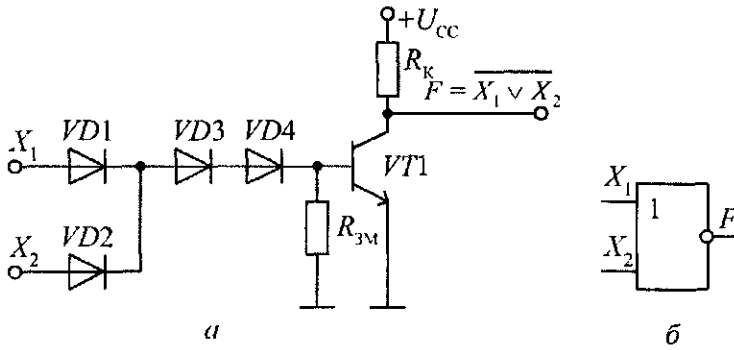


Рис. 2.5. Елемент НЕ ЧИ: а — схема; б — умовне графічне позначення

Схему ДТЛ-елемента НЕ ЧИ будують підключенням виходу діодного елемента ЧИ до входу інвертора. Зміцувальні діоди $VD3$ і $VD4$ відіграють роль, як і у схемі, зображеній на рис. 2.4, а. Через резистор зміщення R_{3M} протікає струм I_{KO} . Транзистор $VT1$ відкривається і формує низький рівень вихідної напруги U_{OL} тільки у тому випадку, коли на одному чи обох входах є високий рівень напруги U_{IH} . При цьому вмикаючий струм $I_{Б1}$ проходить від джерел вхідних сигналів.

Часові діаграми роботи (ідеалізовані) елемента НЕ ЧИ показані на рис. 2.6. Тривалість перехідних процесів практично визначається швидкодією транзистора.

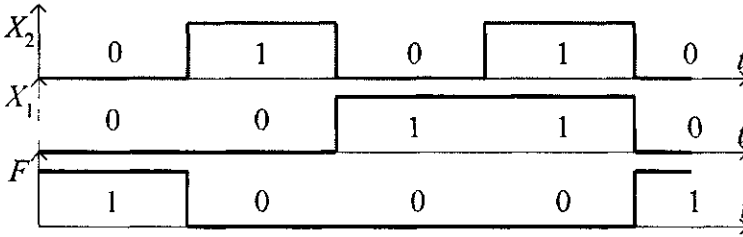


Рис. 2.6. Часові діаграми роботи елемента НЕ ЧИ

Логіка роботи двовходового елемента НЕ І подана в табл.2.5, на основі якої одержують вираз для булевої функції $F = \overline{X_1 X_2}$. На основі принципу суперпозиції функцію F можна записати для довільного числа змінних:

$$F = \overline{X_1 X_2 \dots X_n}$$

Схема ДТЛ-елемента, що реалізує булеву операцію НЕ І для двох змінних X_1 і X_2 , і його умовне графічне зображення показані на рис. 2.7.

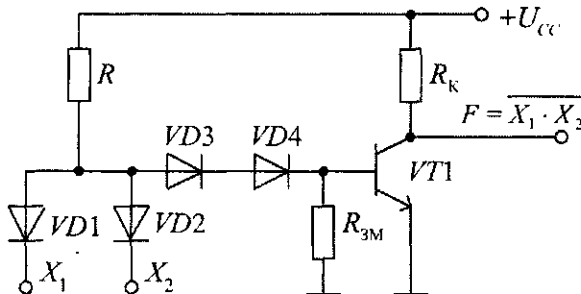


Рис. 2.7. Схема ДТЛ-елемента НЕ І

Таблиця 2.5

X_1	X_2	F
0	0	1
0	1	1
1	0	1
1	1	0

Схему ДТЛ-елемента НЕ 1 будують підключенням виходу діодного елемента 1 до входу інвертора. Зміщувальні діоди $VD3$ і $VD4$ відіграють роль, як і у схемі, зображеній на рис. 2.4.

Транзистор $VT1$ відкривається і формує низький рівень вихідної напруги U_{OH} тільки при збігу високих рівнів вхідних сигналів U_{IH} на обох входах одночасно. При цьому вмикаючий струм I_{B1} протікає від джерела струму U_{CC} по колу: резистор R , діоди $VD3$, $VD4$ і база транзистора $VT1$. При подачі хоч би на один із входів (або на обидва) сигналу низького рівня весь струм від джерела живлення U_{CC} замикається по колу: резистор R , діод $VD1$ або $VD2$ (або обидва) на відповідне джерело вхідних сигналів. В результаті цього транзистор $VT1$ закривається і на виході встановлюється високий рівень напруги U_{OH} . Тривалість перемикання елемента практично визначається швидкодією транзистора. Часові діаграми роботи елемента НЕ 1 показані на рис. 2.8.

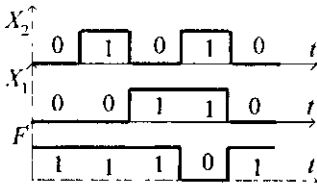


Рис. 2.8. Часові діаграми роботи елемента НЕ 1

2.1.4. Транзисторна логіка

Елементи ТЛ будують паралельним підключенням колекторів транзисторів до спільного резистора колекторного навантаження R_K . Вихідна напруга знімається з об'єднаних колекторів. Якщо на один із входів надходить високий рівень напруги U_{IH} , то відповідний транзистор відкривається і на виході встановлюється низький рівень сигналу U_{OH} . На виході формується високий рівень напруги тільки у випадку подачі на всі входи низьких рівнів U_{IL} .

Логіка роботи елементів ТЛ задана в табл. 2.4. В елементах ТЛБЗ вхідні сигнали подають безпосередньо на бази транзисторів $VT1$ і $VT2$ (рис. 2.9, а). У схемі ТЛРЗ у колі баз транзисторів увімкнені резистори R_B (рис. 2.9, б).

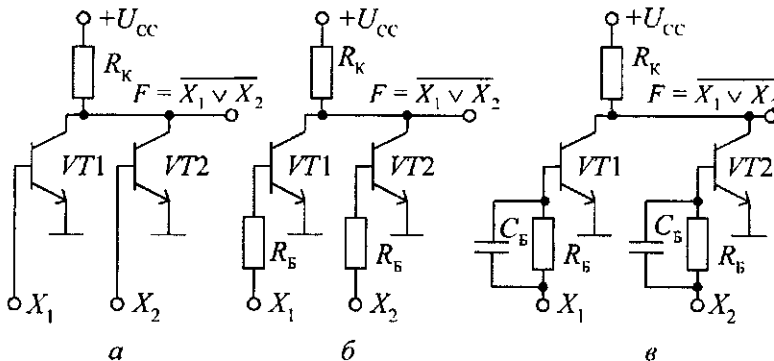


Рис. 2.9. Елементи ТЛ: а — ТЛБЗ; б — ТЛРЗ; в — ТЛРКЗ

У схемі ТЛРКЗ для прискорення перехідних процесів резистори R_B шунтовані конденсаторами C_B . Їхня роль описана при розгляді схеми інвертора (див. рис. 2.4). Часові діаграми роботи елементів ТЛ збігаються з процесами, зображеними на рис. 2.6.

Інтегральна інжекційна логіка

Різновидом транзисторних схем є елементи інтегральної інжекційної логіки (ІІЛ або І²Л). Схемотехніку І²Л використовують для побудови мікропроцесорних і запам'ятовуючих ВІС (серії К582, К583, К584 та ін.).

Схема логічного елемента І²Л показана на рис. 2.10.

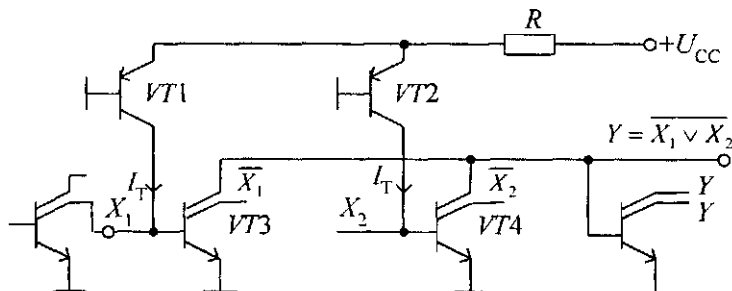


Рис. 2.10. Схема елемента І²Л

Схема вміщує інжекційні *p-n-p* транзистори *VT1*, *VT2*, увімкнені за схемою зі спільною базою, і входні багатоколекторні *n-p-n* транзистори *VT3*, *VT4*, увімкнені за схемою зі спільним емітером. Емітери транзисторів *VT1*, *VT2* називаються інжекторами, а протікаючий через них дірковий струм — інжекційним. Кожний з транзисторів *VT1*, *VT2* утворює разом із джерелом живлення і зовнішнім резистором *R* джерело струму, яке живить індивідуальним струмом I_T входи транзисторів *VT3*, *VT4*.

Особливостями елементів І²Л є:

- “безрезисторність”, характерна для МОН-структур, яка вперше була реалізована в схемотехніці І²Л;
- з'єднання областей бази і колектора інжекційних *p-n-p* транзисторів відповідно з областями емітера і колектора входних *n-p-n* транзисторів, а також мале число схемних компонентів і з'єднань між ними (число операцій маскування і дифузії в два рази менше порівняно з елементами ТТЛ);
- низький рівень напруги $U_L=0,01$ В знімається з колектора насиченого транзистора, а високий рівень напруги $U_H=0,8$ В — з колектора закритого транзистора, причому цей рівень обмежується напругою бази насиченого транзистора навантаження; використовується режим мікрострумів, в якому струми колектора змінюються від десятків до сотень мікроампер; працездатність елементів зберігається при зміні значення струму в них на декілька порядків;
- на колекторах входного транзистора реалізується інверсія змінної, а на сполучених колекторах транзисторів *VT3*, *VT4* виконуються операції НЕ ЧИ.

Вхідні транзистори керуються перемиканням струму на їхніх входах. Якщо до входу X_1 підключений колектор лівого насиченого транзистора, то струм I_T замикається на ньому і не надходить у базу транзистора *VT3*, який закривається і створює на своїх колекторах режим розімкнутих контактів. Якщо до входу X_1 підключений колектор лівого закритого транзистора, то струм I_T входить у базу *VT3*, насичує його і забезпечує на колекторах режим замкнутих контактів.

Затримка поширення сигналу в елементі І²Л при струмі 100 мкА дорівнює приблизно 5–10 нс, потужність споживання — до 20 мкВт, робота перемикання дорів-

нює 0,1 пДж (наприклад, у ТТЛШ серії КР1533 робота перемикачів складає 6 пДж). Зазначені властивості елементів І²Л і ВІС на їхній основі надають їм технологічності, компактності, вони мають невисоку вартість при високій швидкодії.

2.2. Транзисторно-транзисторні логічні елементи

2.2.1. Загальна характеристика транзисторно-транзисторних логічних елементів

Елементи ТТЛ з'явилися в результаті розвитку схем ДТЛ у напрямку скорочення числа компонентів, зменшення ємності переходів і врахування специфіки інтегральної технології. Спочатку головною особливістю елементів ТТЛ було використання на вході багатоемітного транзистора (БЕТ) для реалізації операції І. Кожен емітер БЕТ використовують як логічний вхід. Число емітерів визначає коефіцієнт на вході $N_I = 1 \dots 8$ і більше.

Промисловість випускає такі серії ТТЛ і ТТЛШ:

- стандартної середньої швидкодії (ТТЛ) — К133, К155, швидкодіючі — К130, К131, К599 і малопотужні — К134, К158 (1963 р.);
- стандартні швидкодіючі (ТТЛШ) — К530, К531 і малопотужні — К533, К555 (1970 р.);
- надшвидкодіючі (ТТЛШ) — КР1530 (тип АS), високої швидкодії — КР1531 (тип FАSТ) і малопотужні — КР1533 (тип АLС) (1980 р.). У цих серіях застосована ізопланарна технологія на основі іонної імплантації, прецизійної фотолітографії, які забезпечують малі площі і ємності переходів.

У мікросхемах серій К533, К555, КР1531 і КР1533 замість БЕТ на вході знову поставили діодні схеми збігу, проте назва схемотехніки ТТЛШ залишилась.

2.2.2. Транзисторно-транзисторний логічний елемент з простим інвертором

Схема найпростішого ТТЛ елемента містить БЕТ, колектор якого підключено до бази інвертуючого транзистора VT2 (рис. 2.11, а).

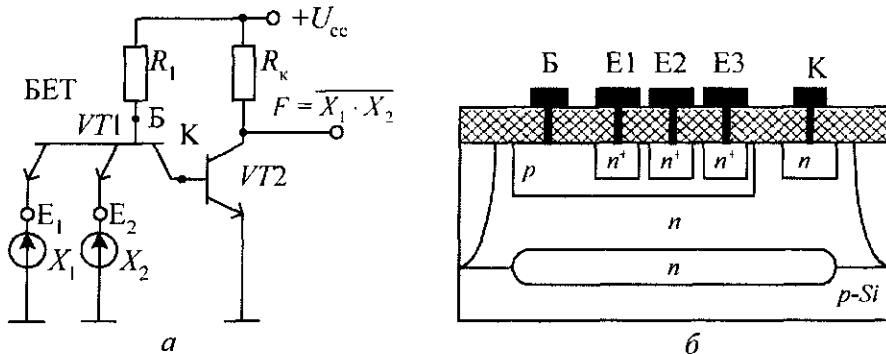


Рис. 2.11. Елемент ТТЛ з простим інвертором: а — схема; б — топологія

Багатоємітерний транзистор виконує операцію I_1 , транзистор $VT2$ — операцію НЕ. Тому елемент у цілому реалізує функцію $F = \overline{X_1 X_2}$. У першому наближенні БЕТ з топологією, показаною на рис. 2.11, б, розглядають як сукупність із n (за числом емітерів) окремих транзисторів із спільними базою і колектором. Для виключення взаємного впливу емітерні переходи розташовують один від одного на відстані 10...15 мкм, що перевищує дифузійну довжину пробігу носіїв у базовому прошарку.

У схемі ТТЛ елемента БЕТ працює в двох основних режимах — інверсному і насичення. Процеси, що відбуваються в інверсному режимі, відображають еквівалентною схемою (рис. 2.12, а).

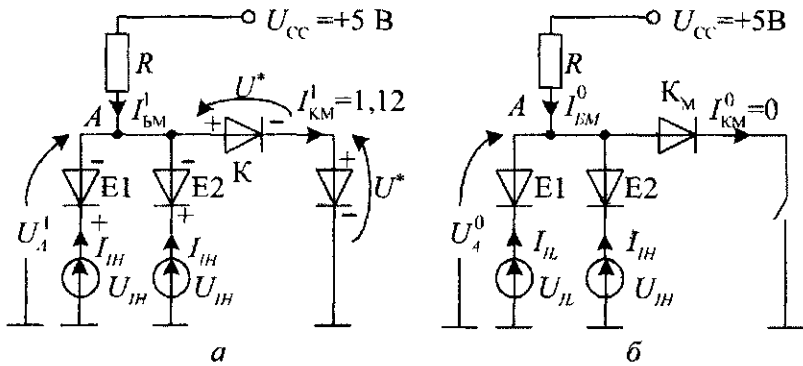


Рис. 2.12. Еквівалентні схеми елемента ТТЛ:
а — в інверсному режимі; б — в режимі насичення

Інверсний режим виникає після подачі на всі входи високих рівнів напруг U_{IH} . У результаті емітери $E1$ і $E2$ закриваються, колекторний перехід знаходиться під прямою напругою $U^* = 0,8$ В і він інжектує електрони у спільну базу.

З метою зменшення зворотних струмів I_{IH} , які втікають в емітер, технологічними методами спеціально зменшують значення інверсного коефіцієнта передачі струму β_I від бази до емітера до 0,01–0,02. У цьому випадку значення струму I_{IH} знаходять із співвідношення

$$I_{IH} = \beta_I I_{BM}^1,$$

де I_{BM}^1 — струм бази БЕТ.

На основі схеми рис. 2.12, а для струму I_{KM}^1 , що витікає з колектора БЕТ, справедливий вираз

$$I_{KM}^1 = I_{BM}^1 + 2I_{IH} = I_{BM}^1 + 2\beta_I I_{BM}^1 = I_{BM}^1 (1 + 2\beta_I).$$

Струм I_{BM}^1 знаходять з рівності

$$I_{BM}^1 = (U_{CC} - 2U^*)/R,$$

де $2U^* = 1,6$ В — сумарне падіння напруги на двох послідовно увімкнених кремнієвих переходах — колектора БЕТ і бази $VT2$. Для типових значень $U_{CC} = 5$ В, $R = 3$ кОм, одержимо:

$$I_{BM}^1 = (5 - 1,6) / 3 \cdot 10^3 = 1,12 \text{ мА};$$

$$I_{IH} = 0,02 \cdot 1,12 \cdot 10^{-3} = 25 \text{ мкА};$$

$$I_{KM}^1 = I_{BM}^1 (1 + 2 \cdot 0,02) \approx I_{BM}^1 = 1,12 \text{ мА}.$$

Таким чином, значення колекторного струму I_{KM}^1 достатнє для надійного наси-

чення транзистора $VT2$, при цьому на виході встановлюється низький рівень напруги $U_{OH} \leq 0,1 \text{ В}$

Режим насичення виникає при подачі на один з входів (або на обидва), наприклад X_1 , низького рівня входної напруги U_{II} (рис 2.12, б). При цьому емітер $E1$ відкривається, оскільки на ньому падіння напруги $U^* = 0,8 \text{ В}$, й інжектує електрони в базу БЕТ.

Колекторний перехід БЕТ також знаходиться під прямим зсувом. Проте він не може збирати ("колекціонувати") електрони, це рівнозначно появі струму, який втікає в колектор, чого не допускає база транзистора $VT2$. Вважають, що у колі колектора БЕТ увімкнено нескінченно великий опір навантаження, тому в області колектора БЕТ накопичується об'ємний заряд електронів, що характерно для режиму насичення. Транзистор $VT2$ — закритий і на його колекторі встановлюється високий рівень напруги $U_{OH} \approx U_{CC}$.

Значення витікаючого (від'ємного) з відкритого емітера $E1$ струму I_{II} знаходять з рівності

$$I_{II} = [U_{CC} - (U_{II} + U^*)] / R$$

Припустивши, що $U_{II} = 0,4 \text{ В}$, одержимо $I_{II} = 1,23 \text{ мА}$.

Таким чином, у розглянутій схемі елемента ТТЛ практично постійний струм бази $I_{БМ}^I$ комутується або в базу транзистора $VT2$ (інверсний режим), або у відкритий емітер БЕТ (режим насичення).

2.2.3. Транзисторно-транзисторний логічний елемент із складним інвертором

У мікросхемах ТТЛ використовують складні інвертори, які підвищують швидкість і навантажувальну здатність елементів. Схема базового (типового) елемента ТТЛ із складним інвертором містить три основних каскади (рис 2.13, а).

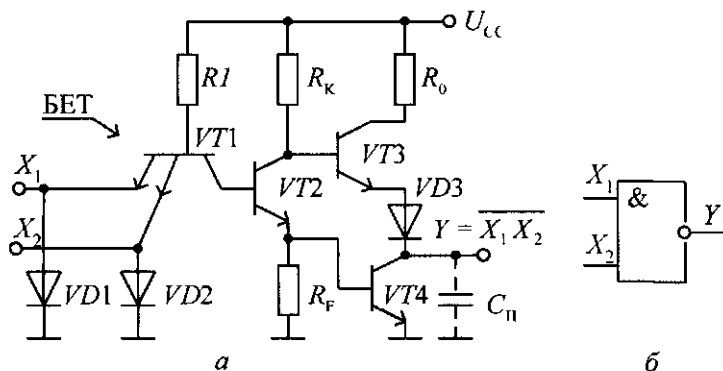


Рис. 2.13. Елемент ТТЛ із складним інвертором: а — схема; б — умовне позначення;

1. Входний каскад, який реалізує операцію І (транзистор $VT1$, резистор $R1$). До усіх входів БЕТ підключені демпфруючі (антидзвінні) діоди, які обмежують вплив імпульсів перешкод від'ємної полярності.
2. Фазоінверсний каскад (транзистор $VT2$, резистори R_k і R_E), що керує вихідними транзисторами за допомогою протифазних змін напруг на колекторі й емітері $VT2$.

3. Вихідний двоканальний підсилювач (транзистори $VT3$, $VT4$, зміщуючий діод $VD3$, резистор R_0) Складний інвертор утворюється спільною роботою фазоінверсного і вихідного каскадів

При збігу на входах елементів високих рівнів напруг БЕТ перемикається в інверсний режим і своїм колекторним струмом відкриває транзистор $VT2$. Частина емітерного струму транзистора $VT2$ втікає в базу транзистора $VT4$ і відкриває його. Після швидкого розрядження паразитної ємності C_{Π} через колектор насиченого транзистора $VT4$ на виході встановлюється низький рівень напруги. При цьому транзистор $VT3$ — закритий, оскільки напруга, що прикладається до послідовно включених переходів бази і діода $VD3$, недостатня для його відкриття.

При подачі на один із входів напруги низького рівня БЕТ перемикається в режим насичення, струм його колектора дорівнює нулю, внаслідок чого закриваються транзистори $VT2$ і $VT4$. При цьому відкритий транзистор $VT3$ працює в режимі емітерного повторювача: на його вхід надходить високий рівень напруги з колектора закритого транзистора $VT2$, а навантаженням служить опір закритого транзистора $VT4$. Емітерний повторювач передає на вихід високу напругу

$$U_{\text{ВН}} = U_{\text{CC}} - 2U^*,$$

де $2U^*$ — пряме падіння напруги на двох послідовно увімкнених переходах — бази транзистора $VT3$ і діода $VD3$. Повторювач створює у навантаженні струм, який у 50–100 разів перевищує його вхідне значення. Це також забезпечує швидке зарядження паразитної ємності C_{Π} . Часові діаграми роботи ТТЛ елемента збігаються з рис. 2.8.

У процесі перемикання є короткочасний інтервал, коли транзистор $VT4$ уже відкритий, а $VT3$ — ще не встиг закритися. При цьому виникає значний імпульс струму від джерела живлення на землю (наскрізний струм). Для зменшення амплітуди наскрізного струму в колекторі транзистора $VT3$ встановлено обмежувальний резистор $R_0 = 100 \text{--} 200 \text{ Ом}$.

Розглянута схема елемента ТТЛ із складним інвертором є типовою для ТТЛ серій К131, К133, К155 та ін.

Принцип роботи транзисторів Шотки

Елементи ТТЛШ порівняно з ТТЛ мають вищу швидкість і меншу споживану потужність, що досягається застосуванням діодів Шотки. Принцип роботи діода Шотки заснований на використанні потенційного бар'єра, що утворюється в приконтактній області між металом і напівпровідником. У діодах Шотки немає накопичення надлишкових зарядів, оскільки струм визначається переходом основних носіїв з напівпровідника в метал. Час перемикання діодів Шотки дуже малий (до 0,1 нс) і не залежить від температури. Порівняно з кремнієвими діодами пряме падіння напруги в діодах Шотки вдвічі менше (біля 0,3–0,4 В).

У режимі насичення на колекторі кремнієвого транзистора діє пряма напруга $U_{\text{КБ}} = 0,7 \text{ В}$, внаслідок чого колектор відкривається й інжектує електрони в базу. Це викликає затримку вимикання, обумовлену часом розсмоктування $t_{\text{роз}}$ (рис. 2.14, а). При наявності між базою і колектором діода Шотки (рис. 2.14, б) колектор при відкриванні транзистора не переходить у режим насичення, оскільки пряма напруга $U_{\text{КБ}} = 0,4 \text{ В}$. Транзистор з діодом Шотки між базою і колектором називають транзистором Шотки (рис. 2.14, в).

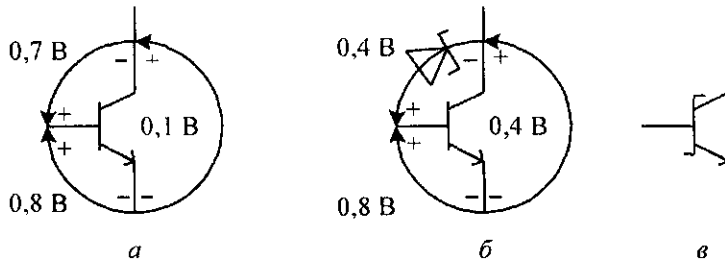


Рис 2.14 Розподіл напруг у відкритому транзисторі

а — звичайному, б — з діодом Шотки, в — умовне позначення транзистора Шотки

Таким чином, транзистор Шотки не переходить у режим насичення і тим самим виключається затримка вимикання. При цьому швидкодія збільшується приблизно в 3–5 разів.

Елементи ТТЛШ серії К530, К531

Схема типового елемента ТТЛШ серії К530 і К531 показана на рис. 2.15. Елемент реалізує операцію НЕ І для двох змінних X_1 і X_2 .

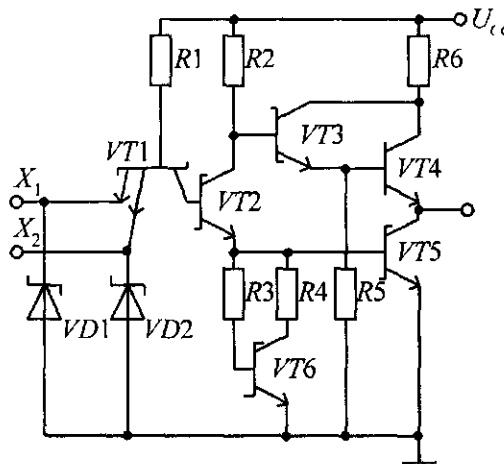


Рис 2.15 Схема елемента ТТЛШ серії К530, К531

Порівняно з елементом ТТЛ із складним інвертором (див. рис. 2.13) в схемі ТТЛШ серії К530 і К531 внесені такі зміни:

- використовують тільки діоди і транзистори Шотки (за винятком транзистора VT_4 , що не переходить у режим насичення),
- у фазоінверсний каскад додано корегуючий ланцюжок (транзистор VT_6 , резистори R_3 і R_4), який змищує поріг вмикання елемента в бік більших входних напруг, завдяки чому підвищується завадостійкість схеми,
- у вихідний каскад вмонтована схема Дарлінгтона на транзисторах VT_3 і VT_4 , яка подвоює значення коефіцієнта підсилення базового струму, що забезпечує великі струми у навантаженні та підвищує швидкодію елемента,
- у колекторі транзистора VT_4 включений резистор R_6 , який обмежує амплітуду “наскрізного” струму від джерела живлення на спільний провід у моменти

перемикання вихідних транзисторів, резистор $R5$ забезпечує проходження оберненого струму I_{KO} транзистора $VT4$

Схема працює аналогічно елементу ТТЛ із складним інвертором. При описі роботи елементів ТТЛШ інших серій збережені буквено-цифрові позначення резисторів, діодів і транзисторів, призначення яких описано стосовно до схеми, зображеної на рис. 2.15. Умовне графічне позначення розглянутого елемента подане на рис. 2.13, б.

Елементи ТТЛШ серій К533, К535

Схема типового елемента ТТЛШ серій К533 і К535 показана на рис. 2.16. Елемент реалізує операцію НЕ і для двох змінних X_1 і X_2 .

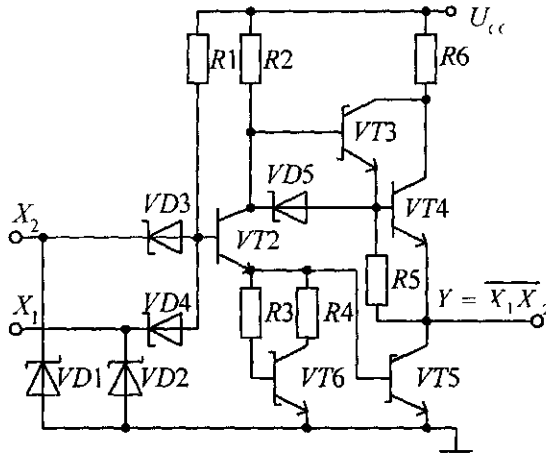


Рис. 2.16. Схема елемента ТТЛШ серій К533, К535

У схемі аналізованого елемента не застосовують БЕТ. Вхідний каскад утворюється схемою збігу на діодах $VD3$, $VD4$ і резисторі $R1$. Колектор транзистора $VT2$ додатково сполучений з базою транзистора $VT4$ через діод $VD5$ і резистор $R5$, це сприяє зменшенню часу перезарядження паразитних ємностей навантаження. При збігу високих рівнів вхідних напруг діоди $VD3$ і $VD4$ закриваються і струм від джерела живлення через резистор $R1$ відкриває транзистор $VT2$, при цьому вмикається також транзистор $VT5$ і на виході встановлюється низький рівень напруги.

Якщо на один із входів подати низький рівень напруги, то даний діод відкривається і через нього протікає струм, який створюється джерелом живлення через резистор $R1$, у цьому випадку транзистори $VT2$ і $VT5$ закриті і схема Дарлінгтона встановлює високий рівень вихідної напруги.

Елементи ТТЛШ серій КР1531 і КР1533

Елементи ТТЛШ нових серій КР1531 (умовна назва FAST) і КР1533 (умовна назва ALS) виготовляють за новітньою технологією "Ізопланар II", яка використовує юнну імплантацію (точне дозоване впровадження атомів домішки в області напівпровідника), прецизійну фотолітографію, що дозволяє у вісім разів зменшити площу, яку елементи займають у кристалі. Істотно зменшені споживана потужність і робота

перемикання, вхідні струми при низьких рівнях вхідних напруг ($I_{II} \leq 0,1$ мА). Схема типового елемента ТТЛШ серії КР1531 показана на рис. 2.17. Елемент реалізує операцію НЕ І для двох змінних X_1 і X_2 .

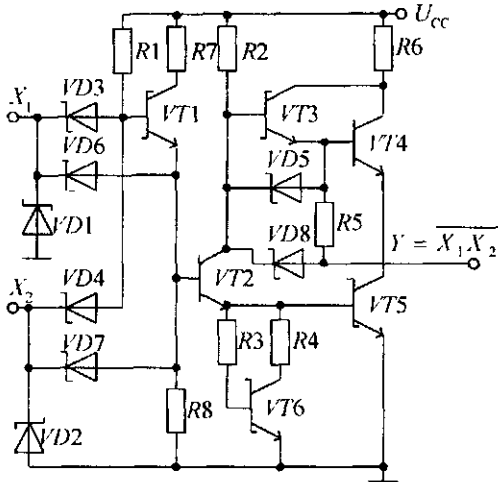


Рис. 2.17. Схема елемента ТТЛШ серії КР1531

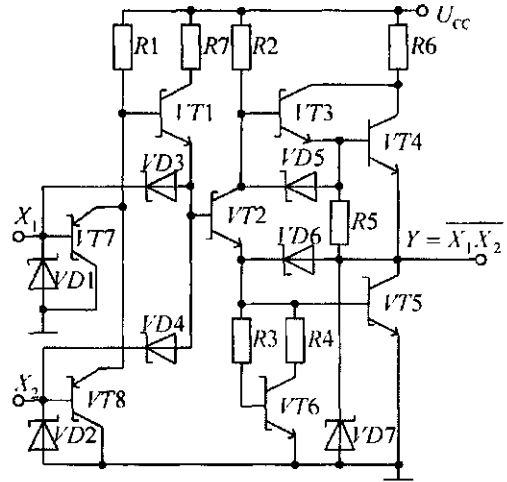


Рис. 2.18. Схема елемента ТТЛШ серії КР1533

У розглянутому елементі на виході діодної схеми збігу увімкнений додатковий підсилювач на транзисторі $VT1$. При збігу високих вхідних рівнів напруги діоди $VD3$ і $VD4$ закриваються, а транзистор $VT1$ відкривається. Струм його емітера створює на резисторі $R8$ падіння напруги, яке керує фазоінверсним каскадом. Додаткові діоди $VD6$, $VD7$ емнісними струмами своїх переходів прискорюють процес перемикання транзистора $VT1$.

У елементах серії КР1533 (рис. 2.18) як діоди схеми збігу використовують емітерні переходи $p-n-p$ транзисторів $VT7$ і $VT8$. Переходи закриті при збігу високих рівнів на входах; відкриваються транзистори $VT1$, $VT2$, $VT5$ і $VT6$. Якщо хоч би на один з входів подано низький рівень, то струм, що протікає через резистор $R1$, замикається на спільний провід по колу емітер — колектор відкритого $p-n-p$ транзистора. Внаслідок цього транзистори $VT1$, $VT2$, $VT5$ і $VT6$ закриваються, а $VT3$, $VT4$ — відкриваються. Застосування схеми збігу на переходах $p-n-p$ транзисторів дозволило порівняно з ТТЛШ серії КР1531 у 20 разів зменшити вхідні струми I_{II} , які витікають із входів.

2.2.4. Елементи ТТЛ і ТТЛШ з відкритим колекторним виходом і трьома станами

Для роботи на нестандартне навантаження, наприклад, лампу розжарювання (ЛР), обмотку реле випускають схеми елементів ТТЛ і ТТЛШ з відкритим колектором (рис. 2.19, а). При збігу високих рівнів на входах транзистор $VT3$ відкривається і ЛР світиться. Якщо хоча б на одному з входів буде низький рівень напруги, транзистор $VT3$ розмикається і ЛР гасне.

Якщо виходи декількох елементів із відкритим колектором підключаються до джерела живлення через загальний колекторний резистор, то така схема реалізує функцію НЕ-І-ЧИ (рис. 2.19, в).

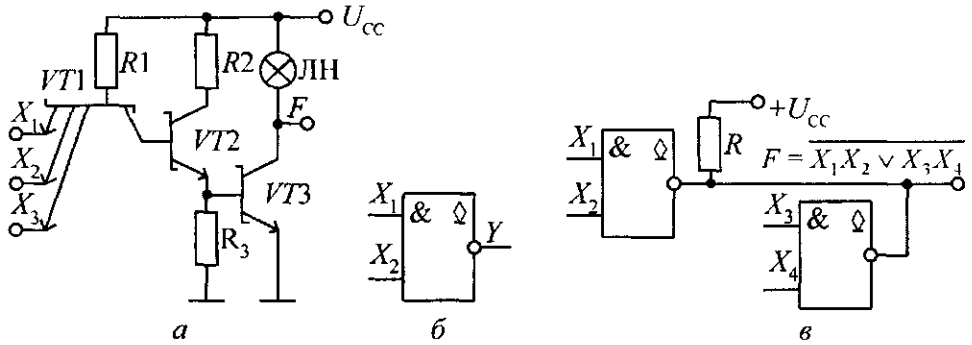


Рис. 2.19. Елемент ТТЛШ з відкритим колектором:

а — схема; б — умовне позначення; в — підключення до загального навантаження

У схемах елементів ТТЛ і ТТЛШ з'єднання виходів декількох елементів для уникнення протікання високого струму від виходу з високим рівнем U_{OH} на вихід з низьким рівнем U_{OL} неприпустиме, оскільки може призвести до виходу з ладу мікросхеми. При необхідності такого прямого з'єднання виходів (засіб "монтажне ЧИ") використовують елементи з трьома станами. Два стани виходів — це звичайна видача значень лог. 0 і лог. 1.

Третій стан характеризується нескінченно високим вихідним опором, коли елемент практично цілком відключається від навантаження, тобто не споживає і не видає струму. Це досягається рядом схемних рішень, у тому числі й найпростішим, показаним на рис. 2.20, а. У цьому випадку, коли діод $VD1$ підключено до емітера і колектора транзисторів $VT1$ і $VT2$, а на катод діода подається керуючий сигнал Z з високим рівнем напруги (лог. 1), то схема працює як елемент НЕ І.

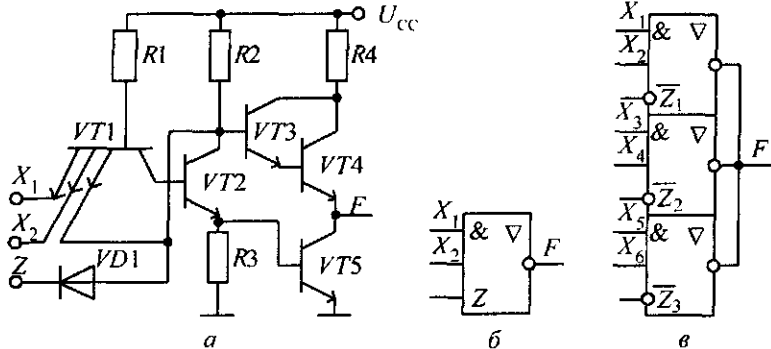


Рис. 2.20. Елемент ТТЛ з трьома станами:

а — схема; б — умовне позначення; в — підключення до загального навантаження

Якщо керуючий сигнал Z представлений низьким рівнем напруги (лог. 0), то БЕТ і колектор транзистора $VT2$ (а відповідно і база транзистора $VT3$) підключаються через відкритий діод $VD1$ до шини землі. У цьому випадку всі транзистори закриті й елемент переходить у третій стан (Z -стан). У серіях мікросхем часто керуючий сигнал є інверсним. При об'єднанні виходів елементів ТТЛ з трьома станами (рис. 2.20, в) керуючі сигнали \bar{Z}_1 , \bar{Z}_2 і \bar{Z}_3 мають подаватися тільки по чергово.

Для елементів з трьома станами введені такі часові параметри для задання затримок поширення:

- t_{LZ} і t_{HZ} — тривалість затримки при переході з низького і високого рівнів вихідної напруги в стан "вимкнено" відповідно,
- t_{ZL} і t_{ZH} — тривалість затримки при переході зі стану "вимкнено" у стан низького і високого рівнів вихідної напруги відповідно

2.2.5. Порівняльний аналіз параметрів елементів ТТЛ і ТТЛШ

Серії мікросхем ТТЛ і ТТЛШ вміщують широкий функціональний набір логічних елементів, тригерів, вузлів (реєстри, лічильники, суматори і т. ін.) Наявність готових вузлів у серіях дозволяє зменшувати число корпусів мікросхем і одержувати значний вигреш в об'ємі апаратури

Серії елементів ТТЛ і ТТЛШ мають функціональну і технічну повноту, працюють від джерела напруги плюс 5 В, сумісні за рівнями логічних сигналів, а частина — і за розведенням виводів корпусу мікросхеми. Рівні напруги лог 0 і лог 1 при позитивному кодуванні становлять для елементів ТТЛ — $U_L \leq 0,4$ В, $U_H \geq 2,4$ В, для елементів ТТЛШ — $U_L \leq 0,5$ В, $U_H \geq 2,7$ В

У елементах ТТЛ і ТТЛШ вдало поєднуються схемотехнічні, технологічні й конструктивні якості

- високий рівень схемно-технологічного відпрацювання, що забезпечує високий відсоток виходу придатних мікросхем і низьку вартість виготовлення,
- порівняно високі швидкодія і завадостійкість,
- висока навантажувальна здатність $N_0 = 10 \cdot 30$ і помірне споживана потужність,
- широкий функціональний набір елементів і зручність застосування (монтаж, компонування, охолодження)

Значення затримок і споживаної потужності елементів ТТЛ і ТТЛШ різноманітних серій наведені в табл. 2.6

Таблиця 2.6

Групи мікросхем	Номер серії	t_p , нс	P_{CC} , мВт	$A_{п}$, пДж
Стандартні ТТЛ	K133, K135	9	10	90
Швидкодіючі ТТЛ	K130, K131	6	22	132
Малопотужні ТТЛ	K134, K136	33	5	165
Стандартні ТТЛШ	K530, K531	3	19	57
Малопотужні ТТЛШ	K533, K531	9,5	2	19
FAST	KP1531	3	4	12
ALS	KP1533	4	1,2	4,8

До недоліків елементів ТТЛШ відносяться менша швидкодія порівняно з елементами ЕЗЛ, труднощі узгодження з низькоомним навантаженням, високий рівень утворюваних перешкод і зростання споживаної потужності з підвищенням частоти перемикання, значне імпульсне споживання струму під час перемикання, особливо при ємнісному навантаженні. Зусилля розробників спрямовані на розширення функціонального складу серій елементів ТТЛШ, функцій елементів і вузлів, а також на підвищення швидкодії, зменшення споживаної потужності та зниження вартості

Елементи ТТЛ і ТТЛШ у даний час складають основу елементного базису комп'ютерної схемотехніки. Вибір конкретної серії мікросхем для побудови цифрових пристроїв провадиться на основі таких рекомендацій:

- якщо до пристроїв не пред'являється особливих вимог, то використовують стандартні серії;
- якщо за умовами роботи потрібно забезпечити мале енергопостачання і допустимо знижену швидкодюю, то використовують малопотужні (економічні) серії;
- пристрої підвищеної продуктивності проектують на швидкодіючих серіях елементів.

2.3. Логічні елементи емітерно-зв'язної логіки

Схемотехніка елементів ЕЗЛ заснована на використанні диференціального підсилювача в режимі перемикання струму. Елементи ЕЗЛ з'явилися в 1967 р. і в даний час є надшвидкодіючими серед напівпровідникових елементів на основі кремнію. Затримки поширення сигналів в елементах ЕЗЛ зменшилися до субнаносекундного діапазону (приблизно 1 нс).

Надшвидкодія елементів ЕЗЛ досягається за рахунок використання ненасиченого режиму роботи транзисторів, вихідних емітерних повторювачів, малих амплітуд логічних сигналів (біля 0,8 В). У логічних елементах ЕЗЛ є парафазний вихід, що дозволяє одночасно одержувати пряме та інверсне значення реалізованої функції. Це дає помітне зниження загальної кількості мікросхем в апаратурі.

Особливостями схемотехніки ЕЗЛ та її характеристик є:

- можливість об'єднання виходів декількох елементів для утворення нових функцій;
- можливість роботи на низькоомному навантаженні завдяки наявності емітерних повторювачів;
- невелике значення роботи перемикання і незалежність споживаної потужності від частоти перемикання;
- висока стабільність динамічних параметрів при зміні температури і напруги живлення;
- використання від'ємного джерела живлення і заземлення колекторних кіл, що зменшує залежність вихідних сигналів від перешкод у шинах живлення.

До недоліків елементів ЕЗЛ відносять складність схем, значне споживання потужності та труднощі узгодження з мікросхемами ТТЛ і ТТЛШ.

Промисловість випускає ряд серій ЕЗЛ: 100, 137, 138, 187, 223, 229, 700, 500 і К1500. Високі техніко-економічні характеристики мікросхем серій 500 і К1500 обумовили їхнє широке застосування у швидкодійних цифрових пристроях. Типові значення параметрів елементів ЕЗЛ наведені в табл. 2.7.

Таблиця 2.7

Номер серії елементів ЕЗЛ	t _P , нс	P _{СС} , мВт	АП, пДж
500	2	25	50
К1500	0,75	40	30

Мікросхеми ЕЗЛ серії 500 виготовляють за напівпровідниковою дифузійною планарно-епітаксальною технологією. Усі компоненти мікросхеми розташовують в одному кристалі кремнію й ізолюють обернено зміщеними $p-n$ переходами. Компоненти формуються дифузією домішок у тонкому епітаксальному прошарку монокристалічного кремнію.

Схема типового логічного елемента ЕЗЛ серії 500 показана на рис. 2.21.

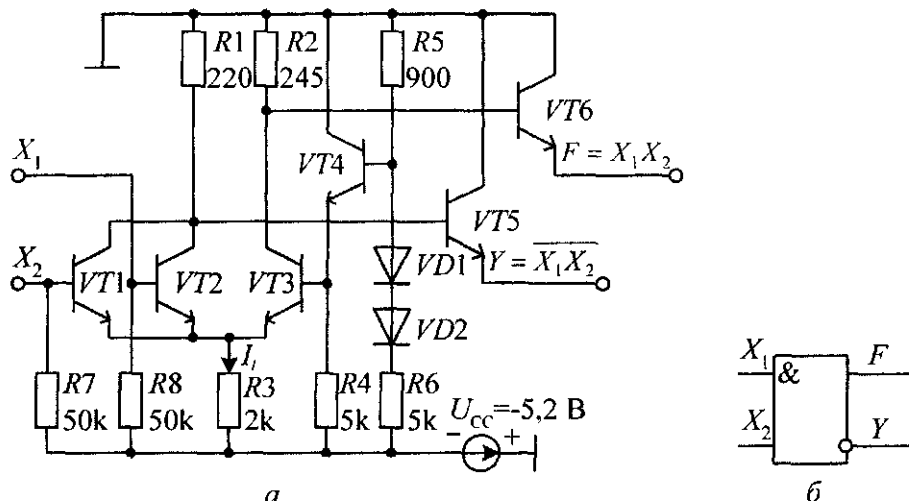


Рис. 2.21. Елемент ЕЗЛ серії 500: а — схема; б — умовне позначення

Схема ЕЗЛ вміщує

- перемикач струму (логічні транзистори $VT1$, $VT2$, опорний транзистор $VT3$, резистори $R1$, $R2$, $R3$),
- джерело опорного зміщення (транзистор $VT4$, діоди $VD1$, $VD2$, резистори $R5$, $R6$),
- вихідні емітерні повторювачі (транзистори $VT5$, $VT6$)

Напруга на відкритому переході база-емітер кремнієвого транзистора є постійним параметром $U^* = 0,8\text{ В}$. При описуванні роботи елемента беруть узгодження від'ємної логіки: значення лог 0 відображається високим (з урахуванням знака) рівнем напруги $U_H = -0,9\text{ В}$, значення лог 1 відображається низьким рівнем напруги $U_L = -1,7\text{ В}$. Амплітуда (перепад) логічного сигналу $U_m = U_H - U_L = 0,8\text{ В}$. Опорна напруга перемикача струму розташовується симетрично щодо рівнів двійкових сигналів і визначається як їхнє середнє $U_{оп} = -(U_H + U_L)/2 = -1,3\text{ В}$.

Якщо хоч на один із входів подана напруга $U_H = -0,9\text{ В}$, то даний транзистор відкривається, на нього перемикається струм I_1 , що створює на резисторі $R1$ падіння напруги мінус $0,9\text{ В}$. При цьому опорний транзистор закритий і на резисторі $R2$ падіння напруги дорівнює мінус $0,1\text{ В}$.

Якщо на усіх входах подані рівні $U_L = -1,7\text{ В}$, то транзистори $VT1$ і $VT2$ закриваються, а транзистор $VT3$ відкривається і на нього перемикається струм I_1 . У цьому випадку падіння напруги на резисторі $R1$ дорівнює мінус $0,1\text{ В}$, а на резисторі $R2$ мінус $0,9\text{ В}$. При цьому на колекторах лівого і правого плеча рівні напруги не відповідають взятому значенню двійкового сигналу. Емітерні повторювачі на транзисторах

$VT5$ і $VT6$ зміщують рівні сигналів, які надходять на їхні бази з лівого і правого плеча перемикача, на $U^* = 0,8$ В у бік від'ємних сигналів:

$$U_{OH} = -(U^* + 0,1) \text{ В} = -0,9 \text{ В}; \quad U_{OI} = -(U^* + 0,9) \text{ В} = -1,7 \text{ В}.$$

Внаслідок цього рівні вхідних і вихідних сигналів відповідають чинному стандарту. Часові діаграми роботи елемента ЕЗЛ показані на рис. 2.22.

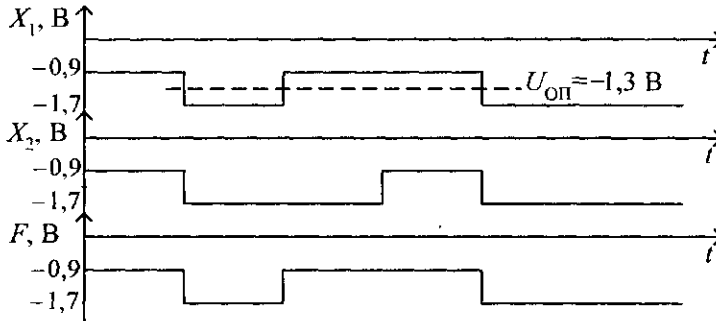


Рис. 2.22. Часові діаграми роботи елемента ЕЗЛ

Як випливає з часових діаграм, вхідні й вихідні логічні сигнали змінюються на $\pm 0,4$ В щодо опорної напруги.

У від'ємній логіці елемент ЕЗЛ реалізує на прямому виході F функцію І, а на інверсному виході Y — функцію НЕ І, що записується як НЕ-І/І. У позитивній логіці елемент ЕЗЛ реалізує функцію НЕ-ЧИ/ЧИ.

Резистори $R7$ і $R8$ опором 50 кОм підключені до джерела від'ємного живлення і забезпечують протікання зворотного базового струму, а також надійне запирання вхідних транзисторів, які не використовуються (незадіяні входи не потрібно підключати до лог. 1). Крім того, ці резистори служать навантаженням для джерел вхідних сигналів.

У елементах ЕЗЛ серії 500 навантажувальні резистори емітерного повторювача (ЕП) винесені за межі мікросхеми, що знижує на кристалі потужність, яка розсіюється, і забезпечує можливість організації монтажною логіки. Зовнішні навантажувальні резистори опором $0,3$ – $3,0$ кОм підключаються між виходом ЕП і основним джерелом від'ємного живлення мінус $5,2$ В. При роботі на низькоомне навантаження 50 – 200 Ом резистори можуть підключатися до додаткового зниженого джерела живлення мінус $2,0$ В.

Підключення від'ємної напруги живлення до емітерного кола і заземлення колекторів забезпечує кращу стійкість до перешкод і меншу залежність вихідних сигналів від наведень у колах емітера. Опір резистора $R3$ у декілька разів більший опорів $R1$ і $R2$, чим досягається сталість струму I_E у плечах перемикача струму. Крім того, співвідношення опорів резисторів $R1$, $R2$ і $R3$ підібрано так, щоб на виході закритого плеча встановлювалася напруга мінус $0,1$ В за рахунок протікання струму бази ЕП, а на виході відкритого плеча напруга дорівнювала мінус $0,9$ В за рахунок протікання струму I_E .

Колектори ЕП підключені до окремої шини землі; це пов'язано з тим, що струми повторювачів мають імпульсний характер і вони створюють перешкоди у провідниках, які підводять напругу.

У елементах ЕЗЛ допускається об'єднання виходів, як показано на рис 2 23 Об'єднання прямих виходів збільшує коефіцієнт об'єднання по І, об'єднання інверсних виходів дозволяє реалізувати функцію НЕ-І-ЧИ

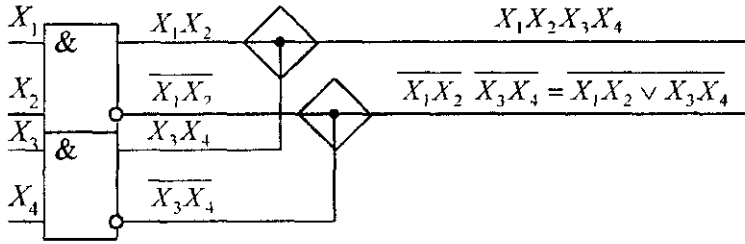


Рис 2 23. Об'єднання виходів елемента ЕЗЛ

Елементи ЕЗЛ серії 500 мають функціональну і технічну повноту Серія 500 вміщує широкий набір мікросхем логічні елементи, тригери, регістри, лічильники, дешифратори мультіплексори, АЛП

2.4. Логічні елементи на МОН-транзисторах

Розвиток комп'ютерної схемотехніки на основі МОН-транзисторів почався з появою в 1962 р польового транзистора з індукованим каналом Схеми на МОН-транзисторах характеризуються відносною простотою виготовлення, компактністю, малою споживаною потужністю, високою стійкістю до перешкод при зміні напруги живлення

МОН-транзистори мають структуру метал-діелектрик-напівпровідник і в загальному випадку називаються МДН-транзисторами (рис 2 24) Оскільки діелектрик реалізується на основі оксиду SiO_2 , то застосовують назву МОН-транзистори (уніполярні, каналні)

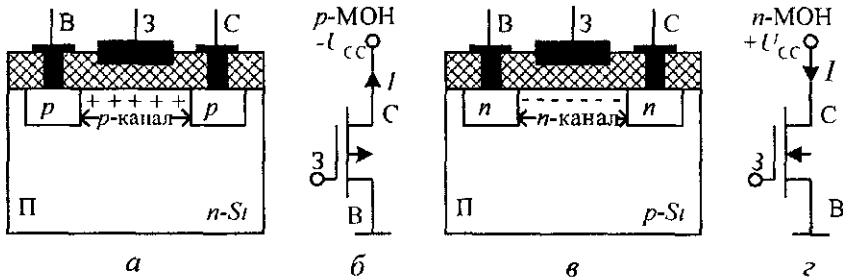


Рис. 2 24 Топологія і умовні позначення МОН-транзисторів. а,б — р-МОП; в,г — n-МОП

Металевий електрод, на який надходить керуюча напруга, називається **затвором** (З) а два інших електроди — **виток** (В) і **сток** (С) Від витоку до стоку протікає робочий струм Для р-каналу полярність стоку від'ємна, а для n-каналу — додатна Основна пластина напівпровідника називається **підкладкою** (П) **Канал** — це приповерхневий провідний прошарок між витоком і стоком, у якому значення струму визначається за допомогою електричного поля Процеси інжекції й дифузії в каналі відсутні Робочий струм у каналі обумовлений дрейфом в електричному полі електронів у n-каналах і дірок у р-каналах

При нульовому значенні керуючої напруги канал відсутній і струм не протікає. Канал, що утворюється під дією зовнішньої керуючої напруги, називається індукованим. Напруга, при якій утворюється канал, називається **пороговою** U_{TH} . Канал з початковою додатковою концентрацією зарядів називається **умонтованим**.

Швидкодія n -МОН транзисторів у 5–8 разів вища за швидкодію p -МОН транзисторів, оскільки рухливість електронів суттєво більша дірок. У МОН-схемах цілком виключені резистори, їхню роль виконують МОН-транзистори.

Схеми логічних елементів НЕ на МОН-транзисторах показані на рис. 2.25.

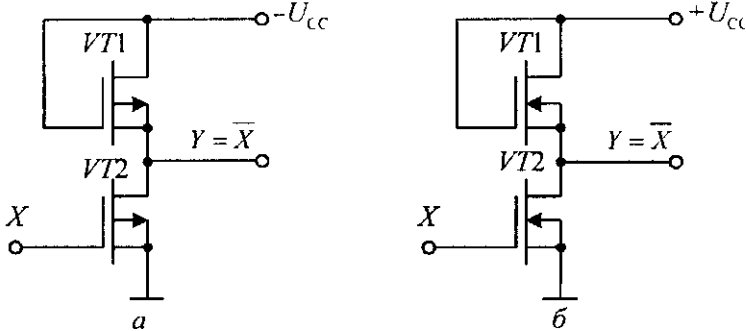


Рис. 2.25 Схеми елементів НЕ на МОН-транзисторах:
а — з p -каналами; б — з n -каналами

У схемі елемента НЕ на p -МОН транзисторах застосовують навантажувальний транзистор $VT1$, стік якого підключається до від'ємного джерела живлення мінус U_{CC} (рис. 2.25, а). Напруга від'ємної полярності вхідної змінної X надходить на затвор вхідного транзистора $VT2$. У цій схемі застосовують транзистори з індукованими каналами.

У схемі елемента НЕ на n -МОН транзисторах використовують навантажувальний транзистор $VT1$ із вбудованим каналом, який підключається до позитивного джерела живлення плюс U_{CC} . Позитивна напруга вхідної змінної X надходить на затвор вхідного транзистора $VT2$ з індукованим каналом (рис. 2.25, б). Навантажувальні транзистори увімкнені за схемою двополюсника.

Якщо вхідна напруга $U_i < U_{TH}$, то транзистор $VT2$ закритий, а $VT1$ — відкритий і на виході встановлюється рівень напруги, близький до значення U_{CC} .

Якщо вхідна напруга $U_i > U_{TH}$, то обидва транзистори відкриті і вихідна напруга знімається з дільника, утвореного опором каналів

$$U_0 = (U_{CC} R_B) / (R_B + R_H),$$

де R_B і R_H — опори каналів вхідного і навантажувального транзисторів. Порогові напруги для p -МОН $U_{TH} = -(5 \div 7)$ В, а для n -МОН $U_{TH} = +(1,5 \div 2)$ В.

Для того щоб напруга U_{OH} була меншою за 0,1 В, потрібно виконання умови $R_B \ll R_H$, яка досягається виготовленням транзисторів з різними опором каналів. У схемах на p -МОН транзисторах опір каналів транзистора $VT1$ у 25 разів більше опору каналу транзистора $VT2$. Специфіка n -МОН транзисторів така, що достатньо співвідношення опорів, що дорівнює біля чотирьох.

Схеми елементів НЕ ЧИ і НЕ І на n -МОН транзисторах показані на рис. 2.26.

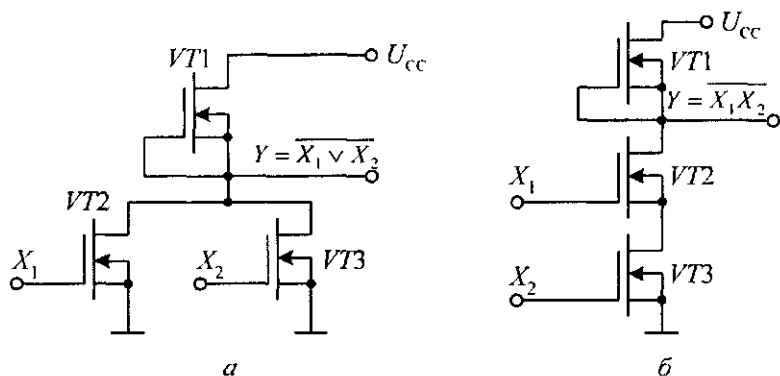


Рис. 2.26. Логічні елементи на n -МОП транзисторах: а — НЕ ЧИ; б — НЕ І

Елемент НЕ ЧИ утворюється паралельним з'єднанням вхідних транзисторів (рис. 2.26, а), а елемент НЕ І послідовним з'єднанням (рис. 2.26, б). Значення лог. 0 відображається напругою $U_L = 0,1$ В, а лог. 1 — напругою $U_H = U_{CC}$. На виході елемента НЕ ЧИ встановлюється інверсне значення логічної суми вхідних змінних, а на виході елемента НЕ І встановлюється інверсне значення логічного добутку вхідних змінних. Наприклад, на виході елемента НЕ ЧИ рівень лог. 0 встановлюється при наявності хоч на одному вході одиничного сигналу. На виході елемента НЕ І рівень лог. 0 встановлюється при збігу високих рівнів напруг на двох входах, коли одночасно відкриваються транзистори VT_2 і VT_3 .

У комплементарній МОП-структурі (логіка КМОП) використовуються одночасно p - і n -канальні транзистори. Елемент НЕ в схемотехніці КМОП побудовано на двох транзисторах з індукованими каналами: навантажувальному VT_2 з каналом p -типу і вхідному VT_1 з каналом n -типу (рис. 2.27, а). Витік транзистора VT_2 підключений до джерела позитивного живлення U_{CC} , напруга вхідної змінної X надходить на затвори обох транзисторів; вихідна напруга знімається з об'єднаних стоків.

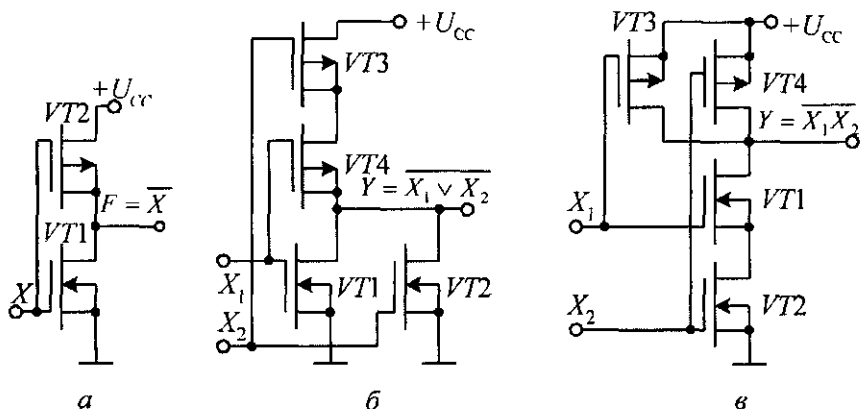


Рис. 2.27. Схеми елементів КМОП: а — НЕ; б — НЕ ЧИ; в — НЕ І

При вхідному рівні U_{IH} транзистор VT_1 відкритий, а VT_2 — закритий, оскільки між його затвором і витком є нульова напруга. На виході встановлюється рівень U_{OL} і струм у колі не протікає. При вхідному рівні U_{IL} транзистор VT_1 закритий, а VT_2 —

відкритий, тому що між його затвором і витоком є напруга U_{CC} . На виході — рівень U_{OH} і струм у колі не протікає.

На рис. 2.27, б показана схема КМОН елемента НЕ ЧИ, у якій входні транзистори VT_1 , VT_2 увімкнені паралельно, а навантажувальні VT_3 , VT_4 — послідовно. Якщо хоча б на одному із входів є рівень напруги U_{IH} , то даний транзистор відкривається, на виході встановлюється рівень U_{OI} , навантажувальні транзистори закриті, струм у колі не протікає. Якщо на обох входах є рівні напруги U_{II} , то транзистори VT_1 і VT_2 закриті, а VT_3 і VT_4 — відкриті й на виході встановлюється напруга U_{OH} і струм у колі не протікає.

На рис. 2.27, в показана схема КМОН елемента НЕ І, у якій входні транзистори VT_1 і VT_2 увімкнені послідовно, а навантажувальні VT_3 і VT_4 — паралельно. Якщо на затвори входних транзисторів одночасно надходять сигнали U_{IH} , то транзистори VT_1 і VT_2 відкриваються, вихідний рівень дорівнює U_{OI} , навантажувальні транзистори закриті, струм у колі не протікає. Якщо хоча б на одному із входів є рівень напруги U_{II} , то транзистори VT_1 і VT_2 закриті і відкривається один з навантажувальних транзисторів VT_3 або VT_4 . На виході встановлюється рівень U_{OH} і струм у колі не протікає.

Таким чином, у схемах КМОН у статичному стані протікає дуже малий робочий струм, оскільки при відкритих входних транзисторах закриті навантажувальні і навпаки. Сумарна потужність споживання в основному визначається енергією, яка витрачається на перезаряд паразитних ємностей.

Промисловість випускає такі серії КМОН: 176, 564, 561, КР1561 і КР1554. Мікросхеми швидкодіючої серії КР1554 мають функціональну і технічну повноту і вміщують логічні елементи, тригери, регістри, лічильники, дешифратори, мультиплексори і т. ін.

Елементи серії КР1554 характеризуються такими параметрами: діапазон напруги живлення становить 3–6 В, рівні логічних сигналів $U_{OI} = 0,1$ В, $U_{OH} = U_{CC}$, $U_{II} \leq 1,35$ В, $U_{IH} \geq 3,15$ В, струм споживання в статичному режимі на логічний елемент — 0,25 мкА (у тисячу разів менше порівняно із серією КР1533), потужність споживання на елемент — 2,5 мкВт, затримка поширення сигналів — 4 нс, робота перемикачання $A_{II} = 0,01$ пДж, частота перемикачання D-тригера — 150 МГц, діапазон температур мінус 40 — плюс 85°C. Як впливає з цих даних, мікросхеми КМОН серії КР1554 є серйозним конкурентом схемам ТТЛШ.

2.4.1. Логічні елементи на МЕН-транзисторах

Польові транзистори МЕН-типу мають структуру "метал-напівпровідник", їх будують на основі арсеніду галію (AsGa — сполука галію з миш'яком). Порівняно з германієм AsGa має такі переваги:

- вищу рухливість електронів у слабких електричних полях (приблизно у п'ять разів),
- майже в півтора раза ширшу заборонену зону, яка забезпечує високий питомий опір підкладки (як наслідок, підкладки з AsGa служать напівізолюючим матеріалом),
- дуже малі паразитні ємності між електродами МЕН-транзистора.

На основі арсеніду галію досягається десятикратне підвищення швидкодії схем

при зниженні потужності споживання вдвічі. Але арсенід галію не дозволяє будувати МОН-транзистори з ізольованим затвором, оскільки він не утворює стійких оксидів. В польових МЕН-транзисторах використовують бар'єр Шотки на границі контакту металу з напівпровідником. Тому МЕН-транзистори також називають "польовими транзисторами з бар'єром Шотки" (ПТШ).

Структура арсенід-галієвого МЕН-транзистора (поперечний розріз) показана на рис. 2.28, а. Транзистор створюється на підкладці з нелегованого AsGa. На поверхні підкладки іонним методом формують сильно леговані n^+ -області витоку і стоку, а потім тонкий шар каналу n -типу товщиною $d_0 = 0,2$ мкм. На поверхню підкладки в каналі наносять металевий електрод затвора (наприклад, сплав титан-вольфрам)

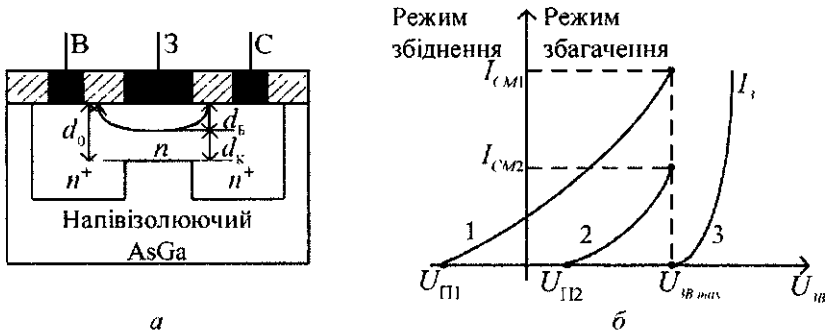


Рис. 2.28. МЕН-транзистор:

а — поперечний розріз; б — сток-затворні характеристики

Для забезпечення омичних контактів з витоком і стоком застосовують металеві електроди на основі композиції германій-золото. На поверхню підкладки між контактами наносять шар діелектрика, наприклад, діоксиду кремнію. Металевий електрод затвора створює в каналі збіднений електронами шар — бар'єр Шотки висотою 0,8 В. Просторові розміри бар'єру змінюються під дією напруги затвора. Власне провідний канал товщиною d_k обмежений областю бар'єра та підкладкою. Між затвором і витоком подається керуюча напруга $U_{зв}$, а на стік — напруга живлення плюс U_{CC} . При зміні керуючої напруги змінюється товщина збідненого шару d_b і провідного каналу $d_k = d_0 - d_b$, його провідність і струм стоку.

Порогова напруга U_{π} визначається з рівняння $U_{\pi} = C_1 - C_2 N_d d_0^2$, де C_1 , C_2 — константи, N_d — концентрація донорів у каналі. Якщо напруга затвор-витік $U_{зв}$ досягає порогового значення, то границя збідненого шару змикається з підкладкою $d_k = d_0 - d_b = 0$; при цьому товщина провідного шару і струм стоку I_c дорівнюють нулю.

Якщо при напрузі затвор-витік $U_{зв} = 0$ є провідний канал і протікає робочий струм, то МЕН-транзистор називається **нормально відкритим**, він працює в режимі збіднення. Якщо при $U_{зв} = 0$ бар'єр Шотки перебиває весь канал і робочий струм не протікає, то МЕН-транзистор називається **нормально закритим**; він працює в режимі збагачення.

На рис. 2.28, б наведені стік-затворні характеристики нормально відкритого (крива 1) і нормально закритого (крива 2) МЕН-транзистора, а також їхня вхідна характеристика (крива 3). Для нормально відкритих МЕН-транзисторів керуюча напруга на затворі, при якій протікає струм стоку I_c , може змінюватися від від'ємних значень

до невеликих додатних (не більше 0,6 В). При напругах, більших $U_{3В} > 0,6$ В в його каналі з'являється струм затвора I_3 , оскільки відкривається перехід метал-напівпровідник. Тому струм стоку обмежений значенням I_{CM1} . Для нормально закритих транзисторів напруга затвора, при якій протікає струм стоку, додатна і може змінюватися тільки у вузьких границях (0...0,6 В). Максимальний струм стоку обмежений значенням I_{CM2} . Для транзисторів з однаковими розмірами каналу (довжиною і шириною) $I_{CM1} \gg I_{CM2}$. У схемотехніці застосовують нормально закриті й нормально відкриті МЕН-транзистори.

Варіанти схем логічних елементів НЕ, НЕ ЧИ на МЕН-транзисторах показані на рис. 2.29.

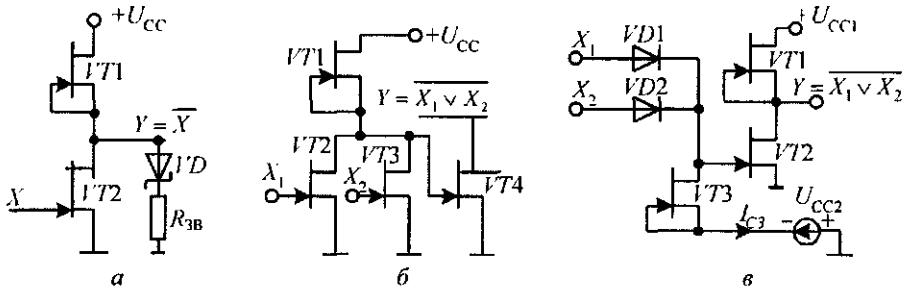


Рис. 2.29. Логічні елементи на МЕН-транзисторах: а — НЕ; б, в — НЕ ЧИ

Схема НЕ (рис. 2.29, а) містить пасивний транзистор $VT1$ (нормально відкритий) і вхідний активний транзистор $VT2$ (нормально закритий). Пасивний транзистор $VT1$ увімкнений за схемою двополюсника і виконує роль джерела стокового струму, значення якого практично не змінюється в широкому діапазоні зміни напруги між стоком і витокком.

До виходу елемента НЕ підключається аналогічний інвертор. Він у статичному режимі подається еквівалентною схемою з послідовно увімкнених діода Шотки (бар'єра метал-напівпровідник) і опору між затвором і витокком $R_{3В}$. Напруга джерела живлення у схемі $U_{CC} = 1,5$ В; усереднені значення порогових напруг для транзисторів $VT1$ і $VT2$ відповідно дорівнюють $U_{П1} = -0,3$ В і $U_{П2} = 0,15$ В.

При $U_{ВХ} = U_{П1} < U_{П2}$ транзистор $VT2$ закритий, струм стоку $I_{C2} = 0$ і на виході устанавлюється рівень напруги $U_{ОН}$. Струм I_{C1} відкритого транзистора $VT1$ перемикається в затвор інвертора навантаження. Оскільки струм I_{C2} становить одиниці міліамперів, а опір $R_{3В}$ вимірюється десятками омів, то вихідний рівень практично визначається прямою напругою на діоді VD :

$$U_{ОН} = U_{ДШ} + I_{C1} R_{3В} \approx U_{ДШ} = 0,6 \text{ В.}$$

Рівень напруги $U_{ОН}$ мало залежить від значень джерела живлення U_{CC} , струму I_{C1} і знижується із збільшенням коефіцієнта об'єднання та температури.

При $U_{ВХ} = U_{П1} > U_{П2}$ транзистор $VT2$ відкривається і через нього протікають струми стоків транзистора $VT1$ і попереднього транзистора (джерело вхідного сигналу). На виході встановлюється низький рівень $U_{ОЛ} = 0,05$ В.

Рівень $U_{ОЛ}$ знижується із зменшенням струмів стоків і опору $R_{СВ}$ між стоком і витокком, який становить одиниці слів. Якщо припустити, що $U^1 = U_{Н}$, $U^0 = U_{Л}$, то амплітуда логічного сигналу $U_m = U^1 - U^0 = 0,55$ В. Завадостійкість елемента НЕ $M_L = 0,16$ В, $M_H = 0,26$ В, що значно менше, ніж для схем на n -МОН транзисторах.

Схема двохвального елемента НЕ ЧИ містить нормально відкритий пасивний транзистор $VT1$ вхідні нормально закриті транзистори $VT2$ і $VT3$ увімкнені паралельно (рис 2 29, б)

При $U_{X1} = U_{X2} = U_{II}$ транзистори $VT2$ і $VT3$ закриті на виході Y встановлюється високий рівень напруги U_{OH} . Якщо на одному із входів або на обох діє напруга U_{IH} , то відповідний транзистор (чи обидва) відкриваються і на виході встановлюється рівень U_{OI} . Параметри елемента НЕ ЧИ аналогічні схемам, зображеним на рис 2 29, а

Схема двохвального елемента НЕ ЧИ з підвищеною завадостійкістю показана на рис 2 29, в. Вхідні діоди Шотки $VD1$ і $VD2$ реалізують операцію ЧИ, транзистори $VT1$ і $VT2$ створюють інвертор, а транзистор $VT3$ разом з діодами — це ланцюг зміщення рівня порога транзистора $VT2$.

В даному елементі порівняно зі схемами, зображеними на рис 2 29, а і б, забезпечується більша завадостійкість і менша її залежність від технологічного розкиду порогових напруг транзисторів. Це досягається ускладненням схеми, збільшенням її площі на кристалі та використанням другого джерела живлення мінус U_{CC2} . У схемі застосовують тільки нормально відкриті транзистори з напругами живлення $U_{CC1} = 1,5$ В, $U_{CC2} = -1$ В. Для транзисторів $VT1$ і $VT3$ $U_{П1} = U_{П3} = -0,7$ В, а для транзистора $VT2$ $U_{П2} = -0,45$ В.

Пряма напруга на діодах Шотки при протіканні через них струму I_{C3} дорівнює приблизно 0,7 В (мінус на катоді). Напруга на затворі транзистора $VT2$ визначається із залежності $U_{3B} = U_{BX} - U_{ДШ}$, яка є практично лінійною, оскільки $U_{ДШ} = \text{const}$.

При $U_{BX} = U_{II} = 0$ напруга $U_{3B} = -0,7$ В, що менше від $U_{П2}$, і транзистор $VT2$ закритий, на його виході встановлюється високий рівень напруги $U_{OH} \approx 1$ В. При $U_{BX} = U_{IH} = 1$ В напруга $U_{3B} = +0,3$ В, що більше від $U_{П2}$, і транзистор $VT2$ відкритий, на його виході встановлюється низький рівень напруги $U_{OI} \approx 0,01$ В. Таким чином, використання ланцюга зміщення дозволяє зменшити значення U^0 , збільшити U^1 і логічний перепад $U_m = U^1 - U^0 = 1$ В.

Промисловість випускає схеми на МЕН-транзисторах серії К6500, які характеризуються такими параметрами: $U_{CC1} = 4$ В, $U_{CC2} = -2,4$ В, $U_H = 0,9$ В, $U_L = 0,1$ В, середня затримка поширення $t_{П} = 0,15$ нс, потужність, споживана одним логічним елементом, $P_{CC} = 5$ мВт.

2.5. Імпульсна і потенціально-імпульсна системи елементів. Магнітна схемотехніка

2.5.1. Імпульсна система елементів

В імпульсній системі елементів використовуються тільки імпульсні інформаційні сигнали. У цій системі звичайно застосовуються логічні елементи ЧИ, І, НЕ та імпульсні (динамічні) тригери. Інформаційні сигнали з виходів одних імпульсних елементів надходять до входів інших через конденсатори, обмотки трансформаторів, які виключають взаємозв'язок елементів за постійним струмом.

Імпульсні елементи характеризуються відносно простою схемою і малою споживаною потужністю. Недоліком імпульсних елементів є труднощі реалізації конденсаторів великої ємності та трансформаторів методами інтегральної технології.

Схема імпульсного діодного елемента ЧИ показана на рис 2 30, а, вона відрізняється від аналогічного потенціального елемента наявністю на вході розподільного RC -ланцюга, причому постійна часу $\tau = RC \gg t_i$, де t_i — тривалість вхідних інформаційних імпульсів. Сигнал на виході елемента ЧИ з'являється у випадку наявності імпульсу хоча б на одному вході (рис. 2 30, б)

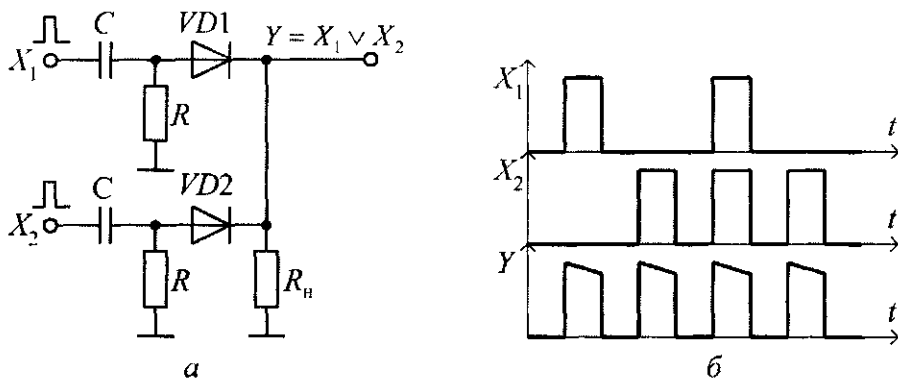


Рис. 2.30. Імпульсний діодний елемент ЧИ:

а — схема; б — часові діаграми роботи

Схема імпульсного діодного елемента І показана на рис 2 31, а, вона відрізняється від аналогічного потенціального елемента наявністю на вході розподільного RC -ланцюга, причому постійна часу $\tau = RC \gg t_i$, і має виконуватися умова $R1 \gg R$. Сигнал на виході елемента І з'являється при збігу імпульсів на всіх входах (рис 2 31, б)

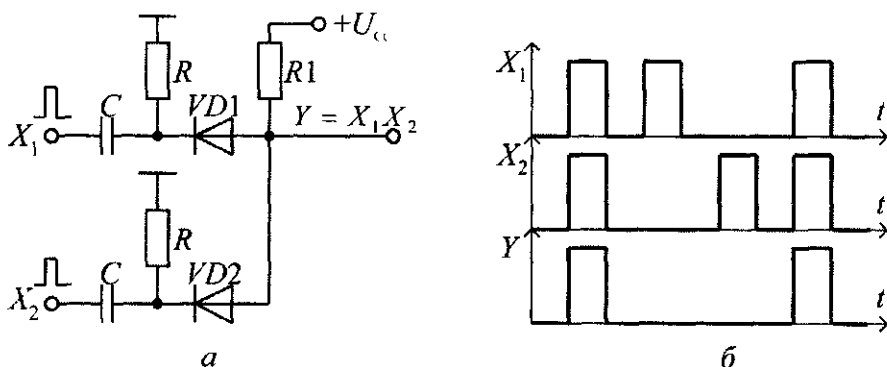


Рис. 2.31. Імпульсний діодний елемент І:

а — схема; б — часові діаграми роботи

Схема імпульсного елемента ЗАБОРОНА, яка реалізує логічну функцію $Y = X_1 \overline{X_2}$, показана на рис 2 32, а. Якщо $X_2 = 0$, то на виході елемента повторюється значення змінної X_1 . При наявності імпульсу на вході X_2 на вторинній обмотці W_2 трансформатора Tr наводиться напруга, яка компенсує зачіняюче зміщення плюс U_{3M} . З цієї причини діод $VD2$ відкривається і вихід елемента практично підключається до потенціалу землі, тобто забезпечується значення лог 0 (рис. 2.32, б).

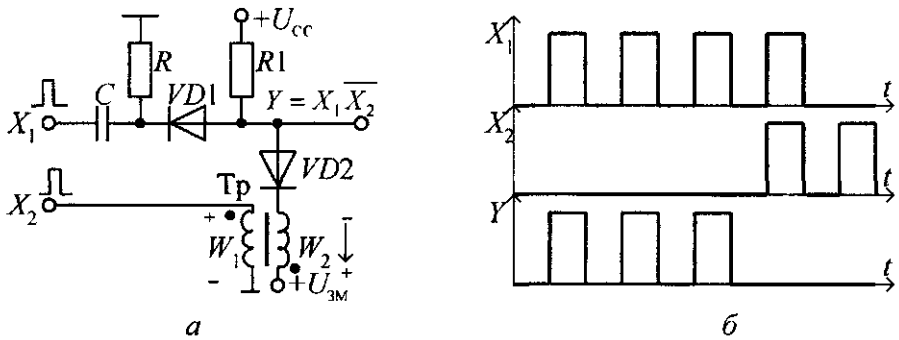


Рис. 2.32. Імпульсний діодний елемент ЗАБОРОНА: а — схема; б — часові діаграми роботи

В імпульсних схемах часто використовуються штучні лінії затримки, які складаються з LC -ланки Т-подібного типу. Тривалість затримки визначається із співвідношення $t_3 = n\sqrt{LC}$, де n — кількість ланок, L — індуктивність, мкГн; C — ємність, Пф (рис. 2.33, а).

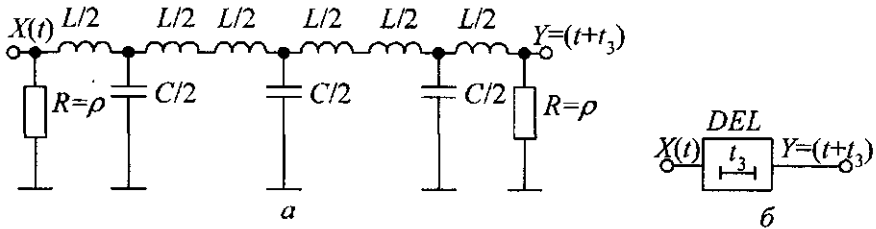


Рис. 2.33. Штучна лінія затримки: а — схема; б — умовне позначення

Для виключення перешкод на вході та виході лінії затримки вмикають резистори з хвильовим опором $\rho = \sqrt{L/C}$.

2.5.2. Потенціально-імпульсна система елементів

У потенціально-імпульсній системі елементів використовують потенціальні та імпульсні інформаційні сигнали. У цій системі застосовують як чисто імпульсні та потенціальні елементи, так і спеціальні потенціально-імпульсні схеми на основі діодів, транзисторів і трансформаторів. Потенціально-імпульсні елементи широко застосовувались у комп'ютерах першого та другого покоління; зараз їх використовують у спеціалізованих цифрових пристроях. Потенціально-імпульсні елементи за енергоспоживанням займають проміжне положення порівняно з імпульсними і потенціальними схемами. Схема потенціально-імпульсного діодно-трансформаторного логічного елемента І ЧИ, який реалізує функцію $Y = I_1\Pi_1 \vee I_2\Pi_2$, де I_1, I_2 — імпульсні сигнали; Π_1, Π_2 — потенціальні сигнали, показана на рис. 2.34, а. Наявність імпульсу позитивної полярності заданої амплітуди відображає лог. 1, а його відсутність — лог. 0. Низький рівень потенціального сигналу кодує лог. 1, високий рівень — лог. 0.

У діодно-трансформаторній схемі І ЧИ діоди $VD1$ і $VD2$ виконують роль ключів: вони відкриваються в тому випадку, коли на аноді діє відпираючий позитивний імпульс, а на катоді — потенціал землі. При цьому до первинної обмотки W_{11} або W_{12} прикладається імпульс напруги, який трансформується на вихідній обмотці W_2

трансформатора Tr (рис. 2.34, б). Резистор $R_{ш}$ і діод $D_{ш}$ утворюють шунтуючий (демпфуючий) ланцюг, який зменшує вихідні післяімпульсні коливання. При наявності на потенціальних входах Π_1 та Π_2 високого рівня напруги, діоди $VD1$ і $VD2$ закриваються, і первинні обмотки відключаються від імпульсів напруги: на виході імпульсний сигнал відсутній.

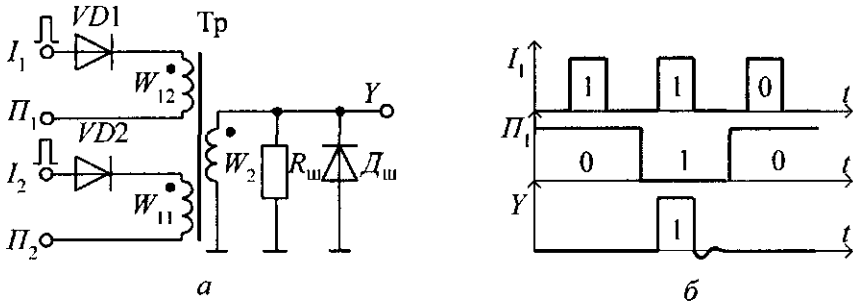


Рис. 2.34. Потенціально-імпульсний елемент I CI: а — схема; б — часові діаграми роботи

Схема потенціально-імпульсного елемента I CI з підсилювачем-формувавцем на виході показана на рис. 2.35. Підсилювач-формувавч побудований на транзисторі $VT1$ з імпульсним трансформатором $Tr2$ у електричному колі колектора.

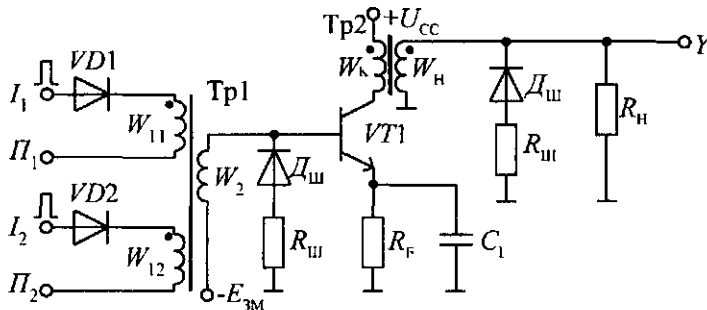


Рис. 2.35. Елемент I CI з підсилювачем-формувавцем

Основне призначення підсилювача-формувача полягає в тому, щоб забезпечити вихідний сигнал необхідної форми (переважно прямокутний), амплітуди і тривалості. У початковому стані транзистор $n-p-n$ типу $VT1$ закритий негативною напругою зміщення $E_{3m} = -1$ В, яка подається через обмотку W_2 трансформатора $Tr1$ на базу, струм в електричному колі колектора не протікає, і вихідний імпульс відсутній.

При збігові імпульсного і потенціального сигналів на входах обмоток W_{11} або W_{12} (або обох одночасно) на вихідній обмотці трансформатора $Tr1$ індукуюється напруга, яка компенсує зачиняючу напругу й відкриває транзистор $VT1$. Потенціал між колектором і емітером насиченого транзистора близький до нуля, тому напруга джерела живлення U_{cc} практично повністю прикладається до обмотки W_k трансформатора $Tr2$. На вихідній обмотці W_n формується імпульс напруги з постійною амплітудою $U_m = (U_{cc} W_n) / W_k$. Тривалість вихідного імпульсу визначається часом заряду емітерним струмом конденсатора C_E до рівня напруги, яка закриває транзистор $VT1$. Ланцюг з резистора $R_{ш}$ і діода $D_{ш}$ зменшує післяімпульсні викиди у вторинних обмотках трансформаторів.

Для підвищення надійності та завадостійкості цифрових систем використовують

мажоритарні логічні елементи. Мажоритарні елементи, інвертор, константи "0" і "1" створюють функціонально повну систему логічних елементів.

Мажоритарний логічний елемент має непарну кількість входів $n = 3, 5, 7, \dots$ і один вихід, стан якого визначається за більшістю входів. У мінімізованій диз'юнктивній нормальній формі мажоритарної функції в кожному добутку є $m = (n+1)/2$ змінних без інверсії. Наприклад, мажоритарна функція для $n = 3$ має вигляд

$$M(X_1, X_2, X_3) = X_1 X_2 \vee X_1 X_3 \vee X_2 X_3$$

У 1960 р. була введена операція мажоритарності із символічним зображенням #. Тоді мажоритарна функція на n входів має вигляд

$$M(X_1, X_2, \dots, X_n) = X_1 \# X_2 \# X_3 \# \dots \# X_n$$

Найбільше практичне використання знайшли мажоритарні елементи з кількістю входів $n = 3$ (рис. 2.36) рідше — з $n = 5$.

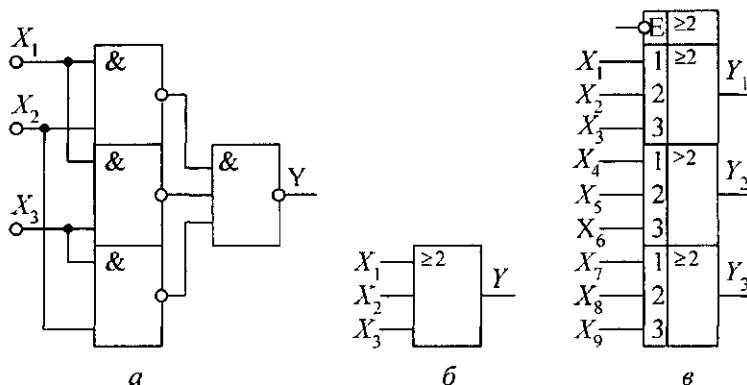


Рис. 2.36. Мажоритарні елементи: а — схема на елементах НЕ І;

б — умовне позначення; в — потрібний мажоритарний елемент типу КР1533ЛПЗ

Мікросхема КР1533ЛПЗ (рис. 2.36, в) являє собою три мажоритарних елементи зі спільним входом стробування E . При $E = 0$ логічний стан кожного виходу визначається збігом одиниць на будь-яких двох входах з трьох. Якщо $E = 1$, то виходи елементів повторюють стан третього входу.

2.5.3. Загальна характеристика магнітних схем

Магнітні схеми (МС) будують на основі електромагнітного кола, частиною якого є магнітний матеріал. Їх застосовують для перетворення, обробки і зберігання інформації. Для побудови МС використовують ряд фізичних явищ: феромагнітних, магнітно-напівпровідникових, магнітооптичних, надпровідності та інших.

Явище феромагнетизму характеризується

- нелінійним характером процесу намагнічення і високою магнітною проникливістю,
- ефектом магнітного гістерезису, який використовується для зберігання інформації.

До переваг МС відносять високу надійність роботи і радіаційну стійкість, зберігання інформації без споживання енергії, високі температурну стабільність і стійкість до перешкод.

Застосовують декілька методів інтеграції й мініатюризації МС:

- конструктивний (технологічний) — перехід до інтегральної схемотехніки;
- функціональний — МС виконує декілька функцій;
- фізичний — в одному й тому ж феромагнетикі використовуються різноманітні фізичні явища, наприклад, магнітні й магнітоакустичні

За видом оброблюваної інформації МС поділяються на аналогові та цифрові. В магнітній схемотехніці виділяють такі основні напрямки:

- на кільцевих осердях,
- на конструкціях із складним магнітопроводом,
- на магнітних доменах і ефекті Джозефсона

Магнітні схеми на кільцевих осердях

У комп'ютерній схемотехніці застосовують магнітні елементи на феритових кільцевих осердях. Гранична крива намагнічування $B = f(H)$ являє собою майже прямокутну петлю гістерезису (рис. 2.37, а). На цій характеристиці вживані такі позначення: H — напруга магнітного поля; B — магнітна індукція; $\pm B_r$ — залишкова магнітна індукція; B_m — індукція насичення (максимальне значення), що досягається в осерді під дією імпульсу напруги H_m .

Найпростішим магнітним елементом є кільцеве феритове осердя з інформаційною, керуючою та вихідною обмотками (рис. 2.37, б).

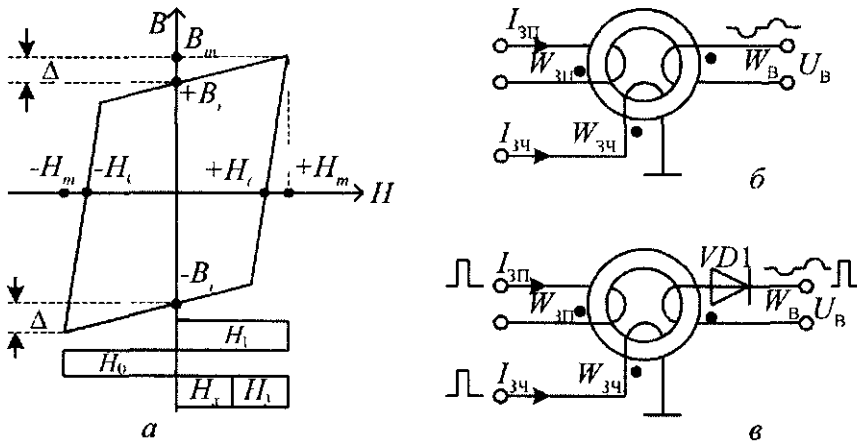


Рис. 2.37. Магнітні схеми: а — крива намагнічення; б, в — магнітні елементи

Інформаційна $W_{3П}$ і керуюча $W_{3Ч}$ обмотки служать для записування та зчитування інформації. Вихідна обмотка W_B призначена для отримання електричного сигналу, який відображає значення функції елемента. У загальному випадку кількість обмоток, їхнє використання і найменування визначаються призначенням даного магнітного елемента.

Зазвичай припускають, що позитивна залишкова індукція плюс B_r відображає стан "1", а від'ємна мінус B_r — стан "0". Початки обмоток осердя позначені точками. Умовно вважають, що струм зчитування $I_{3Ч}$, який надходить до початку обмотки $W_{3Ч}$, намагнічує осердя у стан "0", а струм записування $I_{3П}$, який витікає з початку обмотки $W_{3П}$, перемагнічує осердя у стан "1". Намагнічувальна сила, яка діє в осерді, визна-

часться алгебраїчною сумою ампервитків у всіх одночасно діючих обмотках. Тому, залежно від напрямку струму і увімкнення обмоток, окремі складові магнітної індукції можуть додаватися або відніматися, що розширює можливості використання магнітних елементів. Струм записування і струм зчитування повинні мати амплітуду, яка забезпечує розрахункове значення напруги поля H_m і тривалість, не меншу за час перемикання τ осердя.

У процесі переманічування осердя з одного стану в інший на вихідній обмотці W_B індукується знакозмінний імпульс напруги $U_m = \pm W_B S \Delta B / \tau$, де W_B — кількість витків вихідної обмотки, S — площа поперечного перерізу осердя, $\Delta B = 2B_r + \Delta$, де $\Delta = B_m - B_r$, τ — час перемикання (рис. 2.37, б). Перемикання осердя в стан "1" супроводжується індуктуванням на вихідній обмотці негативного імпульсу напруги, для його відсікання від навантаження у вихідному колі вмикається обмежувачий діод $VD1$ (рис. 2.37, в).

При повторній подачі імпульсів записування або зчитування магнітна індукція змінюється на Δ і на виході виникають імпульси перешкоди невеликої амплітуди. До особливостей кодування інформації відноситься те, що вхідний електричний сигнал записування перетворюється в магнітну залишкову індукцію і тільки при зчитуванні знову перетворюється в електричний сигнал. Наявність позитивного імпульсу на вихідній обмотці в момент зчитування вважається лог "1", а його відсутність — лог "0".

Для зображення магнітного елемента використовують "дзеркальний" метод, при якому осердя позначають потовщеною вертикальною лінією, обмотки — у вигляді горизонтальних тонких ліній. Струм по вхідних обмотках протікає зліва направо. Спосіб увімкнення обмоток зображений тонкими похилими лініями під кутом 45° . Вхідний струм, відображаючись по похилій лінії вгору, перемикає осердя у стан "1", а відображаючись униз — у стан "0". На вихідній обмотці виникнення позитивного імпульсу напруги відбувається в момент перемикання із стану "1" у стан "0" (рис. 2.38).

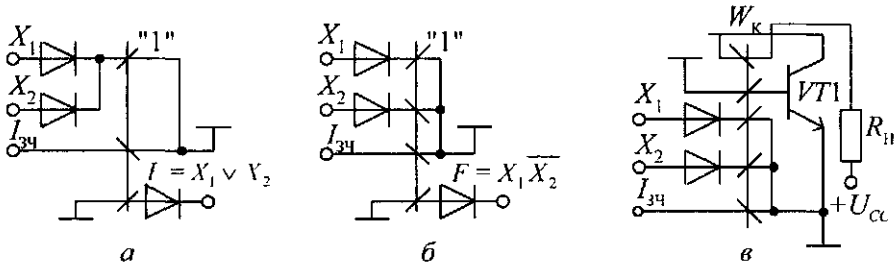


Рис. 2.38. Феродіоди (а, б) і феротранзисторні (в) логічні елементи

У магнітних схемах використовують феродіодні та феротранзисторні елементи.

Функціонально повним набором логічних елементів на феродіодних схемах є диз'юнктор, ЗАБОРОНА та генератор "1". Схема диз'юнктора, який реалізує функцію $F = X_1 \vee X_2$ показана на рис. 2.38, а. Осердя перемикається в стан "1" у першому такті сигналами X_1 чи X_2 або одночасно обома. Вихідний сигнал F з'являється у вихідній обмотці в другому такті при зчитуванні. Схема ЗАБОРОНА, яка реалізує функцію $F = X_1 \overline{X_2}$ показана на рис. 2.38, б. Інформаційні вхідні обмотки увімкнені стрічно, тому перемикання осердя можливе тільки за умови $X_1 = 1, X_2 = 0$.

Спрощена схема феротранзисторного диз'юнктора, в якій за допомогою транзистора $VT1$ і колекторної обмотки W_K утворений позитивний зворотний зв'язок, що

прискорює перемагнічування осердя зі стану "1" у стан "0", показана на рис. 2.38, в. У початковому стані транзистор $VT1$ закритий, при зчитуванні лог. 1 на базовій обмотці виникає позитивний імпульс напруги, який відкриває транзистор $VT1$. Струм колектора створює додатковий перемагнічувальний потік, який діє згідно з потоком обмотки зчитування. В результаті час перемикання осердя значно зменшується.

Універсальна магнітна схема для реалізації восьми мінтермів для трьох перемінних X_1, X_2, X_3 (рис. 2.39) має вигляд:

$$M_0 = \overline{X_1} \overline{X_2} \overline{X_3}; M_1 = \overline{X_1} \overline{X_2} X_3; M_2 = \overline{X_1} X_2 \overline{X_3};$$

$$M_3 = \overline{X_1} X_2 X_3; M_4 = X_1 \overline{X_2} \overline{X_3}; M_5 = X_1 \overline{X_2} X_3;$$

$$M_6 = X_1 X_2 \overline{X_3}; M_7 = X_1 X_2 X_3.$$

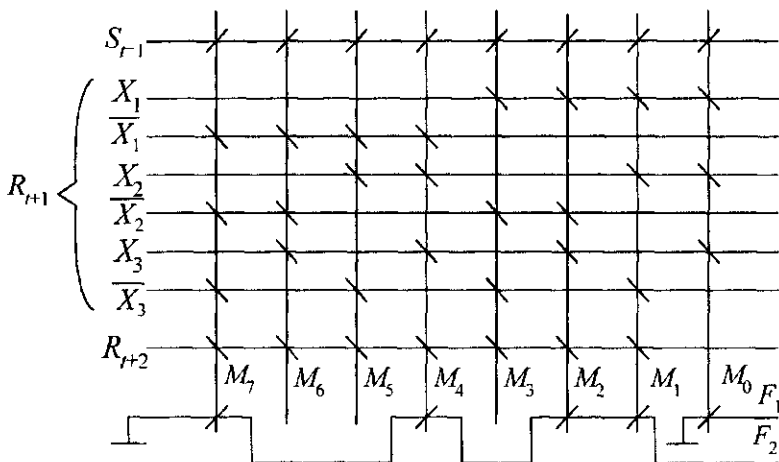


Рис. 2.39. Універсальна магнітна схема

Якщо прошити загальним проводом декілька осердь, то реалізується диз'юнкція мінтермів. Для прикладу в схемі (рис. 2.39) за допомогою прошивання реалізуються такі функції:

$$F_1 = M_0 = \overline{X_1} \overline{X_2} \overline{X_3};$$

$$F_2 = M_7 \vee M_4 \vee M_2 \vee M_1 = X_1 X_2 X_3 \vee X_1 \overline{X_2} \overline{X_3} \vee \overline{X_1} X_2 \overline{X_3} \vee \overline{X_1} \overline{X_2} X_3.$$

Для підсилювання сигналів у з'єднаннях між магнітними схемами використовуються інтегральні підсилювачі й формувачі струму (серії 146, 169, 170), а також діодні та транзисторні мікровузли.

2.5.5. Магнітні елементи із складним магнітопроводом

До магнітних елементів із складним магнітопроводом відносяться багатоотвірні пластини, трансфлюксори та ряд інших.

Багатоотвірні пластини — це конструкції з фериту, які мають до ста отворів діаметром 0,65 мм. На такій пластині може розташовуватися більше 25 RS-тригерів або декілька десятків логічних елементів І ЧИ, НЕ ЧИ.

Трансфлюксор — це феритовий диск з двома неоднаковими отворами (рис 2 40, а), що може знаходитися у двох станах, які характеризуються різним розподілом магнітного поля навколо великого отвору і відображають лог 1 (рис 2 40, б) і лог 0 (рис 2 40, в) Характерною особливістю *RS*-тригера на трансфлюксорі є зчитування інформації без її руйнування

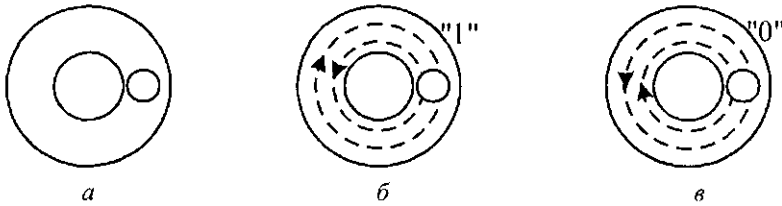


Рис. 2.40. Трансфлюксор: а — конструкція, б, в — стани "1" і "0".

Кріоелектронні магнітні елементи

Кріоелектронні магнітні елементи будують на основі надпровідності, при якій електричний опір ряду металів і сплавів при їхньому охолодженні стрибком спадає до нуля — нижче певної критичної температури T_K . Під дією керуючого магнітного поля з напругою, вищою за критичну H_K , надпровідність зникає. Чим більше охолоджений матеріал, тим більше магнітне поле потрібне для руйнування надпровідності.

Найпростіший кріоелектронний логічний елемент з двома входами та одним виходом називається **кріотроном**. Конструкційно кріотрон виконаний у вигляді двох тонкоплівкових металевих штаб, розділених діелектриком і поміщених на надпровідний металевий екран. Штабу для робочого струму називають вентилям "В", а для керуючого струму — затвором "З". Якщо затвор поміщений поперек вентиля, кріотрон називають поперечним (рис 2 41, а), а якщо паралельно — поздовжнім (рис 2 41, б).

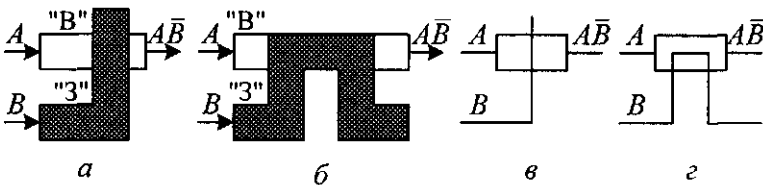


Рис. 2.41. Кріотрони: а — поперечний, б — поздовжній, в, г — умовне позначення

При подачі в затвор струму керування, який створює критичне магнітне поле H_K , вентиль перемикається із стану надпровідності в нормальний омичний стан, при цьому його опір підвищується в $10^{12} - 10^{17}$ разів. Надпровідний стан затвору не змінюється, оскільки він виготовлений з матеріалу з вищим значенням магнітного поля H_K , ніж вентиль. Якщо робочий струм вентиля відображає змінну A , а керуючий струм затвору — змінну B , то кріотрон виконує функцію заборони AB . Увімкнення кріотронів для реалізації елементарних логічних функцій змінних показано на рис 2 42.

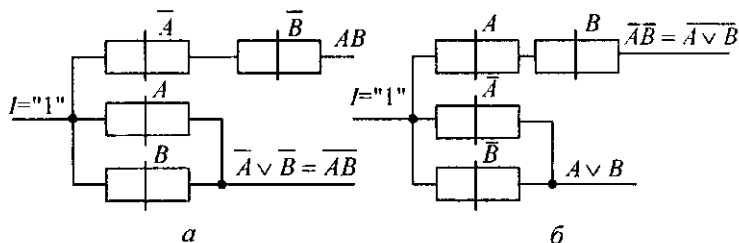


Рис. 2.42. Логічні елементи на кріотронах: а — І, НЕ І; б — ЧИ, НЕ ЧИ

У 1962 р. англійський фізик Б. Джозефсон показав, що надпровідність може виникати в пластинках, розділених діелектриком, товщиною в декілька нанометрів (рис. 2.43, а). При цьому відбувається тунелювання електронів. Конструкції, які називаються тунельними переходами або перемикачами Джозефсона (ПД), мають вольт-амперну характеристику, яка показана на рис. 2.43, б.

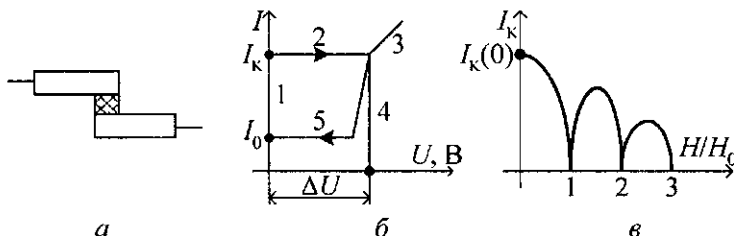


Рис. 2.43. Перемикач Джозефсона:

а — конструкція переходу; б — вольт-амперна характеристика;
в — залежність $I_k = f(H, H_0)$

При струмі $I_0 < I < I_k$ (ділянка 1) перемикач знаходиться у стані надпровідності. При досягненні значення струму $I \geq I_k$ (ділянка 2) надпровідність зникає і на ПД стрибком з'являється напруга $\Delta U \approx 3$ мВ. Подальше збільшення струму I (ділянка 3) супроводжується майже лінійним зростанням напруги на ПД. Зменшення струму можливе до мінімального значення I_0 (ділянка 4), після чого відбувається стрибкоподібне повернення у стан надпровідності (ділянка 1). Така характеристика з петлею гістерезису дозволяє будувати на ПД логічні елементи. Наприклад, стану надпровідності приписується значення лог. 0, а його відсутності, при якій на ПД існує напруга, — значення лог. 1.

Магнітне поле "сплющує" петлю гістерезису у вертикальному напрямку (рис. 2.43, в), в результаті чого ПД перемикається у нормальний стан (так зване "вмикання") при меншому струмі I_k . Зворотнє перемикання відбувається або після зняття вхідних сигналів, або спеціальним зниженням струму ПД нижче рівня I_0 .

Спрощені схеми логічних елементів на ПД показані на рис. 2.44.

Живлення схем — імпульсне; струм живлення дорівнює 0,32 мА; критичний струм — 0,4 мА; керуючий струм — 0,2 мА. Схема ЧИ включається на входах А або В, а схема І — при збігу сигналів на входах А і В. Сигнал лог. 1 на виході елемента ЗАБОРОНА формується за умови, що в момент надходження тактового імпульсу $A = 1$ і $B = 0$.

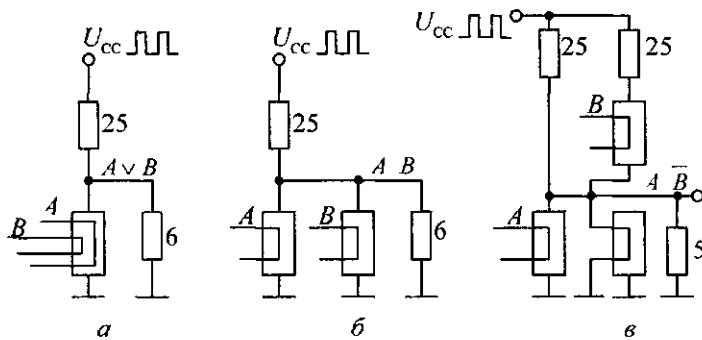


Рис 2.44 Логічні елементи на ПД а — ЧИ, б — І, в — ЗАБОРОНА

Час перемикання ПД складає пікосекунди, робота перемикання дорівнює 10^6 пДж Щільність компонування й ступінь інтеграції криоелектронних елементів досить високі

До недоліків криоелектронних елементів відноситься складність забезпечення компактного, економічного й надійного охолодження

2.6. Асинхронні та синхронні RS-тригери

2.6.1. Визначення та призначення тригерів

Тригер — це запам'ятовуючий елемент з двома стійкими станами, зміна яких відбувається під дією входних сигналів Як елемент комп'ютера, тригер призначений для зберігання одного біта інформації, тобто лог 0 або лог 1 Схеми тригера забезпечує записування, зчитування, стирання та індикацію двійкової інформації яка зберігається На основі тригерів будують типові функціональні вузли комп'ютерів — регістри, лічильники, накопичувальні суматори, а також мікропрограмні автомати

Усі різновиди тригерів являють собою елементарний автомат який вміщує власне елемент пам'яті (ЕП) та схему керування (СхК), яка утворює входну логіку (рис 2.45)

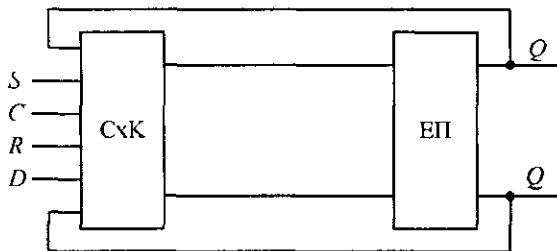


Рис 2.45 Структура тригера у вигляді ЕП і СхК

Стан тригера визначається сигналами на прямому Q та інверсному \bar{Q} виходах При позитивному кодуванні інформації високий рівень напруги на прямому виході відображає значення лог 1 (стан $Q = 1$), а низький рівень — значення лог 0 (стан $Q = 0$)

Зміна стану тригера (його перемикання) забезпечується зовнішніми сигналами й сигналами зворотного зв'язку на виході тригера, які поступають на входи СхК. Звичайно зовнішні сигнали, як і входи тригера, позначають латинськими буквами R, S, T, C, V та іншими. В найпростіших схемах тригерів окрема СхК може бути відсутньою. Оскільки функціональні властивості тригерів визначаються їхньою СхК, то назви основних входів переносяться на всю схему тригера.

2.6.2. Класифікація тригерів

Тригери класифікують за такими ознаками:

- логикою функціонування (RS, JK, D, T та ін.),
- способом записування інформації (асинхронні й синхронні),
- моментом реакції на тактовий сигнал (статичні динамічні),
- кількістю тактів синхронізації (одно-, дво- і тритактові),
- кількістю ступенів (одно- або двоступеневі тригери),
- складом логічних елементів (тригери на елементах НЕ І, НЕ ЧИ, НЕ І ЧИ та ін.)

Відповідно до логіки функціонування розрізняють такі тригери:

- з роздільною установкою станів "0" і "1" (RS -тригери),
- з одним інформаційним входом (D -тригери),
- з лічильним входом (T -тригери),
- універсальні з роздільною установкою станів "0" і "1" (JK -тригери),
- комбіновані (RST -, $RSJK$ -тригери),
- із складною вхідною логикою.

Входи тригерів розділяються на інформаційні (R, S, T та ін.) та керуючі (C, V). Інформаційні (логічні) входи призначені для приймання сигналів інформації, яка запам'ятовується. Назви вхідних сигналів ототожнюють з назвами входів тригера. Керуючі входи служать для керування записуванням інформації. У тригерах може бути два види керуючих сигналів: синхронізуючий (тактовий) сигнал C , який надходить до C -входу (тактового входу) і дозволяючий сигнал V , який надходить до V -входу.

За способом записування (приймання) інформації розрізняють асинхронні й синхронні (тактовні) тригери. Тригери, які не мають C -входу, називаються **асинхронними** (рис. 2.46, а і б). В асинхронних тригерах записування інформації відбувається в будь-який момент часу при надходженні сигналів до інформаційних входів.

Тригери, які мають C -вхід, називаються **синхронними**. У синхронному тригері записування інформації можливе при збігу сигналів на інформаційному й синхронному входах. Цим пояснюється вища стійкість до перешкод синхронних тригерів порівняно з асинхронними.

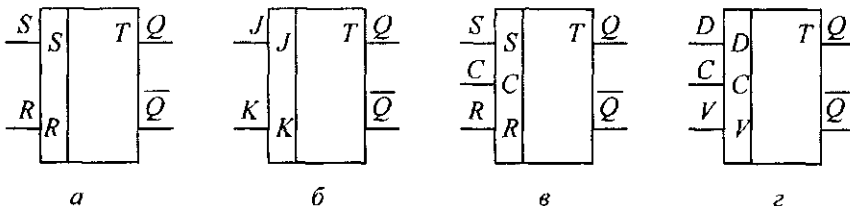


Рис. 2.46 Умовні позначення тригерів. а, б — асинхронних; в, г — синхронних

До V -входів тригера надходять сигнали, які дозволяють ($V = 1$) або забороняють ($V = 0$) записування інформації. У синхронних тригерах з V -входом записування інформації можливе при збігу сигналів на інформаційному, C - і V - виходах (рис. 2.46, з).

Залежно від кількості тактових сигналів, необхідних для формування нового стану, розрізняють одноктактові, двотактові та багатотактові тригери.

За способом керування записуванням (моментом реакції на тактовий сигнал) виділяють синхронні тригери зі статичним (за рівнем), динамічним (за фронтами) та двоступеневим керуванням. В асинхронних тригерах записування нуля і одиниці можливе у будь-який момент часу, при цьому вхідний інформаційний сигнал одночасно є й керуючим. У синхронних тригерах з керуванням за рівнем записування інформації можливе тільки впродовж тривалості тактового сигналу. При цьому тактові сигнали можуть бути прямими (змінюватися від нуля до одиниці) або інверсними (змінюватися від одиниці до нуля) (рис. 2.47, а і б).

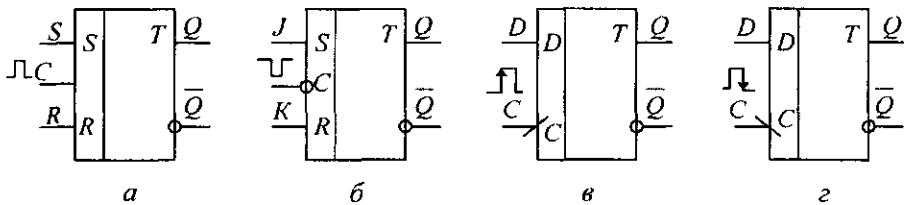


Рис. 2.47. Керуючі входи тригера: а — прямий статичний;

б — інверсний статичний; в — прямий динамічний; г — інверсний динамічний

При керуванні фронтами дозвіл на записування інформації дається тільки в момент перепаду тактового сигналу від нуля до одиниці (прямий динамічний вхід) або від одиниці до нуля (інверсний динамічний вхід). В інші моменти часу тригер не реагує на вхідні інформаційні сигнали незалежно від рівня тактового імпульсу (рис. 2.47, в і г).

2.6.3. Динамічні параметри тригерів

Для тригерів встановлені такі динамічні параметри, які вимірюються на рівні половини амплітуди сигналів (рис. 2.48):

- мінімальна тривалість імпульсу t_C на тактовому вході;
- мінімальний час попереднього установлення сигналу на інформаційному вході $t_{уст}$;
- час відновлення (фіксації) $t_{відн}$ — мінімальний час між початком синхросигналу C і закінченням інформаційного сигналу D ; для асинхронних тригерів $t_{відн}$ — просто тривалість вхідного сигналу;
- час перемикання тригера $t_{п.т}$ — часовий інтервал між фронтом вхідного перемикаючого сигналу та фронтом сигналу на виході Q або \bar{Q} (який пізніше набуває нового положення). Мінімальна тривалість синхроси-

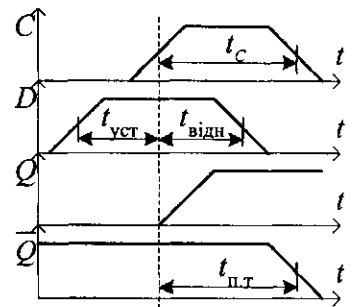


Рис. 2.48. Вимірювання часових параметрів тригерів D-типу

гналу на вході тригера визначається максимальним часом перемикання тригера $t_C \geq t_{П.Т. \max}$. У двоступеновому тригері з однократною синхронізацією час перемикання другого ступеня визначається відносно спаду синхроімпульсу.

2.6.4. Таблиця переходів і логічні рівняння RS-тригера

RS-тригером називають запам'ятовуючий елемент з роздільними інформаційними входами для установлення його в стан "0" (R -вхід) і в стан "1" (S -вхід). Назва "RS-тригер" утворена від перших літер слів RESET (скинення) і SET (установлення).

У таблиці переходів RS-тригера (табл. 2.8) прийняті позначення: R_i, S_i, Q_i — значення логічних змінних у момент часу t на входах R, S і виході Q ; Q_{i+1} — стан тригера після перемикання; K_6, K_7 — невизначені коефіцієнти на тих наборах, де вхідні сигнали R_i і S_i одночасно набувають значення одиниці (заборонена комбінація сигналів).

Таблиця 2.8

R_i	S_i	Q_i	Q_{i+1}
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	K_6
1	1	1	K_7

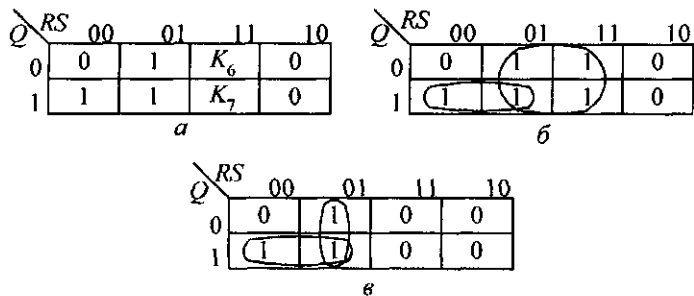


Рис. 2.49. Карта Карно для RS-тригерів

Таблиці переходів відповідає карта Карно (рис. 2.49, а), де значення функції Q_{i+1} для мінтермів R_i, S_i, Q_i і $R_i, S_i, \overline{Q_i}$ замінені невизначеними коефіцієнтами K_6 і K_7 . Припустивши, що комбінації вхідних сигналів $R_i S_i = 1$ не існує, отримаємо карти Карно для $K_6 = K_7 = 1$ (рис. 2.49, б) і $K_6 = K_7 = 0$ (рис. 2.49, в).

Із карт Карно отримуємо логічні рівняння асинхронного RS-тригера:

$$K_6 = K_7 = 1, \quad Q_{i+1} = S_i \vee \overline{R_i} Q_i, \quad (2.1)$$

$$K_6 = K_7 = 0, \quad Q_{i+1} = \overline{R_i} (S_i \vee Q_i). \quad (2.2)$$

Логічні вирази (2.1) і (2.2) визначають новий стан тригера Q_{i+1} залежно від старого стану Q_i та вхідних сигналів R_i і S_i . В подальшому для спрощення індекс i у правій частині логічного виразу опускається.

Асинхронний RS-тригер на елементах НЕ І

Перетворимо логічний вираз (2.1) до виду, зручного для реалізації на елементах НЕ І:

$$\overline{Q_{i+1}} = \overline{S \vee \overline{R} \cdot Q} = \overline{\overline{S} \cdot R \cdot Q}. \quad (2.3)$$

Схема асинхронного RS-тригера на двох елементах НЕ І з логічними зв'язками на основі виразу (2.3) показана на рис. 2.50, а. Особливістю цього тригера є інверсне керування за інформаційними входами, що відображається в умовному графічному позначенні. Із аналізу діаграм роботи RS-тригера випливає, що елементи НЕ І в схе-

мі перемикаються послідовно \bar{S} інтервал часу, коли на обох виходах встановлюються однакові сигнали $Q = 1$ і $\bar{Q} = 1$ (рис 2 50 в, заштриховані області) — явище "ризик"

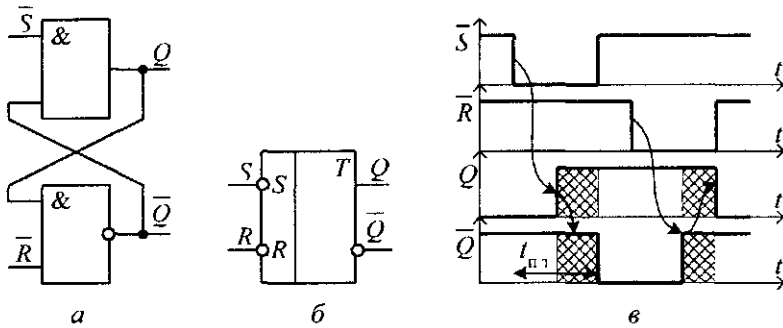


Рис 2 50 Асинхронний RS-тригер на елементах НЕ 1
а — схема, б — умовне позначення, в — часові діаграми

Тривалість перемикання тригера визначається сумою затримок $t_{ПТ} = 2t_p$. Тривалість вхідного сигналу визначається з умови $t_i \geq t_{ПТ}$. На практиці для надійності перемикання тригера тривалість вхідного імпульсу збільшують на одну затримку тобто $t_i = 3t_p$. Максимальна і робоча частоти перемикання тригера відповідно дорівнюють $f_{max} = 1/(2t_p)$ і $f_p = 1/(3t_p)$.

Асинхронний RS-тригер на елементах НЕ ЧИ

Перетворимо логчний вираз (2 2) до вигляду, зручного для реалізації на елементах НЕ ЧИ

$$Q_{i+1} = \overline{\overline{R(S \vee Q)}} = \overline{R \vee (S \vee Q)} \quad (2 4)$$

Схема асинхронного RS-тригера на двох елементах НЕ ЧИ з логчними зв'язками на основі виразу (2 4) показана на рис 2 51, а

Із аналізу діаграм роботи RS-тригера випливає, що елементи НЕ ЧИ в схемі перемикаються послідовно \bar{S} інтервал часу, коли на обох виходах устанавлюються однакові сигнали $Q = 0$ і $\bar{Q} = 0$, — явище "ризик" (рис 2 51, в). Часові параметри даного тригера аналогічні параметрам тригера, зображеного на схемі рис 2 50, а

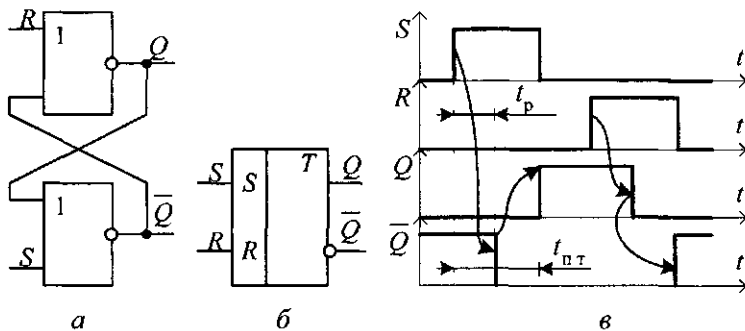


Рис 2 51 Асинхронний RS-тригер на елементах НЕ ЧИ
а — схема, б — умовне позначення, в — часові діаграми

Синхронний RS-тригер на елементах НЕ І

Для побудови синхронного RS-тригера на елементах НЕ І треба замінити в логічному виразі (2.3) змінні S і R на сполучення CS і CR , де C – синхросигнал:

$$Q_{t+1} = \overline{\overline{CS} \overline{CR} Q}. \quad (2.5)$$

Схема синхронного RS-тригера на чотирьох елементах НЕ І з логічними зв'язками на основі виразу (2.5) показана на рис. 2.52, а. Елементи $D1$ і $D2$ складають схему керування з прямими входами, а елементи $D3$ і $D4$ утворюють фіксатор (асинхронний RS-тригер).

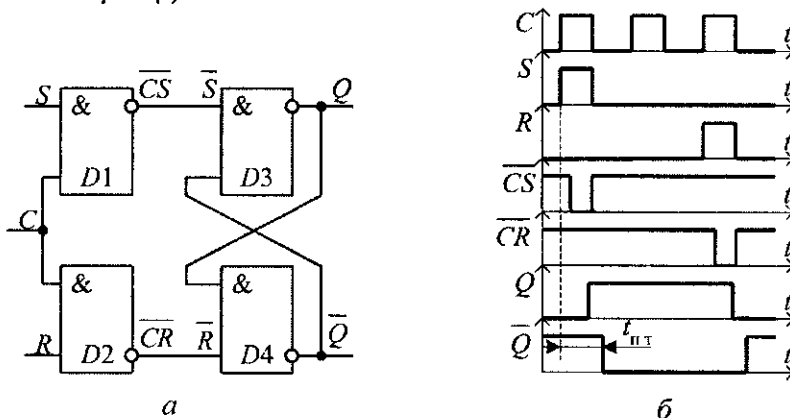


Рис. 2.52. Синхронний RS-тригер на елементах НЕ І:

а — схема; б — часові діаграми

При значенні сигналів $CS = 1$ на виході елемента $D1$ встановлюється лог. 0 і тригер перемикається в стан "1".

При значенні сигналів $CR = 1$ на виході елемента $D2$ встановлюється лог. 0 і тригер перемикається в стан "0". Комбінація входних сигналів $CSR = 1$ заборонена, оскільки призводить до невизначеного стану тригера.

Із часової діаграми (рис. 2.52, б) випливає, що час перемикання тригера $t_{пт} = 3t_p$, а тривалість синхросигналу (з урахуванням запасу на одну затримку) визначається з умови $t_c = 4t_p$. Максимальна і робоча частоти перемикання тригера відповідно дорівнюють: $f_{max} = 1/3t_p$ і $f_p = 1/4t_p$.

Синхронний RS-тригер на елементах НЕ ЧИ

Для побудови синхронного RS-тригера на елементах НЕ ЧИ належить замінити в логічному виразі (2.4) змінні S і R на сполучення \overline{CS} і \overline{CR} :

$$Q_{t+1} = \overline{\overline{CR} \vee (\overline{CS} \vee Q)} = \overline{C} \vee R \vee (C \vee S \vee Q). \quad (2.6)$$

Схема синхронного RS-тригера на чотирьох елементах НЕ ЧИ з логічними зв'язками на основі виразу (2.6) показана на рис. 2.53. Елементи $D1$ і $D2$ складають схему керування з інверсними входами, а елементи $D3$ і $D4$ утворюють фіксатор (асинхронний RS-тригер).

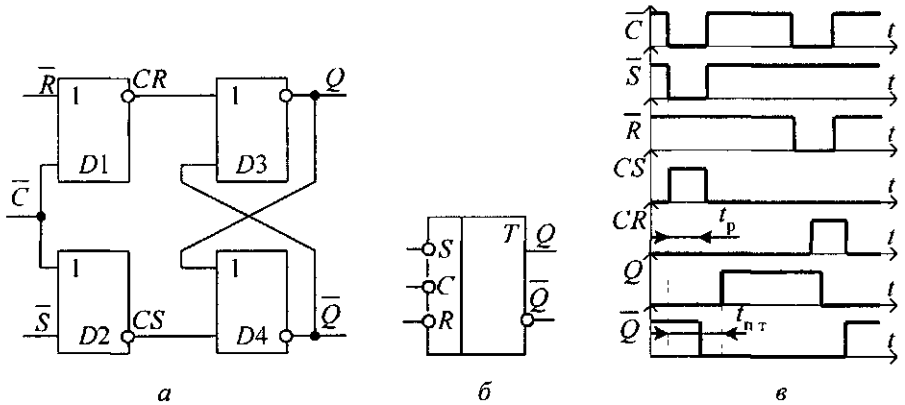


Рис. 2.53. Синхронний RS-тригер на елементах НЕ ЧИ:

а — схема; б — умовне позначення; в — часові діаграми

При значенні сигналів $\bar{C} = 0$ і $\bar{S} = 0$ на виході елемента $D2$ встановлюється лог 1 (тобто $CS = 1$) і тригер переключається в стан "1". При значенні сигналів $\bar{C} = 0$ і $\bar{R} = 0$ на виході елемента $D1$ встановлюється лог 1 (тобто $CR = 1$) і тригер переключається в стан "0". Комбінація сигналів $\bar{C} = \bar{S} = \bar{R} = 0$ заборонена, тому що призводить до невизначеного стану тригера. Часові параметри такого тригера аналогічні параметрам тригера, зображеного на схемі рис. 2.52

Двоступеневі RS-тригери

Двоступеневі тригери будують за способом "M-S" і забезпечують поєднання двох процесів — одночасного записування нової інформації та зчитування старої. Під час дії синхроімпульсу C перший ступінь "M" (Master — основний) приймає нову вхідну інформацію, а другий ступінь "S" (Slave — допоміжний) в цей же час передає у зовнішні схеми стару інформацію. Після закінчення синхроімпульсу C інформація з першого ступеня переписується у другий ступінь.

При однофазному (однотактному) обміні інформацією зв'язок між ступенями реалізується за допомогою інвертора (рис. 2.54, а),

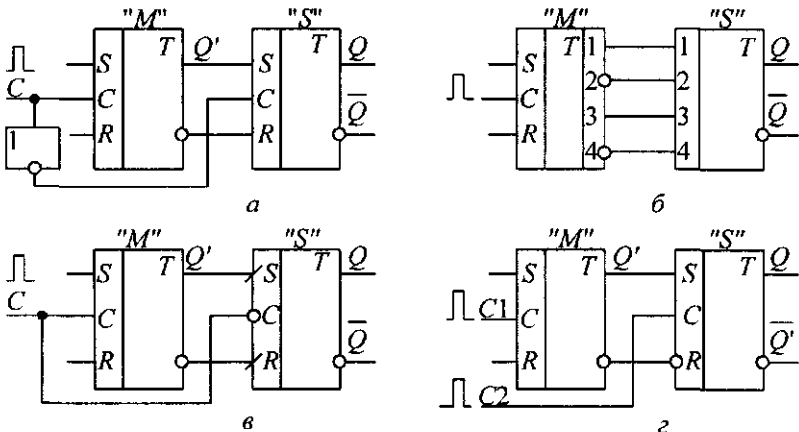


Рис. 2.54. Організація зв'язку між ступенями тригера: а — з інвертором;

б — із забороняючими зв'язками; в — з різнополярним керуванням; г — з двофазним обміном

забороняючих зв'язків (рис. 2.54, б) або різнополярного керування (рис. 2.54, в). При двотактному обміні зв'язок між ступенями забезпечується двома серіями синхросигналів — $C1$ і $C2$ (рис. 2.54, з).

2.7. Тригери типів JK, T, D та DV

2.7.1. Тригери типу JK

Тригером типу JK називається запам'ятовуючий елемент з двома стійкими станами та інформаційними входами J (аналог S) і K (аналог R), які забезпечують відповідно роздільну установку станів "1" і "0". Він функціонує подібно до RS-тригера, але при збігу сигналів $JK = 1$ переключається в протилежний стан, тобто реалізує додавання сигналів за модулем два. Таким чином, JK-тригер не має заборонених комбінацій вхідних сигналів. Тригер типу JK є універсальним, оскільки може виконувати функції RS-тригера (при роздільному надходженні сигналів J і K), T-тригера (при одночасній подачі сигналів J і K), D-тригера (при подачі сигналу від входу J через інвертор на вхід K). Зміна станів JK-тригера наведена в табл. 2.9; за допомогою карти Карно (рис. 2.55) отримуємо наступне рівняння для тригера:

$$Q_{i+1} = \overline{K}_i \cdot Q_i \vee J_i \cdot \overline{Q}_i. \quad (2.7)$$

Таблиця 2.9

K_i	J_i	Q_i	Q_{i+1}
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Q	KJ			
	00	01	11	10
0	0	1	1	0
1	1	1	0	0

Рис. 2.55. Карта Карно для JK-тригера

Для побудови одноступеневого синхронного JK-тригера на елементах НЕ і потрібно замінити в рівнянні (2.7) змінні K і J на сполучення \overline{CK} і \overline{JK} , після чого виконати перетворення на основі правил подвійної інверсії та правил де Моргана:

$$Q_{i+1} = \overline{\overline{C \cdot K} \cdot \overline{Q}} \vee \overline{\overline{C \cdot J} \cdot \overline{Q}} = \overline{\overline{C \cdot K} \cdot \overline{Q} \cdot \overline{C \cdot J} \cdot \overline{Q}}. \quad (2.8)$$

Схема одноступеневого JK-тригера з логічними зв'язками на основі рівняння (2.8) показана на рис. 2.56.

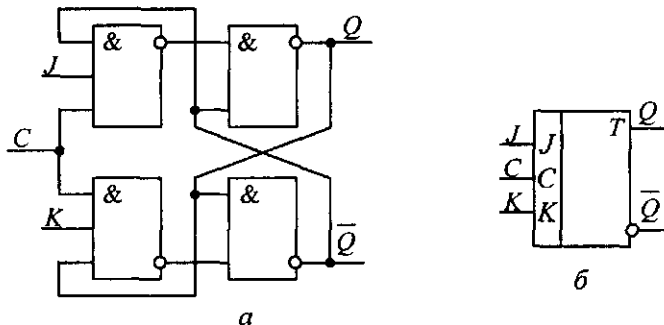


Рис. 2.56. Одноступеневий JK-тригер: а — схема; б — умовне позначення

У схемотехніці потенціальних тригерів в основному застосовують двоступеневі синхронні *JK*-тригери на елементах НЕ І (рис. 2.57). Нова інформація знімається з виходів Q^* основного *M*-ступеня, а стара — з виходів Q допоміжного *S*-ступеня.

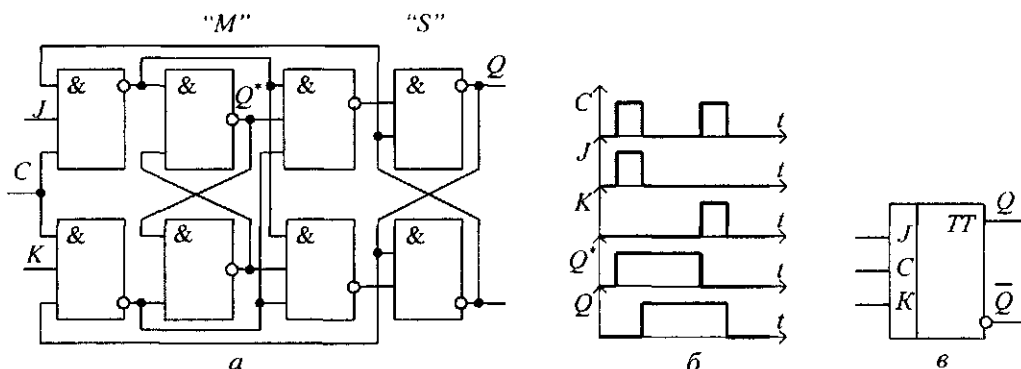


Рис. 2.57. Двоступеневий *JK*-тригер на елементах НЕ І:

а — схема; б — часові діаграми; в — умовне графічне позначення

Із часових діаграм (рис. 2.57, б) виходить: при застосуванні *JK*-тригера для зберігання інформації сигнали на входи *J* і *K* подаються по чергово; при збігові сигналів на входах *J* і *K* реалізується лічильний тригер (на практиці ці входи часто монтажно об'єднуються). Час перемикання *JK*-тригера визначається сумою затримок першого і другого ступенів і дорівнює $t_{птт} = 7t_p$. На виході *S*-ступеня нова інформація з'являється після спаду сигналу *C*.

2.7.2. Таблиця переходів і логічні рівняння *T*-тригера

Тригером типу *T* називається запам'ятовуючий елемент з двома стійкими станами та одним інформаційним *T*-входом. Стан *T*-тригера змінюється на протилежний після кожного надходження лічильного сигналу на *T*-вхід. Логіка функціонування асинхронного лічильного тригера подана таблицею переходів (табл. 2.10) і описується логічним рівнянням

$$Q_{t+1} = \overline{T}_t \cdot Q_t \vee T_t \cdot \overline{Q}_t \tag{2.9}$$

Для побудови асинхронного *RS*-тригера на елементах НЕ І рівняння (2.9) перетворюється до вигляду, зручного для реалізації у заданому елементному базисі:

$$Q_{t+1} = \overline{\overline{T} \cdot Q} \vee \overline{\overline{T} \cdot \overline{Q}} = \overline{\overline{T} \cdot Q} \cdot \overline{\overline{T} \cdot \overline{Q}} \tag{2.10}$$

У рівнянні (2.10) для виключення інверсії сигналу *T* використовується тотожність $\overline{T \cdot Q} = (\overline{T \cdot Q}) \cdot Q$. Перемикання тригера визначається сумісною дією лічильних сигналів "*T*" і зворотного зв'язку виходів Q і \overline{Q} . Для виключення

"гонок" у тригері сигнали зворотного зв'язку не повинні змінюватися під час дії лічильного сигналу "*T*". Затримка вихідного сигналу може здійснюватися лінією затримки (в імпульсно-потенціальній системі елементів) чи додатковим тригером (в потенціальній системі елементів). Схема одноступеневого асинхронного *T*-тригера на елементах НЕ І з логічними зв'язками відповідно до рівняння (2.10) показана на рис. 2.58. Сигнали з ви-

Таблиця 2.10

T_t	Q_t	Q_{t+1}
0	0	0
0	1	1
1	0	1
1	1	0

ходів елементів $D1$ і $D2$ затримуються на час Δt , що дорівнює тривалості лічильного сигналу на T -вході

Надійні структури лічильних тригерів будують з використанням двоступеневих тригерів або тригерів з динамічним керуванням по фронту сигналу C

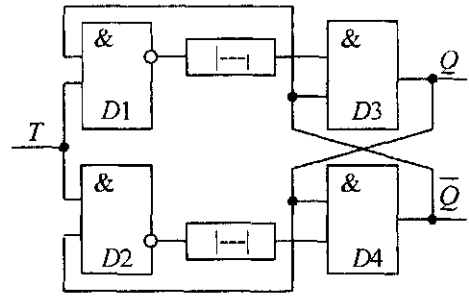


Рис. 2.58. Схема одноступеневого T -тригера

2.7.3. Двоступеневі T -тригери

Схема двоступеневого асинхронного T -тригера на елементах НЕ І з логічними зв'язками згідно з рівнянням (2.10) показана на рис. 2.59, а. Асинхронний T -тригер вміщує два синхронних RS -тригери, при цьому на T -вхід основного ступеня подається сигнал, який рахується, а входи S і R з'єднані відповідно з виходами \bar{Q} і Q допоміжного ступеня

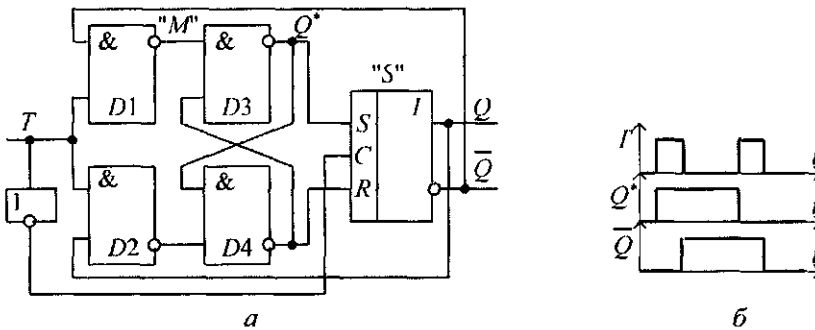


Рис. 2.59. Асинхронний двоступеневий T -тригер. а — схема; б — часові діаграми роботи

Нехай у початковому стані $Q^* = Q = 0$. При надходженні першого лічильного імпульсу відбувається збіг лог 1 на вході елемента $D1$ ($T = 1$, $\bar{Q} = 1$), а основний тригер перемикається у стан "1". Після закінчення вхідного імпульсу інвертор у колі зв'язку між ступенями дозволяє перезаписування інформації у допоміжний тригер (рис. 2.59, б). Після надходження другого імпульсу тригер перемикається у стан "0", тобто здійснюється додавання вхідних сигналів за модулем два. У загальному випадку в T -тригерах з однофазним обміном інформацією зв'язок між ступенями реалізується на основі трьох основних способів (див. рис. 2.54). У схемі двоступеневого синхронного T -тригера перемикавання "M"-ступеня відбувається при збігові імпульсу, який рахується, із синхросигналом (рис. 2.60, а). У схемі двоступеневого синхронного TV -тригера перемикавання відбувається при збігові імпульсу, який рахується, з синхросигналом за умови $V = 1$ (рис. 2.60, б).

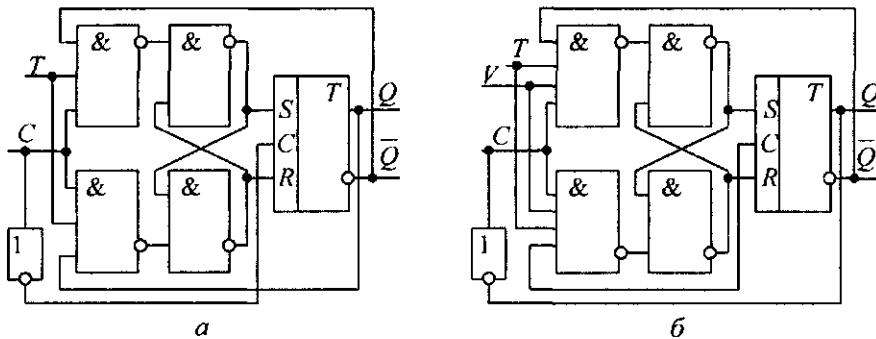


Рис. 2.60. Синхронні двоступеневі лічильні тригери: а — типу Т; б — типу TV

2.7.4. Таблиця переходів і логічне рівняння D-тригера

Тригером типу *D* називається синхронний запам'ятовуючий елемент з двома стійкими станами і одним інформаційним *D*-входом. Закон функціонування *D*-тригера описується логічним рівнянням:

$$Q_{i+1} = C_i D_i.$$

Це рівняння показує, що після перемикавання стан *D*-тригера повторює значення сигналу на *D*-вході в тактові моменти часу. Тому в літературі *D*-тригери часто називають тригерами затримки (від *Delay* — затримка).

Схему *D*-тригера можна побудувати на основі синхронного *RS*-тригера, якщо сигнал по входу *S* одночасно подавати через інвертор на вхід *R* (рис. 2.61, а). Схеми *D*-тригера будують також на основі самостійного логічного рівняння. Перетворимо рівняння (2.5) замінивши сигнал *S* на *D* і сигнал *R* на \bar{D} :

$$Q_{i+1} = \overline{C \cdot S \cdot C \cdot R} \cdot Q = \overline{C \cdot D \cdot C \cdot \bar{D}} \cdot Q. \quad (2.11)$$

Схема *D*-тригера на елементах НЕ І з логічними зв'язками згідно з рівнянням (2.11) показана на рис. 2.61, б.

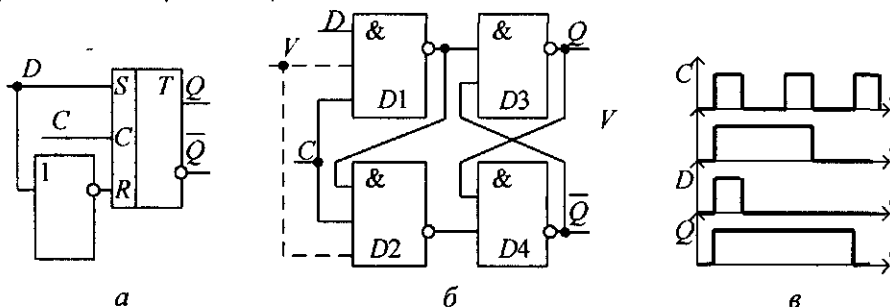


Рис. 2.61. *D*-тригер: а — на основі *RS*-тригера; б — на елементах НЕ І; в — часові діаграми роботи

D-тригер "слідкує" за зміною сигналу на *D*-вході під час дії синхросигналу *C* і зберігає ту інформацію, яка була в момент його закінчення. *RS*-тригери такої властивості не мають і тому вони менше завадостійкі порівняно з *D*-тригерами.

Для затримки інформації в *D*-тригері на довільне число тактів використовується дозволяючий *V*-вхід, як показано штриховою лінією на рис. 2.61, б. Якщо $V = 1$, то

DV-тригер функціонує як звичайний тригер затримки; якщо $V = 0$, то робота схеми за входами блокується і DV-тригер зберігає попередню інформацію.

Схема двоступеневого одноканального DV-тригера на елементах НЕ І із заборонячими зв'язками між ступенями показана на рис. 2.62.

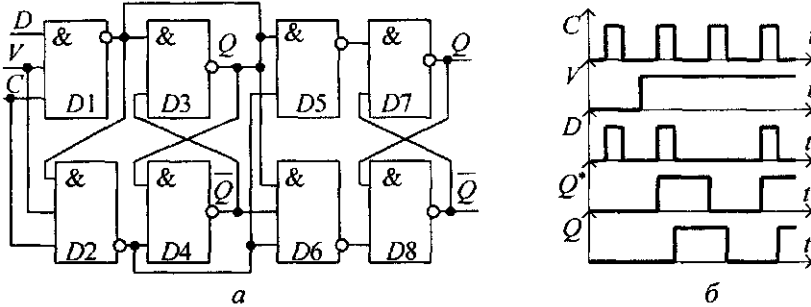


Рис. 2.62. Двоступеневий DV-тригер: а — схема; б — часові діаграми

2.7.5. D-тригер з динамічним керуванням

В усіх синхронних тригерах із статичним керуванням (за рівнем) можливе фальшиве перемикання у випадку зміни сигналів на інформаційних входах під час дії синхросигналу C . Наприклад, якщо в D-тригері інформаційний сигнал змінює своє значення від одиниці до нуля до закінчення сигналу "С", то тригер може знову переключатися з одиничного в нульовий стан. Тому для надійної роботи D-тригера потрібний певний інтервал часу між фронтом синхроімпульсу C і спадом сигналу на D-вході (параметр $t_{\text{вдн}}$).

У тригерах з динамічним керуванням записуванням інформації синхроімпульс C активний лише на короткому інтервалі часу в області фронту чи спаду. Тому D-тригери з динамічним керуванням мають високу завадостійкість. На практиці широке розповсюдження отримали D-тригери з прямим динамічним керуванням за схемою "трьох тригерів". Схема такого D-тригера з прямим динамічним керуванням показана на рис. 2.63, а, а його умовне графічне зображення — на рис. 2.63, б. Тут зберігання інформації здійснює основний вихідний синхронний RS-тригер (елементи $D5$ і $D6$) з інверсним керуванням, а приймання тактового і інформаційного сигналів і задання динамічного режиму роботи забезпечують два вихідних комутуючих тригери (елементи $D1$, $D2$, $D3$, $D4$). Елемент $D4$ подає інверсне значення вхідного сигналу D на входи елементів $D1$ і $D3$ (рис. 2.63, в). Елемент $D1$ повторює значення сигналу D .

При $C = D = 1$ вмикається елемент $D2$ і встановлює основний тригер за входом S_1 в стан "1"; одночасно блокується робота елемента $D3$, в зв'язку з чим схема уже не реагує на зміну вхідного сигналу. При $C = 1$, $D = 0$ вмикається елемент $D3$ і встановлює основний тригер в стан "0"; одночасно відбувається приймання нової інформації елементом $D4$. При $C = 0$ забезпечується режим зберігання записаної інформації.

Час перемикання тригера за C-входом $t_{\text{пт}} = 3t_p$. Після закінчення сигналів на D- і C-входах починається етап відновлення, який характеризується переходом комутуючих тригерів у початковий стан за час $3t_p$. Максимальна частота перемикання D-тригера з динамічним керуванням визначається сумарним часом затримок $f_{\text{max}} = 1/(6t_p)$.

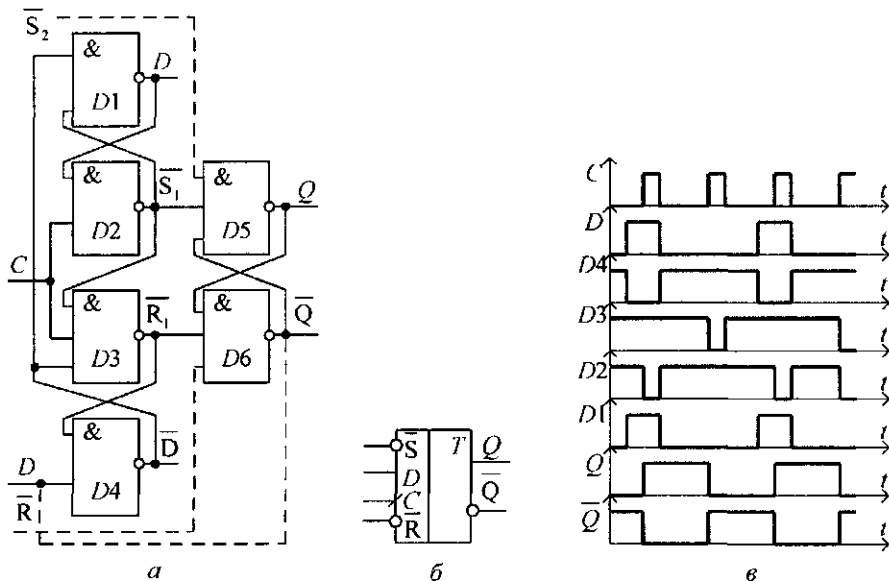


Рис. 2.63. D-тригер з динамічним керуванням:
а – схема; б – умовне позначення; в – часові діаграми

На рис. 2.64 показані схеми ряду тригерів серії КР1533: TP2 — чотири RS-тригери; TB9 — два JK-тригери; TB11 — два JK-тригери із загальним входом скидання і синхронізації за спадом C; TM2 — два D-тригери з динамічним керуванням за фронтом C.

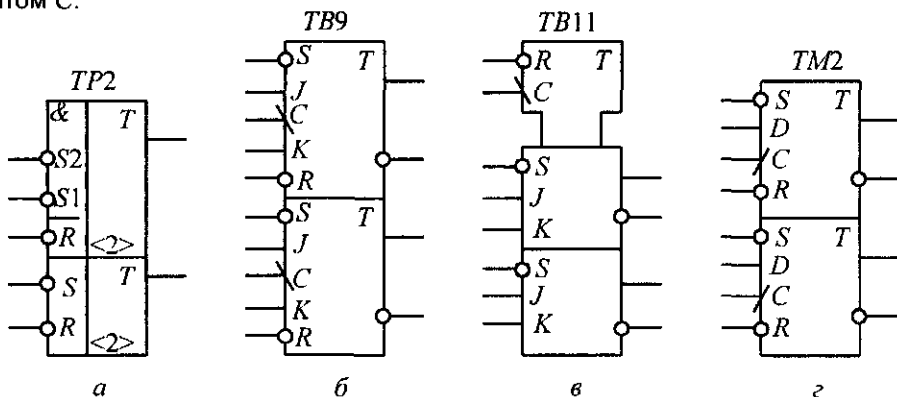


Рис. 2.64. Тригери серії КР-1533: а — TP2; б — TB9; в — TB11; г — TM2

Мікросхеми вказаних тригерів характеризуються наступними параметрами: $U_{CC} = 5 \text{ В}$; $I_{CC} = 4 \dots 5 \text{ мА}$; $P_{CC} = 20 \dots 25 \text{ мВт}$; час перемикання — $15 \dots 20 \text{ нс}$.

Схема імпульсного (динамічного) RS-тригера на елементах ЧИ I та лінії затримки (ЛЗ), реалізована на основі рівняння $Q_{i+1} = \overline{R_i}(S_i \vee Q_i)C_i$, показана на рис. 2.65, а.

Імпульсний тригер встановлюється в стан лог. 1 сигналами $S_i = 1$ і $R_i = 0$. Після цього в замкненому колі схеми тригера циркулюють імпульси з періодом синхронізації T (рис. 2.65, б). Для скидання тригера подається сигнал $R_i = 1$, після чого схема ЗАБОРОНА перериває циркуляцію імпульсів

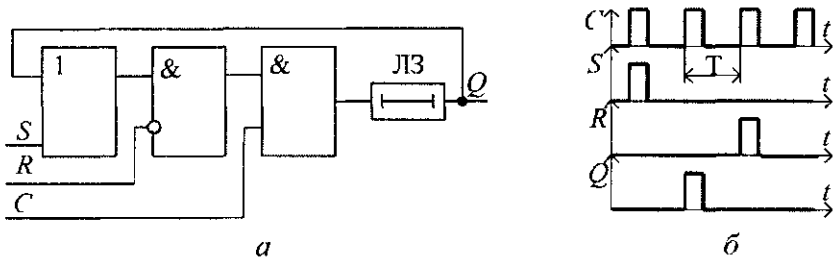


Рис. 2.65. Імпульсний RS-тригер: а — схема; б — часові діаграми роботи

Схема RST-тригера в потенціально-імпульсній системі елементів показана на рис. 2.66.

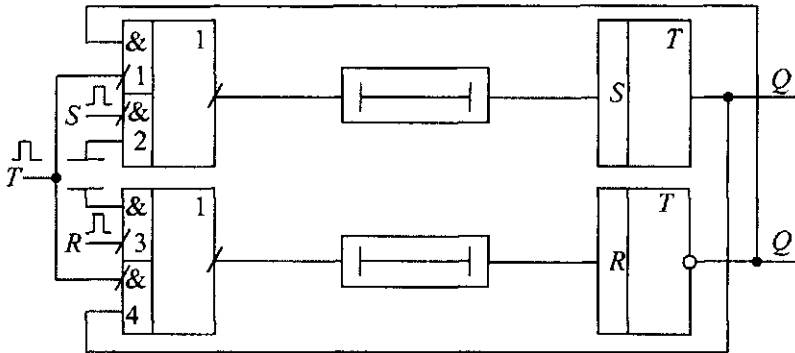


Рис. 2.66. Схема RST-тригера

На потенціальні входи вентилів 1 і 4 подаються сигнали зворотного зв'язку з інверсного \bar{Q} і прямого Q виходів статичного тригера, а імпульсні входи об'єднуються і утворюють T -вхід. Імпульсні входи вентилів 2 і 3 утворюють S - і R - входи.

Схема магнітного RS-тригера, в якому струм I_S переключас осердя в стан "1", а струм I_R — в стан "0", показана на рис. 2.67, а. При зчитуванні стану "1" на виході виникає позитивний імпульс (рис. 2.67, б). Одночасна подача струмів I_S і I_R заборонена.

Функція переходів RS-тригера описується рівнянням

$$Q_{t+1} = S_t \vee \bar{R}_t \cdot Q_t,$$

де індекси t і $t+1$ визначають стани входів до і після переключення. Магнітний RS-тригер може реалізувати функцію кон'юнкції виду:

$$Y_{t+1} = S_t R_{t+1},$$

де змінні S_t і R_t подаються відповідно в першому і другому такті.

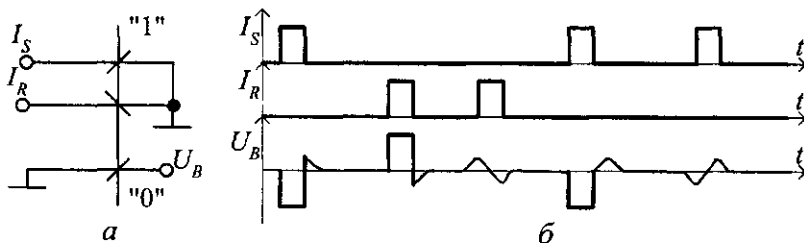


Рис. 2.67. RS-тригер: а — умовне позначення; б — часові діаграми роботи

Збільшенням числа обмоток за входами S і R логічна функція схеми розширюється. Узгоджене вмикання декількох обмоток за входами S і R дає функцію

$$Y_{t+1} = (\bigvee_{i=1}^n S_i)_t (\bigwedge_{j=1}^m R_j)_{t+1}.$$

Наприклад, для $n = 2$, $m = 2$ отримуємо:

$$Y_{t+1} = (S_1 \vee S_2)_t \cdot (R_1 \vee R_2)_{t+1}.$$

Якщо припустити, що в першому такті $S_t = 1$, а в другому такті по декількох обмотках R_{t+1} , ввімкнених узгоджено, надходить m аргументів X_i , а сигнал на виході схеми знімати в третьому додатковому такті R_{t+2} , то реалізується функція НЕ ЧИ (рис. 2.68, а).

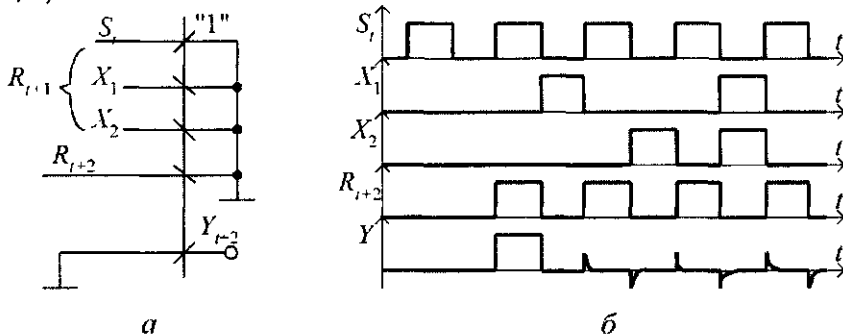


Рис. 2.68. Магнітний елемент НЕ ЧИ: а — схема; б — часові діаграми роботи

Наприклад, для $m = 3$ отримуємо:

$$Y_{t+2} = \bigvee_{i=1}^m X_i; \quad Y_{t+2} = \overline{X_1 \vee X_2 \vee X_3} = \overline{X_1} \cdot \overline{X_2} \cdot \overline{X_3}.$$

Контрольні запитання

1. Поясніть роботу дюдних логічних елементів ЧИ, І.
2. Охарактеризуйте властивості логічних елементів ТТЛШ.
3. За рахунок яких факторів досягається надшвидкодія логічних елементів ЕЗЛ?
4. Що таке кріотрон?
5. Укажіть на відмінності між тригерами типів RS , JK , D і T .
6. Коли використовують двоступеневі тригери?
7. Покажіть достоїнства елементів на КМОН-структурах.
8. В чому полягають переваги синхронних тригерів порівняно з асинхронними.
9. Поясніть, чому JK -тригер називають універсальним.
10. Поясніть призначення логічних елементів з відкритим колектором.
11. Охарактеризуйте роботу логічних елементів з трьома станами.
12. Які переваги мають логічні елементи на основі арсеніду галію порівняно з кремнієвими схемами?
13. Поясніть роботу елемента ЧИ на магнітних схемах.
14. Охарактеризуйте особливості побудови потенціальної, імпульсної та потенціально-імпульсної систем елементів.

Розділ 3

Послідовнісні (накопичувальні) вузли комп'ютерної схемотехніки

3.1. Регістри

3.1.1. Загальна характеристика регістрів

Регістром називається типовий функціональний вузол комп'ютера, призначений для приймання, тимчасового зберігання, перетворення і видачі n -розрядного двійкового слова. Регістр містить регулярний набір однотипних тригерів, в кожному з яких зберігається значення одного двійкового розряду машинного слова. Найчастіше використовують тригери типів RS , JK і D (рис. 3.1).

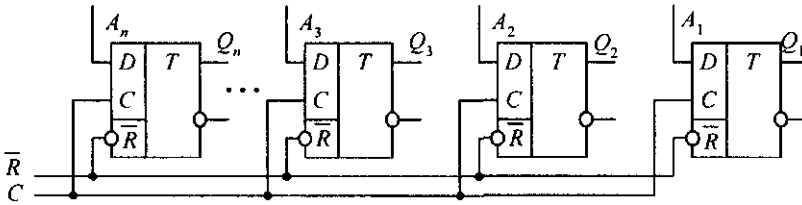


Рис. 3.1. Схема регістра на D -тригерах

Регістри, призначені тільки для приймання (записування), зберігання і передачі інформації, називаються **елементарними** або **фіксаторами**. Регістри, в яких зберігання даних поєднується з мікроопераціями зсуву, називаються **зсувовими**. Елементарні регістри будують на одноступеневих тригерах, а зсувові — на двоступеневих або D -тригерах з динамічним керуванням. Логічна функція регістра позначається буквами RG (*register*). Регістри забезпечують зберігання команд, адреси пам'яті, результатів операцій, індексів та ін.

Регістри класифікують за такими ознаками:

- способом керування записуванням — асинхронні та синхронні;
- способом записування і видачі двійкових слів — паралельні, послідовні й універсальні; у паралельних регістрах записування і видача слів виконується одночасно всіма розрядами, а в послідовних — розряд за розрядом в напрямку від молодших розрядів до старших або навпаки; універсальні регістри забезпечують як паралельний, так і послідовний обмін інформацією;
- числом ліній для представлення значення одного розряду слова (біта інформації) — однофазні й парафазні; при однофазному поданні значення кожного розряду слова передається по одній лінії зв'язку, а при парафазному — по двох лініях (одночасно відображається пряме та інверсне значення розряду);
- числом тактів для записування слова — одно-, дво- і багатотактні;
- складом мікрооперацій, які виконуються — установлювальні, записування,

читання, порозрядні логічні й зсуву, а також перетворення послідовного коду в паралельний і навпаки;

- напрямом зсуву — односторонні (лівий або правий зсув) і двосторонні (реверсивні);
- типом тригерів, що використовуються;
- елементною структурою — потенціальні, імпульсні й потенціально-імпульсні.

3.1.2. Установлювальні мікрооперації. Однофазний і парафазний спосіб записування інформації

Установлювальні мікрооперації служать для переключення регістрів у певний стан. Наприклад, установлення регістра в стан "0" або "1"; установлення парних розрядів у стан "0", а непарних — у стан "1"; записування в регістр деякої константи або обнуління деяких байтів та ін. Установлювальні мікрооперації переважно використовують асинхронні входи регістра, наприклад, вхід скидання \overline{R} для схеми, зображеної на рис. 3.1.

У регістрах на *RS*- або *JK*- тригерах можливий однофазний або парафазний спосіб записування інформації. При однофазному записуванні значення кожного розряду слова $A = A_n A_{n-1} \dots A_i \dots A_1$ надходить по одній лінії зв'язку на вхід *S* (або *J*) відповідних тригерів. Після зчитування записаної інформації регістр має обнулятися по спільному *R* входу (рис. 3.2). Таким чином, при однофазному записуванні частота обміну інформацією зменшується, оскільки процеси введення і скидання чергуються.

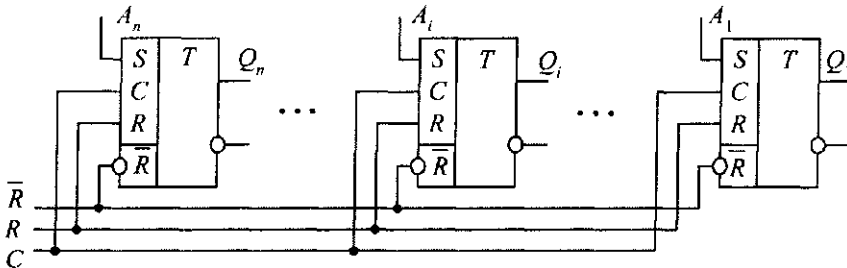


Рис. 3.2. Схема регістра з однофазним записом даних

При парафазному записуванні інформації значення кожного розряду слова *A* передається по двох лініях зв'язку. При цьому пряме значення A_i надходить на вхід *S* (або *J*) відповідних тригерів, а інверсне значення \overline{A}_i — на вхід *R* (або *K*). У цьому випадку не потрібне попереднє скидання регістра в стан "0", тому що таку функцію виконує сигнал \overline{A}_i (рис. 3.3).

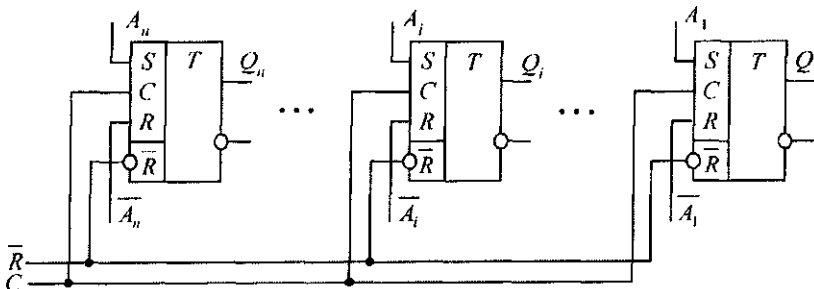


Рис. 3.3. Схема регістра з парафазним записом даних

У регістрах на D -тригерах, які мають один інформаційний вхід, можливий тільки однофазний спосіб записування інформації. З приходом чергового синхроімпульсу записується нова інформація або при відсутності даних на вхідній шині стан регістра автоматично обнуляється (див. рис. 3.1).

При необхідності збереження інформації на декілька тактів у регістрах на D -тригерах потрібно використати дозволяючий V -вхід, або блокувати проходження синхроімпульсу на C -вхід.

3.1.3. Записування інформації від двох джерел

Для записування інформації від декількох джерел (напрямків) на вході кожного тригера ставлять додаткові комбінаційні схеми, які створюють вхідну логіку регістра. Кожний напрямок має свою сукупність електричних ліній (**шину**), по яких передаються сигнали, що відображають значення розрядів слова. Якщо n -розрядне слово передається однофазним кодом, то шина має n ліній зв'язку, а якщо парафазним кодом — то $2n$ ліній. Записування кожного слова ініціюється відповідним керуючим сигналом Y_1, Y_2 та ін.

Для записування в регістр на RS -тригерах однофазним кодом слів A і B потрібно реалізувати такі порозрядні функції збудження входу S_i :

$$S_i = Y_1 A_i \vee Y_2 B_i, \quad (3.1)$$

де A_i, B_i — двійкові розряди слів A і B ; Y_1, Y_2 — сигнали керування приймання слів A і B відповідно. Схема вхідної логіки i -го розряду регістра на основі рівняння (3.1) показана на рис. 3.4, а.

Для записування в регістр на JK -тригерах парафазним кодом слів A і B потрібно реалізувати такі порозрядні функції збудження входів J_i і K_i :

$$J_i = Y_1 A_i \vee Y_2 B_i; \quad K_i = Y_1 \bar{A}_i \vee Y_2 \bar{B}_i. \quad (3.2)$$

Схема вхідної логіки i -го розряду регістра на основі рівнянь (3.2) показана на рис. 3.4, б.

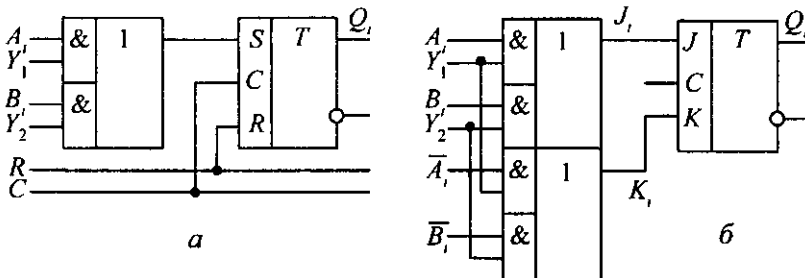


Рис. 3.4. Схеми розряду регістра із записом слів від двох джерел:
а — однофазним кодом; б — парафазним кодом

3.1.4. Зчитування інформації

Інформація, яка зберігається в регістрах, може передаватися у зовнішні схеми однофазним або парафазним способом у прямому або оберненому коді. Для реалізації мікрооперацій зчитування до виходів кожного тригера підключаються комбінаційні схеми, які створюють вихідну логіку регістра.

Схеми вихідної логіки будуються на основі таких порозрядних логічних рівнянь:

- для зчитування однофазним прямим або оберненим кодом

$$\underline{Ш}_i = Y_{\text{пр}} Q_i \vee Y_{\text{об}} \overline{Q}_i; \quad (3.3)$$

- для зчитування парафазним прямим або оберненим кодом

$$\underline{Ш}_i^* = Y_{\text{пр}} Q_i \vee Y_{\text{пр}} \overline{Q}_i; \quad \overline{\underline{Ш}}_i^* = Y_{\text{об}} \overline{Q}_i \vee Y_{\text{об}} Q_i; \quad (3.4)$$

де $Y_{\text{пр}}$ і $Y_{\text{об}}$ — керуючі сигнали видачі відповідно прямого або оберненого коду; Q_i і \overline{Q}_i — пряме та інверсне значення виходу i -го розряду регістра; $\underline{Ш}_i$ — розряд однофазної шини даних; $\underline{Ш}_i^*$ і $\overline{\underline{Ш}}_i^*$ — розряди парафазної шини даних.

Очевидно, що керуючі сигнали $Y_{\text{пр}}$ і $Y_{\text{об}}$ не повинні збігатися в часі. Наприклад, при зчитуванні інформації парафазним оберненим кодом отримаємо:

$$Y_{\text{пр}} = 0; Y_{\text{об}} = 1; \underline{Ш}_i^* = \overline{Q}_i; \overline{\underline{Ш}}_i^* = Q_i.$$

Схеми вихідної логіки для i -го розряду на основі рівнянь (3.3) і (3.4) показані на рис. 3.5.

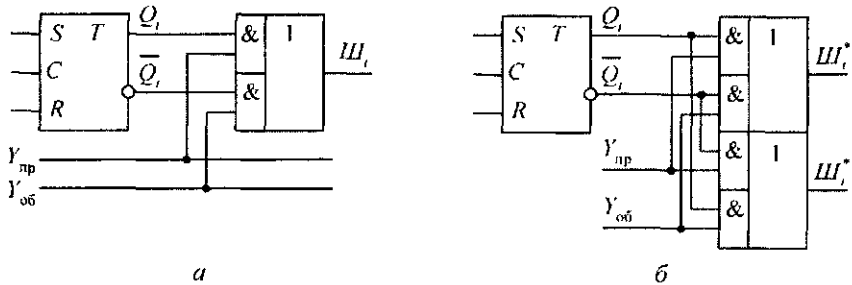


Рис. 3.5. Схеми вихідної логіки i -го розряду регістра для зчитування інформації:
а — однофазним кодом; б — парафазним кодом

3.1.5. Логічні мікрооперації в регістрах

У регістрах можуть виконуватися такі порозрядні (без перенесень) логічні мікрооперації над словами A і B : логічне додавання і множення: $RG1 := A \vee B$; $RG1 := A \wedge B$; додавання за модулем два і його заперечення: $RG1 := A \oplus B$; $RG1 := \overline{A \oplus B}$; інверсія слова: $RG1 := \overline{A}$.

Логічні мікрооперації передбачають наявність першого слова A в регістрі. З урахуванням цього логічне додавання слів A і B в регістрі на RS - або JK -тригерах з однофазним записом виконується введенням слів B без попереднього скидання. Логічне множення реалізується подаванням інверсних значень розрядів слова B на входи R (або K) тригерів регістра. Дійсно, якщо значення $B_i = 0$, то $\overline{B}_i = 1$ і відповідно тригери обнуляються, що і потрібно для порозрядного логічного множення.

Мікрооперації додавання за модулем два і його заперечення реалізуються в регістрах на T -тригерах. Спочатку записується слово A , а потім без попереднього скидання по лічильному входу вводиться слово B . Після цього на прямих виходах тригерів фіксується результат операції $Q = A \oplus B$, а на інверсних виходах — $\overline{Q} = \overline{A \oplus B}$. Мікрооперація інвертування складається з подавання імпульсу на всі T -входи тригерів регістра, в яких зберігається слово A . У підсумку на прямих виходах тригерів встановлюється результат згідно із співвідношенням $Q_i = A_i \oplus 1 = \overline{A}_i$.

3.1.6. Мікрооперації зсуву

Зсув — це одночасне просторове переміщення двійкового слова в розрядній сітці із збереженням порядку слідування нулів і одиниць. Регистри, призначені для виконання мікрооперацій зсуву, називаються регістрами зсуву або зсувовими.

Мікрооперації зсуву використовують у процесі виконання команд множення, ділення і нормалізації. Крім того, за допомогою зсуву здійснюється перетворення паралельного коду в послідовний або навпаки (наприклад, при обміні інформацією з магнітними стрічками і дисками).

Зсув слова може виконуватися вправо (у бік молодших розрядів) або вліво (у бік старших розрядів). Позначимо однорозрядні мікрооперації зсуву вправо і вліво символами R і L відповідно. Розрізняють правий і лівий арифметичний (R_a, L_a), логічний (R_n, L_n) і циклічний (R_u, L_u) зсуви слова.

Нехай в регістрі A записано слово $A_n A_{n-1} \dots A_2 A_1$, де A_1 — молодший розряд, A_n — старший розряд. Символічно мікрооперації зсуву записуються таким чином:

- арифметичні зсуви (знаковий розряд не зсувається)

$$RGA = R_a(A) = A_n 0 A_{n-1} \dots A_2, \quad RGA = L_a(A) = A_n A_{n-2} \dots A_1 0,$$

- логічні зсуви (одночасно зсуваються всі розряди)

$$RGA = R_n(A) = 0 A_n A_{n-1} \dots A_2, \quad RGA = L_n(A) = A_{n-1} A_{n-2} \dots A_1 0,$$

- циклічні зсуви (між старшим і молодшим розрядами є кільцевий зв'язок)

$$RGA = R_u(A) = A_1 A_n A_{n-1} \dots A_2, \quad RGA = L_u(A) = A_{n-1} A_{n-2} \dots A_1 A_n$$

Арифметичні та циклічні зсуви переважно використовують при виконанні команд в процесорах, а логічні зсуви забезпечують перетворення послідовного коду в паралельний і навпаки в пристроях зв'язку з магнітними стрічками і дисками.

Зсувні регистри проектують на двоступеневих RS - (або JK -) або D -тригерах з динамічним керуванням по фронту. Такі тригери забезпечують розділення під час процесів приймання нової інформації в кожному розряді та видачі (зсуву) старої.

Припустимо, що в реверсивному регістрі мають виконуватися паралельний запис слова A за сигналом керування Y_{3n} і зсуви інформації вліво і вправо під впливом керуючих сигналів L_n і R_n . Функції збудження S_i і R_i для кожного розряду регістра зсуву на двоступеневих RS -тригерах мають вигляд:

$$S_i = Y_{3n} A_i \vee R_n Q_{i+1} \vee L_n Q_{i-1}, \quad R_i = Y_{3n} \overline{A_i} \vee R_n \overline{Q_{i+1}} \vee L_n \overline{Q_{i-1}} \quad (3.5)$$

Схема одного розряду регістра зсуву на RS -тригерах згідно з рівнянням (3.5) показана на рис. 3.6, а.

Економічна схема розряду регістра, в якій функція на R -вході тригера утворюється інвертуванням сигналу S_i , показана на рис. 3.6, б. Однак у цьому випадку час записування даних збільшується за рахунок затримки інвертора.

Для регістра зсуву на D -тригерах функції збудження мають вигляд:

$$D_i = Y_{3n} A_i \vee R_n Q_{i+1} \vee L_n Q_{i-1} \quad (3.6)$$

Схема розряду регістра зсуву на D -тригерах згідно з рівнянням (3.6) показана на рис. 3.6, в.

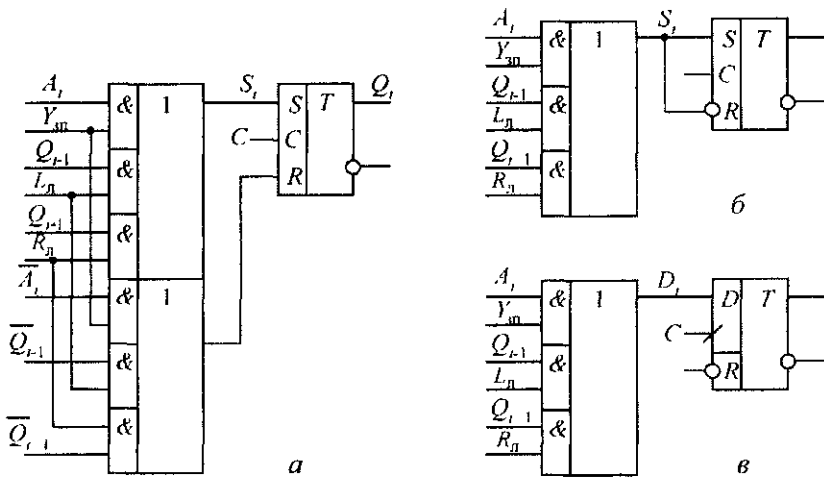


Рис. 3.6. Схема розряду регістра зсуву:

а — на RS-тригери; б — з інвертором на вході R; в — на D-тригери

Приклад побудови реверсивного трирозрядного регістра зсуву на D-тригерах з динамічним керуванням показаний на рис. 3.7.

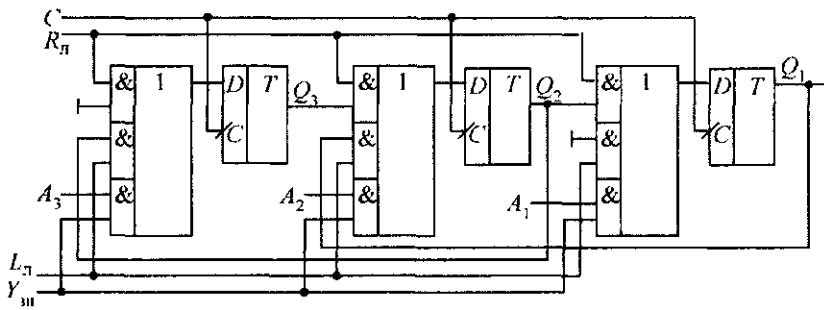


Рис. 3.7. Схема реверсивного регістра зсуву на D-тригерах

Реверсивний регістр зсуву працює таким чином. При значенні сигналу $Y_{шп} = 1$ в регістр записується інформація паралельним однофазним кодом. При значенні сигналу $R_п = 1$ інформація, що зберігається, одночасно зсувається у бік молодших розрядів, при цьому розряд Q_3 обнуляється. При значенні сигналу $L_п = 1$ інформація в регістрі одночасно зсувається у бік старших розрядів, при цьому розряді Q_1 обнуляється. Запис і зсув інформації відбуваються по фронту синхроімпульсу.

3.1.7. Перетворення послідовного коду в паралельний і навпаки

Схема чотирирозрядного регістра зсуву вправо на JK-тригерах, яка забезпечує перетворення кодів, показана на рис. 3.8, а. Старший розряд регістра за допомогою інвертора на K-вході працює в режимі D-тригера.

Нехай від накопичувачів на магнітних дисках або стрічках на вхід регістра по лінії D поступає послідовний код слова $A = 1101$ в напрямку від молодших розрядів до старших. Значення розрядів слова поступає одночасно із синхроімпульсами, які за-

безпечують як приймання коду в старший розряд, так і одночасний зсув вмісту регістра вправо (рис. 3.8, б). Після приходу чотирьох синхроімпульсів на виходах регістра $Q_4 - Q_1$ встановлюється код 1101. Таким чином здійснюється перетворення послідовного коду в паралельний, яке часто називають послідовним введенням слова в регістр.

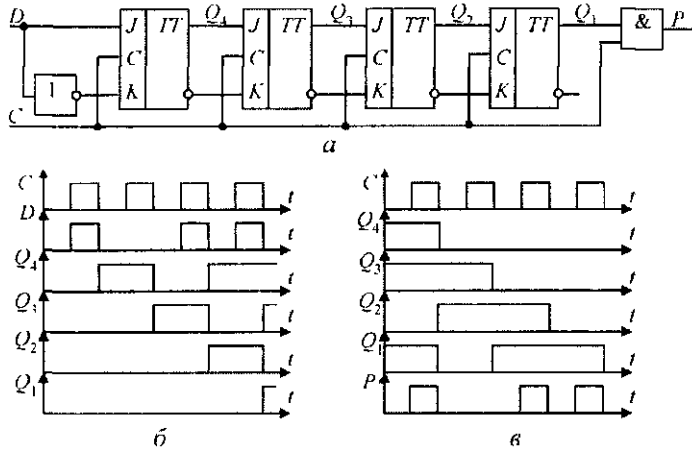


Рис. 3.8. Регістр зсуву :

а — схема ; б, в — перетворення послідовного коду в паралельний і назапки

Перетворення паралельного коду в послідовний також відбувається зсувом слова, яке зберігається. Процес перетворення паралельного коду слова $A = 1101$ в послідовний в напрямку від молодших розрядів до старших за допомогою зсуву вправо показаний на рис. 3.8, в.

3.1.8. Розподільники тактів

У системах синхронізації і пристроях керування використовують розподільники тактів (РТ), які виробляють серії імпульсів. Найпростіший спосіб побудови розподільника тактів полягає в розподілі лічильними тригерами частоти прямокутних імпульсів f від задавального генератора (ЗГ).

На прямому й інверсному виходах тригера за допомогою схем збігу утворюються дві послідовності імпульсів фаз $C_1 = fQ$ і $C_2 = f\bar{Q}$ (рис. 3.9, а). Тривалість такту синхронізації T_C в два рази більша за період надходження імпульсів ЗГ (рис. 3.9, б).

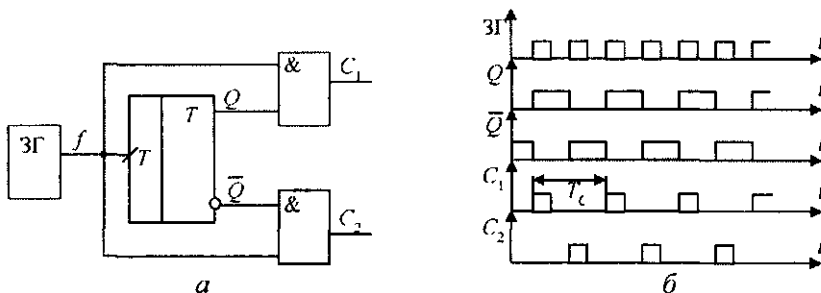


Рис. 3.9. Розподільник тактів: а — схема; б — часові діаграми

Розподільники тактів будуються також на кільцевих регістрах, в яких вихід молодшого розряду з'єднаний з інформаційним входом старшого, якщо реалізований зсув вправо (рис. 3.10, а), або є зв'язком від старшого розряду до молодшого при зсуві вліво.

Кільцевий регістр як розподільник тактів функціонує таким чином. Спочатку через схему ЧИ в старший розряд Q_4 записується одиниця. Потім по фронту кожного синхроімпульсу одиниця зсувається вправо і по чергово з'являється на виходах Q_3 , Q_2 і Q_1 . З приходом четвертого синхроімпульсу одиниця з молодшого розряду Q_1 по кільцевому зв'язку і через схему ЧИ знову записується в старший розряд. Число виходів розподільника тактів визначається кількістю тригерів кільцевого регістра. У розподільнику тактів на основі чотирирозрядного кільцевого регістра формуються імпульси фаз C_1 , C_2 , C_3 і C_4 , які утворюють машинний такт T тривалістю в чотири періоди синхроімпульсів, тобто $T = 4 \cdot T_c$ (рис. 3.10, б).

Недоліком схеми, зображеної на рис. 3.10, а, є можливість збою — появи зайвих одиниць в розрядах, наприклад, 1100, 1001.

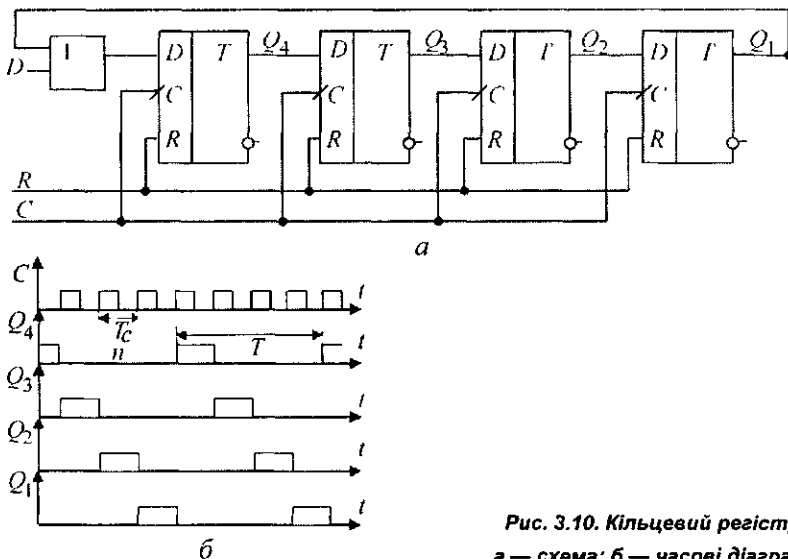


Рис. 3.10. Кільцевий регістр:
а — схема; б — часові діаграми

У кільцевому регістрі із самовідновленням після збою використовується схема контролю, яка при неправильній роботі автоматично повертає регістр у початковий стан.

3.1.9. Регістри в мікросхемах серії КР1533

Серія мікросхем КР1533 містить широкий набір регістрів. Більшість регістрів — восьмирозрядні з третім станом виходу і одно- або двостороннім зсувом (рис. 3.11). Мікросхема КР1533ИР13 (рис. 3.11, а) — це восьмирозрядний реверсивний зсувний регістр. Він має входи для паралельного записування даних D_0 – D_7 , синхронізації C , послідовного приймання DR і DL , задання режимів роботи S_1 , S_0 та вхід скидання R . Інформація зчитується з паралельних виходів Q_0 – Q_7 . Розряди нумеруються зліва направо: Q_0 — старший розряд, Q_7 — молодший.

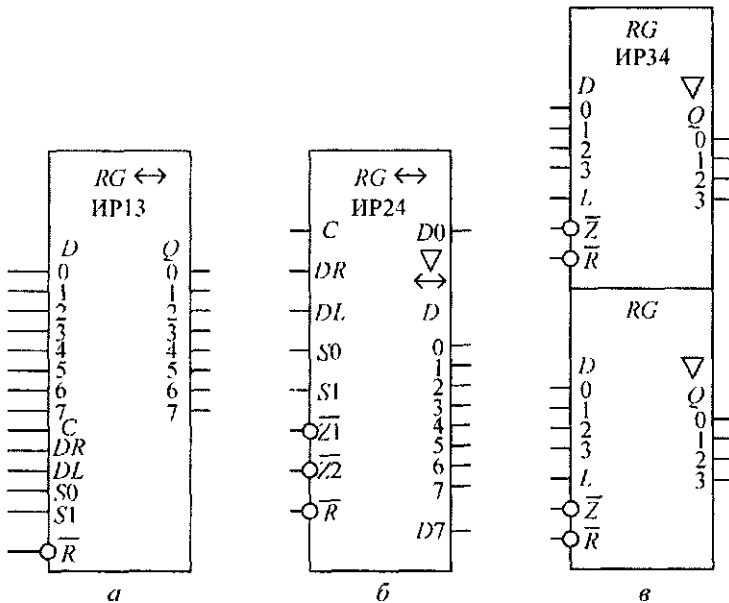


Рис. 3.11. Регістри серії КР1533: а — ІР13; б — ІР24; в — ІР34

Регістр забезпечує чотири режими роботи: $S1 \cdot S0 = 00$ — зберігання даних (блокування); 01 — послідовне приймання зі зсувом вправо по входу DR ; 10 — послідовне приймання зі зсувом вліво по входу DL ; 11 — паралельне записування по входу $D0-D7$. Записування і послідовне приймання даних здійснюються по фронту синхроімпульса C . Під час записування послідовне введення даних блокується. При $R = 0$ всі входи регістра устанавлюються в стан "0".

Мікросхема КР1533ІР24 (рис. 3.11, б) — це восьмирозрядний реверсивний зсувовий регістр з двонаправленими виводами $D0-D7$ для прийому і видачі інформації. Виводи $D0-D7$ мають Z -стани, в які вони переключаються при подачі сигналу лог. 1 хоча б на один із входів $Z1$ чи $Z2$. Регістр має також додаткові виводи $D0$ і $D7$, які зв'язані відповідно із старшим і молодшим розрядами, але не мають Z -станів.

Регістр забезпечує чотири синхронні режими роботи залежно від сигналів на входах $S1 \cdot S0$: 00 — зберігання (блокування); 01 — послідовне записування по входу DR із зсувом вправо; 10 — послідовне записування по входу DL із зсувом вліво; 11 — паралельний прийом даних. Перші три режими реалізуються при $Z1 = Z2 = 0$; при цьому інформація передається на виводи $D0-D7$, а четвертий режим здійснюється незалежно від значень сигналів на виходах $Z1$ і $Z2$. Всі операції виконуються по фронту сигналу синхронізації C .

Регістри ІР13, ІР24 серії КР1533 використовують для тимчасового зберігання інформації, перетворення даних з паралельної форми в послідовну і навпаки.

Мікросхема КР1533ІР34 (рис. 3.11, в) — це два незалежні чотирирозрядні регістри з трьома станами на виходах $Q0-Q3$. Паралельний запис даних по входах $D0-D3$ проводиться при сигналі лог. 1 на вході дозволу L ; якщо $L = 0$, то регістр переходить в режим зберігання. Високий рівень сигналу на вході Z переключає виходи регістра в третій стан. Але при цьому в регістр можуть записуватись нові дані або зберігатись попередні. Мікросхема має велику навантажувальну здатність, що в по-

єднанні з трьома станами дозволяє використовувати регістр в магістральних структурах без додаткових інтерфейсних схем

Потужність споживання регістрів ІР13 ІР24 та ІР34 серії КР1533 становить приблизно 200 мВт, а час записування інформації — 15–20 нс

3.2. Лічильники

3.2.1. Загальна характеристика лічильників

Лічильником називається типовий функціональний вузол комп'ютера, призначений для лічби входних імпульсів Лічильник являє собою зв'язаний ланцюг T -тригерів, які утворюють пам'ять із заданим числом сталих станів (рис 3 12)

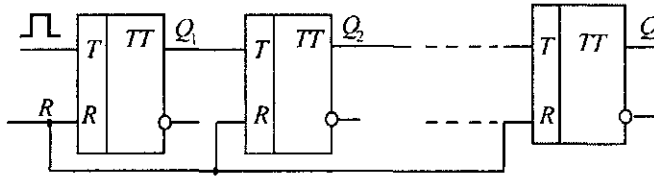


Рис 3 12 Логічна структура лічильника

Розрядність лічильника n дорівнює числу T -тригерів Кожний входний імпульс змінює стан лічильника, який зберігається до надходження наступного сигналу Значення виходів тригерів лічильника Q_n, Q_{n-1}, \dots, Q_1 відображають результат лічби в прийнятій системі числення Логічна функція лічильника позначається буквами CT (counter) Список мікрооперацій лічильника вміщує попереднє встановлення в початковий стан, інкремент або декремент слова, яке зберігається, видачу слів паралельним кодом та ін

Вхідні імпульси можуть надходити на лічильник як періодично, так і довільно розподіленими у часі Амплітуда і тривалість лічильних імпульсів мають задовольняти технічні вимоги для серій мікросхем, які використовуються

Лічильник є одним з основних функціональних вузлів комп'ютера, а також різних цифрових керуючих та інформаційно-вимірювальних систем Основне застосування лічильників

- утворення послідовності адрес команд програми (лічильник команд або програмний лічильник),
- підрахунок числа циклів при виконанні операцій ділення, множення, зсуву (лічильник циклів)
- одержання сигналів мікрооперацій і синхронізації, аналого-цифрові перетворення і побудова електронних таймерів (годинників реального часу)

Лічильник характеризується модулем і ємністю лічби Модуль лічби $K_{лч}$ визначає число станів лічильника Модуль двійкового n -розрядного лічильника визначається цілим степенем двійки $M = 2^n$, в лічильниках інших типів справедлива нерівність $K_{лч} \leq M$ Після лічби числа імпульсів $N_{вх} = K_{лч}$ лічильник повертається в початковий стан Таким чином, модуль лічби, який часто називають коефіцієнтом перерахунку, визначає цикл роботи лічильника, після чого його стан повторюється Тому

число вхідних імпульсів і стан лічильника однозначно визначені тільки для першого циклу

Ємність лічби N_{\max} визначає максимальну кількість вхідних імпульсів, яку може зафіксувати лічильник при одному циклі роботи. Ємність лічби $N_{\max} = K_{\text{ЛЧ}} - 1$ за умови, що робота лічильника починається з нульового початкового стану

У лічильниках використовуються три режими роботи: керування, накопичення і ділення. У режимі керування зчитування інформації виконується після кожного вхідного лічильного імпульсу, наприклад, в лічильнику адреси команд. У режимі накопичення головним є підрахунок заданого числа імпульсів або лічба протягом певного часу. У режимі ділення (перерахунку) основним є зменшення частоти надходження імпульсів в $K_{\text{ЛЧ}}$ разів. Більшість лічильників може працювати в усіх режимах, проте в спеціальних лічильниках-дільниках стани в процесі лічби можуть змінюватися в довільному порядку, що дозволяє спростити схему вузла.

Лічильники класифікують за такими ознаками:

- способом кодування — позиційні та непозиційні,
- модулем лічби — двійкові, десяткові, з довільним постійним або змінним (програмованим) модулем,
- напрямком лічби — прості (підсумовуючі, віднімальні) і реверсивні,
- способом організації міжрозрядних зв'язків — з послідовним, наскрізним, паралельним і комбінованим переносами (позицією),
- типом використовуваних тригерів — T , JK , D в лічильному режимі,
- елементним базисом — потенціальні, імпульсні та потенціально-імпульсні.

У лічильниках з позиційним кодуванням числовий вираз поточного стану лічильника визначається за формулою

$$N = \sum_{i=1}^n r_i Q_i = r_n Q_n + r_{n-1} Q_{n-1} + \dots + r_1 Q_1,$$

де r_i — вага i -го розряду, Q_i — значення виходу i -го розряду, n — число розрядів. Нульове значення всіх розрядів звичайно беруть як початковий стан лічильника. Всі інші стани нумерують за числом вхідних імпульсів, що надійшли.

У лічильниках з непозиційним кодуванням (наприклад, у кодах Грея) розряди не мають постійних ваг і кожному набору станів Q_n, Q_{n-1}, \dots, Q_1 приписується певна кількість вхідних імпульсів. У комп'ютерах переважно використовують лічильники з позиційним кодуванням.

За видом переходів прості лічильники (ЛЧ) розподіляються на підсумовуючі (прямої лічби) і віднімальні (зворотної лічби). У підсумовуючих лічильниках кожний доданий імпульс U^+ збільшує стан на одиницю, тобто реалізується мікрооперація інкремента $\text{ЛЧ} = \text{ЛЧ} + 1$. Граф переходів підсумовуючого лічильника показаний на рис. 3.13, а, де вершини означають стійкі стани. Із $M-1$ -го стану черговий сигнал U^+ повертає лічильник у початковий стан і видає сигнал переповнення P .

У віднімальних лічильниках кожний віднімальний імпульс U^- зменшує стан на одиницю, тобто реалізується мікрооперація декремента $\text{ЛЧ} = \text{ЛЧ} - 1$. Граф переходів віднімального лічильника (рис. 3.13, б) характеризується наявністю переходів тільки в зворотному напрямку від деякого (наприклад, $M-1$ -го) попередньо встановленого стану. Після віднімання M імпульсів лічильник видає сигнал позики Z і повертається в початковий $M-1$ -й стан.

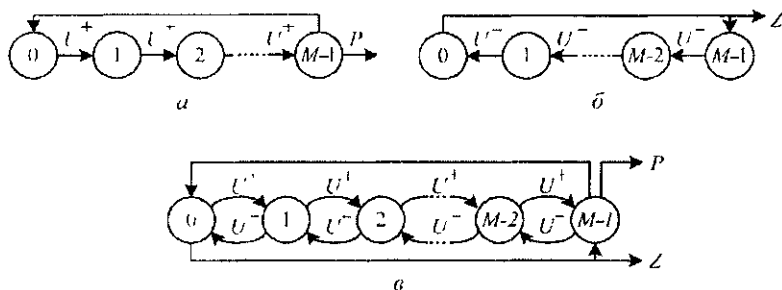


Рис. 3.13. Графи лічильників: а — підсумовуючого; б — віднімального; в — реверсивного

Реверсивні лічильники мають переходи в прямому і зворотному напрямках, що дозволяє рахувати підсумовуючі та віднімальні імпульси (рис. 3.13, в). У процесі лічби має виконуватися умова $\sum U^+ + N_n \geq \sum U^-$, де N_n — попередньо записане число. За поточним станом виходів лічильника визначається результат реверсивної лічби:

$$\Delta N = \sum U^+ + N_n - \sum U^-.$$

До часових характеристик лічильників відносяться роздільна здатність, швидкодія і час встановлення (перемикання) коду.

Роздільна здатність t_{pz} визначається мінімальним інтервалом часу між двома вхідними імпульсами, при якому ще зберігається працездатність лічильника. Параметр t_{pz} задають часом перемикання t_f першого (молодшого) тригера лічильника, тобто $t_{pz} = t_f$, оскільки він перемикається під дією кожного вхідного імпульсу.

Швидкодія лічильника визначається максимальною частотою F_m надходження вхідних імпульсів в режимі ділення й обчислюється за формулою $F_m = 1/t_f$. Час встановлення коду $t_{вст}$ відраховується від початку вхідного імпульсу до моменту отримання нового стану. Даний параметр дозволяє обчислювати швидкодію лічильника в режимі керування із співвідношення $F_{m,к} = 1/(t_{вст} + t_{зч})$, де $t_{зч}$ — час зчитування інформації.

Міжрозрядні зв'язки забезпечують вироблення сигналів перенесення в старші розряди при додаванні імпульсів і сигналів позики — при відніманні. Від виду реалізації міжрозрядних зв'язків суттєво залежать параметри $t_{вст}$ і $F_{m,к}$.

У лічильниках з послідовними перенесеннями тригери перемикаються по чергово після кожного вхідного імпульсу в напрямку від молодших розрядів до старших. Такі лічильники називаються послідовними або асинхронними. У лічильниках з паралельними перенесеннями тригери перемикаються одночасно після кожного вхідного імпульсу, такі лічильники називаються паралельними або синхронними.

3.2.2. Двійкові підсумовуючі та віднімальні лічильники

Двійкові лічильники реалізують лічбу вхідних імпульсів у двійковій системі числення. Число розрядів n двійкового підсумовуючого лічильника для заданого модуля M знаходять із виразу $n = \log_2 M$. Значення поточного числа N^+ вхідних імпульсів n -розрядного підсумовуючого лічильника при відліку з нульового початкового стану визначають за формулою

$$N^+ = \sum_{i=1}^n 2^{i-1} Q_i = 2^{n-1} Q_n + 2^{n-2} Q_{n-1} + \dots + 2^0 Q_1,$$

де 2^{i-1} — вага i -го розряду, $Q_i \in \{0,1\}$ — логічне значення прямого виходу тригера i -го розряду. Розряди двійкового лічильника будуються на двоступеневих T -тригерах або D -тригерах з динамічним керуванням по фронту синхросигналу (в лічильному режимі)

У двійковому підсумовуючому лічильнику перенесення P_i в сусідній старший розряд Q_{i+1} виникає в тому випадку, коли в момент надходження чергового лічильного імпульсу U^+ всі молодші розряди знаходяться в одиничному стані, тобто $P_i = U^+ Q_i Q_{i-1} \dots Q_1 = 1$. Після вироблення перенесення старший розряд перемикається в стан "1", а всі молодші розряди — в стан "0".

Асинхронні підсумовуючі лічильники на двоступеневих T -тригерах будуються так, щоб вхідні імпульси U^+ надходили на лічильний вхід тільки першого (молодшого) розряду. Сигнали перенесення передаються асинхронно (попередовно в часі) з прямих виходів молодших розрядів на T -входи сусідніх старших, як показано на рис 3 14 для трирозрядного лічильника

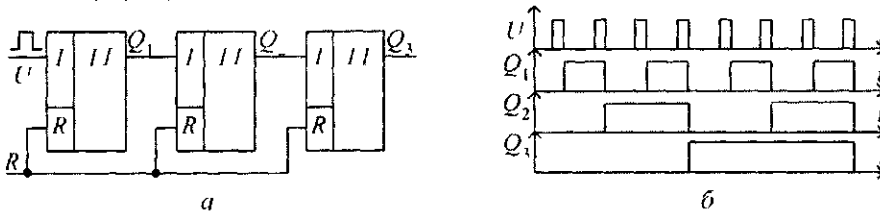


Рис 3 14. Асинхронний підсумовуючий лічильник на двоступеневих T -тригерах

а — схема; б — часові діаграми роботи

Зміна станів тригерів відбувається за спадом лічильного імпульсу для першого розряду, а для останніх — за спадом сигналу перенесення (рис 3 14, б)

Після підрахунку семи імпульсів на виході трирозрядного лічильника встановлюється двійковий код $Q_3 Q_2 Q_1 = 111$ (тобто максимальне значення або ємність лічби). Після приходу восьмого вхідного імпульсу U^+ трирозрядний підсумовуючий лічильник перемикається у початковий нульовий стан послідовно (асинхронно) в часі спочатку спадає напруга на виході Q_1 , потім — на виході Q_2 і т.д.

За допомогою імпульсу по входу скидання R лічильник повертається в нульовий стан у будь-який момент часу.

Часові параметри n -розрядного асинхронного лічильника на двоступеневих тригерах визначаються із співвідношень

- роздільна здатність $t_{рз} = t_{T1} + t_{T2} = 2t_T$, де t_{T1} і t_{T2} — час перемикавання першого і другого ступенів тригера, причому $t_{T1} = t_{T2} = t_T$,
- час встановлення $t_{вст} = t_{T1} + nt_T \approx nt_T$,
- максимальна частота лічби в режимі ділення і керування відповідно $F_m = 1/2t_T$, $F_{мк} \approx 1/nt_T$

В режимі керування швидкодія асинхронного лічильника приблизно в n раз менша ніж в режимі ділення.

Схема трирозрядного асинхронного двійкового підсумовуючого лічильника на T -тригерах з динамічним керуванням по фронту показана на рис 3 15. Лічильні імпульси U^+ надходять на T -вхід тільки першого (молодшого) розряду, наступні тригери перемикаються асинхронно від сигналів перенесення з інверсних виходів сусідніх молодших розрядів.

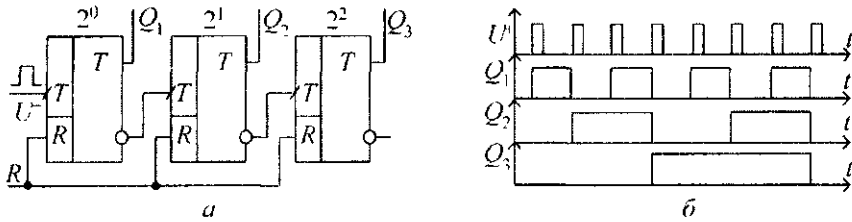


Рис. 3.15. Асинхронний підсумовуючий лічильник на тригерах з динамічним керуванням по фронту: а — схема; б — часові діаграми роботи

Часові параметри асинхронного n -розрядного двійкового лічильника на тригерах з динамічним керуванням: $t_{pz} = t_T$, $t_{acr} = nt_T$, $F_m = 1/t_T$, $F_{mк} \approx 1/nt_T$.

Перевагою асинхронних лічильників є простота схеми: збільшення розрядності виконується підключенням необхідного числа тригерів. До недоліків асинхронних лічильників відносяться порівняно низька швидкодія в режимі керування та її залежність від числа розрядів, а також поява проміжних вихідних двійкових кодів у процесі послідовного перемикавання тригерів у новий стан.

Для одержання мінімального часу перемикавання лічильника використовують паралельні перенесення (рис. 3.16). Для цього в кожному розряді синхронного лічильника є схема збігу, за допомогою якої аналізуються стани всіх попередніх молодших тригерів і вироблюються функції перенесення згідно з такими логічними співвідношеннями:

$$P_1 = U Q_1; P_2 = U Q_2 Q_1; P_3 = U Q_3 Q_2 Q_1; P_4 = U' Q_4 Q_3 Q_2 Q_1.$$

При надходженні чергового лічильного імпульсу U' перемикаються тільки ті тригери, для яких усі попередні (молодші) розряди знаходяться в цей момент в одиничному стані.

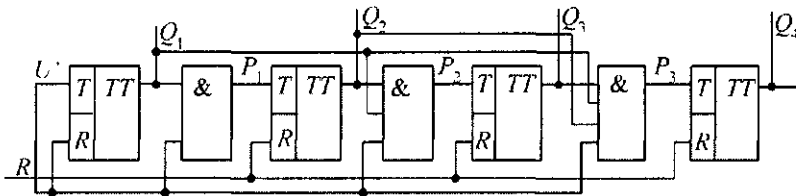


Рис. 3.16. Схема підсумовуючого лічильника з паралельними перенесеннями

Час установлення синхронного лічильника не залежить від числа розрядів і дорівнює $t_{вст} = t_{TT} + t_P$, де t_{TT} — час перемикавання двоступеневого тригера; t_P — час затримки розповсюдження сигналу вентилем у ланцюзі перенесення. Максимальна частота лічби в режимі керування $F_m \approx 1/t_{TT}$. Таким чином, синхронні лічильники забезпечують найбільшу швидкоддю в режимі керування.

При побудові багаторозрядних синхронних лічильників з'являються труднощі — зростання числа входів вентилів у ланцюзі перенесення і збільшення навантаження на виходи тригерів.

У двійковому віднімальному лічильнику кожний віднімальний імпульс U зменшує стан на одиницю. Поточне значення кількості вхідних імпульсів n -розрядного двійкового віднімального лічильника визначається за формулою

$$\sum U = N_n - N,$$

де N — значення коду на прямих виходах тригерів лічильника; N_n — попередньо записане початкове число; при цьому має виконуватися умова $\sum U \leq N_n$.

У віднімальних лічильниках сигнали міжрозрядного зв'язку називаються позиками. За правилом двійкового віднімання в момент надходження лічильного імпульсу U позика із старшого розряду з одиничним значенням виникає за умови, що всі молодші тригери знаходяться в нульовому стані. Після цього всі вони перемикаються в стан "1", а старші — в стан "0". Сигнали позики утворюються на інверсних виходах двоступеневих тригерів або на прямих виходах тригерів з динамічним керуванням по фронту.

Параметри віднімального лічильника (модуль і ємність лічби, швидкодія) збігаються з аналогічними характеристиками підсумовуючих лічильників.

Схема трирозрядного двійкового асинхронного віднімального лічильника на двоступеневих тригерах показана на рис. 3.17, а.

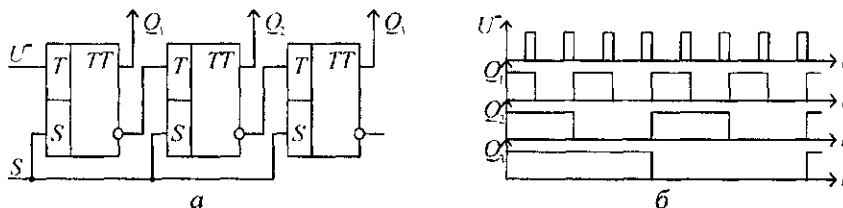


Рис. 3.17. Асинхронний віднімальний лічильник на двоступеневих тригерах:

а — схема; б — часові діаграми роботи

Перед початком роботи за допомогою сигналу на спільному вході S всі тригери лічильника встановлюються в стан "1", утворюючи вихідний код 111. Віднімальний імпульс U надходить на лічильний вхід лише першого молодшого розряду, міжрозрядні сигнали позики знімаються асинхронно з інверсних виходів тригера.

Після надходження семи віднімальних імпульсів усі тригери лічильника устанавлюються в стан "0", утворюючи вихідний код 000. Восьмий віднімальний імпульс перемикає лічильник в стан 111 (за умови, що розглядається трирозрядна схема).

У віднімальному лічильнику на тригерах з динамічним керуванням по фронту сигнали позики знімаються з прямих виходів тригерів.

3.2.3. Двійкові реверсивні лічильники

Двійкові реверсивні лічильники мають переходи у двох напрямках: в прямому (при лічбі підсумовуючих сигналів U) і в зворотному (при переліку віднімальних сигналів U). Поточне значення різниці підрахованих імпульсів визначається із співвідношення

$$\sum U - \sum U = N - N_n,$$

де N — значення коду на прямих виходах тригерів лічильника; N_n — попередньо записане в лічильник початкове число.

В лічбі має виконуватися умова $\sum U \leq N_n + \sum U \leq 2^n - 1$.

Розрізняють одноканальні та двоканальні реверсивні лічильники. В одноканальних реверсивних лічильниках підсумовуючі U і віднімальні U сигнали по чергово надходять на спільний лічильний вхід, а напрямок лічби задається напрямком ланцюгів міжрозрядних перенесень або позик. Для перемикання міжрозрядних зв'язків у

одноканальному реверсивному лічильнику потрібні додаткові керуючі сигнали

Двоканальні реверсивні лічильники мають два лічильних входи один для підсумовуючих імпульсів U , другий — для віднімальних U . Перемикання ланцюгів міжрозрядних зв'язків здійснюється автоматично лічильними сигналами для переносів — імпульсами U для позики — імпульсами U . Схема одноканального трирозрядного двійкового реверсивного лічильника показана на рис. 3.18. Міжрозрядні зв'язки комутуються за допомогою логічних елементів І ЧИ.

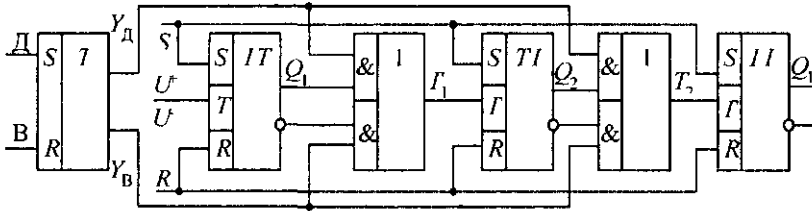


Рис. 3.18. Схема одноканального реверсивного лічильника

Для задання напрямку лічби використовують додатковий RS-тригер з його прямого виходу знімається сигнал керування додаванням Y_d (вмикає ланцюги перенесення), а з інверсного виходу — сигнал керування відніманням Y_b (вмикає ланцюги позики). На виходах елементів І ЧИ (які називаються "схеми реверса") виробляється сигнал T_i для лічильних входів старших розрядів

$$T_i = Y_d Q_i \vee Y_b \overline{Q_i}, \quad i = 1, 2, 3, \dots, n$$

Таким чином, якщо керуючий RS-тригер знаходиться в стані "1", то лічильник реалізує режим прямої лічби вхідних імпульсів (тобто підсумовування), в іншому випадку — забезпечує режим зворотної лічби (віднімання). В обох режимах роботи тригери перемикаються асинхронно.

Схема двоканального чотирирозрядного двійкового реверсивного лічильника показана на рис. 3.19. Лічильні T-входи в тригерах внутрішньо зв'язані схемою ЧИ.

Підсумовуючі імпульси U надходять на лічильний вхід першого (молодшого) розряду лічильника і одночасно — на входи всіх вентилів у ланцюзі паралельного перенесення. При цьому формуються імпульси міжрозрядних перенесень на основі логічних виразів

$$P_1 = U^+ Q_1, \quad P_2 = U^+ Q_1 Q_2, \quad P_3 = U^+ Q_1 Q_2 Q_3, \quad P_4 = U^+ Q_1 Q_2 Q_3 Q_4$$

Віднімальні імпульси U надходять на лічильний вхід першого розряду лічильника і одночасно на входи всіх вентилів у ланцюзі паралельних позик. При цьому формуються імпульси міжрозрядних позик на основі таких логічних виразів

$$Z_1 = U^- \overline{Q_1}, \quad Z_2 = U^- \overline{Q_1} \overline{Q_2}, \quad Z_3 = U^- \overline{Q_1} \overline{Q_2} \overline{Q_3}, \quad Z_4 = U^- \overline{Q_1} \overline{Q_2} \overline{Q_3} \overline{Q_4}$$

Таким чином, у двоканальних реверсивних лічильниках напрямок лічби безпосередньо задається підсумовуючими U або віднімальними U імпульсами. Забороняється одночасне надходження на входи двоканального реверсивного лічильника підсумовуючих і віднімаючих імпульсів.

На практиці з урахуванням схемотехнічних можливостей мікросхем середнього ступеня інтеграції багаторозрядні реверсивні лічильники будують у вигляді групової структури. При цьому кожна група представляється наприклад мікросхемою чоти-

рирозрядного реверсивного лічильника з паралельними перенесеннями і позиками. Між групами можуть бути утворені послідовні або паралельні зв'язки.

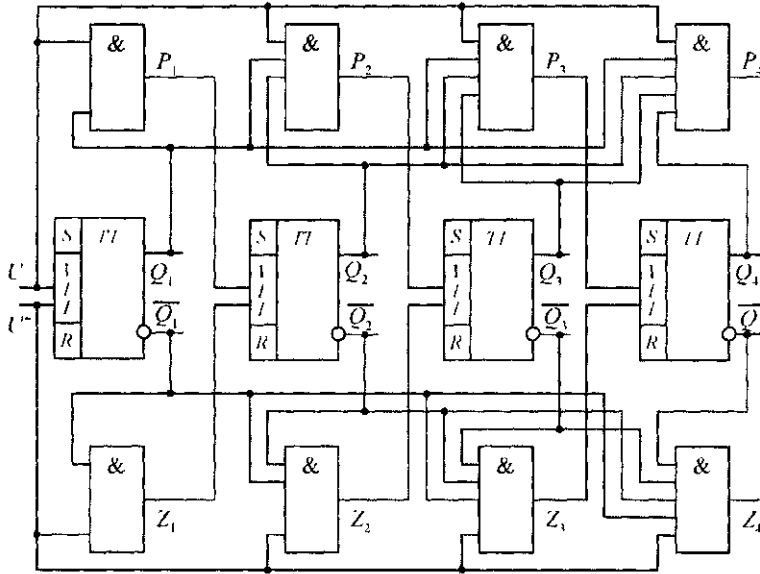


Рис. 3.19. Схема двокального реверсивного лічильника

Схема 12-розрядного двокального реверсивного лічильника групової структури з послідовними перенесеннями між групами (мікросхеми КР1533ІЕ7) показана на рис. 3.20; СТ2 — функція двійкового лічильника.

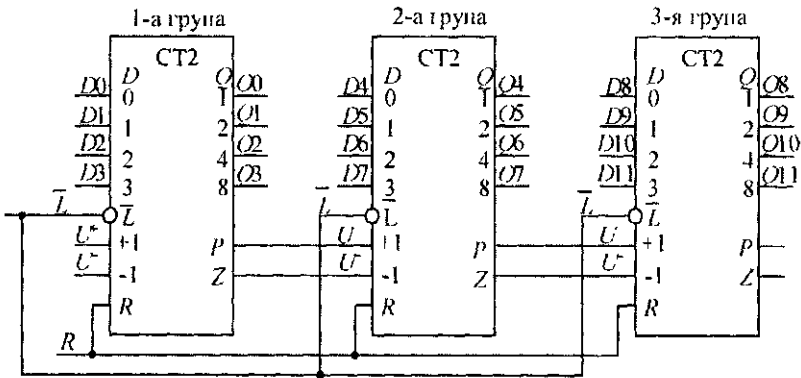


Рис. 3.20. Схема 12-розрядного двокального реверсивного лічильника з груповою структурою

Мікросхема КР1533ІЕ7 — це двійковий чотирирозрядний реверсивний синхронний лічильник, який має такі режими роботи:

- паралельне завантаження початкових даних $D3-D0$ при $\bar{L} = 0$;
- підсумовування імпульсів U за входом "+1";
- віднімання імпульсів U за входом "-1";
- скидання в початковий стан сигналом лог. 1 на R -вході.

Інформація зчитується з виходів $Q_{11}-Q_0$ лічильника. Підсумовуючі імпульси U в лічильнику з груповою структурою поступають на вхід прямої лічби "+1" першої молодшої групи. Для другої групи сигнали перенесення формуються за умови $P = U \overline{Q_3} \overline{Q_2} \overline{Q_1} \overline{Q_0}$. Віднімальні імпульси U поступають на вхід зворотної лічби "-1". Для другої групи сигнали позики виробляються за умови $Z = U \overline{Q_3} \overline{Q_2} \overline{Q_1} \overline{Q_0}$. Аналогічно будуються зв'язки і між старшими групами.

Тривалість імпульсів на лічильних входах має бути не меншою 20 нс, а максимальна частота лічби не перевищувати 30 МГц.

3.2.4. Двійково-десяткові лічильники

Двійково-десяткові лічильники реалізують лічбу імпульсів у десятковій системі числення, причому кожна десяткова цифра від нуля до дев'яти кодується чотирирозрядним двійковим кодом (тетрадою). Ці лічильники часто називають десятковими або декадними, оскільки вони працюють з модулем лічби, кратним десяти.

Багаторозрядний двійково-десятковий лічильник будується на основі регулярного ланцюга декад, при цьому перша (молодша) декада має вагу 10^0 , друга — 10^1 , третя — 10^2 і т.д.

Декада будується на основі чотирирозрядного двійкового лічильника, в якому вилучається надлишкове число станів. Вилучення зайвих шести станів у декаді досягається багатьма способами:

- попереднім записуванням числа 6 (двійковий код 0110), після лічби дев'ятого імпульсу вихідний код дорівнює 1111 і десятковий сигнал повертає лічильник у початковий стан 0110, отже, тут результат лічби фіксується двійковим кодом з надлишком 6;
- блокуванням переносів лічби імпульсів до дев'яти здійснюється у двійковому коді, після чого вмикаються логічні зв'язки блокування перенесень, з надходженням десятого імпульсу лічильник закінчує цикл роботи і повертається в початковий нульовий стан;
- введенням обернених зв'язків, які забезпечують лічбу в двійковому коді й примусовим перемиканням лічильника в нульовий початковий стан після надходження десятого імпульсу.

Схема синхронного десяткового лічильника з блокуванням перенесень показана на рис. 3.21. У цій схемі С-входи використовуються як лічильні. З надходженням десятого імпульсу на С-вхід молодшого розряду JK-тригера обнуляються перший і четвертий розряди і сигналом з виходу Q_4 блокують перемикання другого і третього розряду. Схема підсумовуючого лічильника з оберненими зв'язками (один розряд) показана на рис. 3.22.

Після скидання в нульовий початковий стан на лічильний вхід першого тригера поступають підсумовуючі імпульси U . Сигнали перенесення в старші розряди формуються звичайним асинхронним способом. Лічба до дев'яти ведеться в двійковому коді. Після приходу десятого вхідного імпульсу обернений зв'язок на основі схеми збігу виробляє сигнал $P = U^+ \overline{Q_4} \overline{Q_3} \overline{Q_2} \overline{Q_1}$, який є переносом для старшої декади і одночасно перемикає лічильник в нульовий стан. Далі цикл роботи лічильника повторюється.

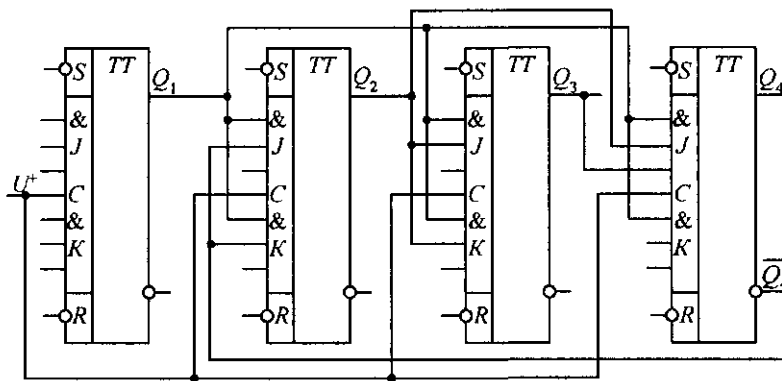


Рис. 3.21. Схема десятичного лічильника на JK-тригерах

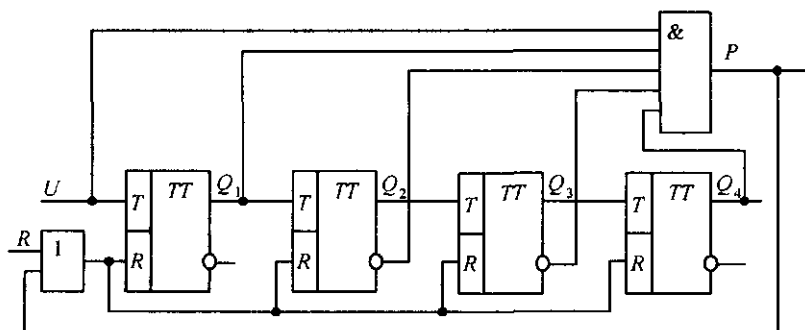


Рис. 3.22. Схема десятичного підсумовуючого лічильника з оберненими зв'язками

Схема п'ятирозрядного підсумовуючого двійково-десятичного лічильника показана на рис. 3.23.

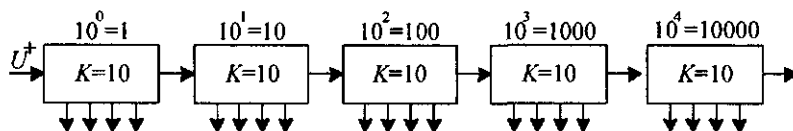


Рис. 3.23. Схема п'ятирозрядного підсумовуючого двійково-десятичного лічильника

Модуль даного лічильника складає $K_{лч} = 10^5 = 100000$, ємність лічби $N_{max} = K_{лч} - 1 = 99999$.

Виходи тригерів кожної декади підключаються до входів дешифраторів, які забезпечують візуальну індикацію стану лічильника за допомогою різного роду світлових табло.

3.2.5. Лічильники з одиничним кодуванням

При одиничному (унітарному) кодуванні стани n -розрядного лічильника розрізняються лише місцеліченням однієї одиниці, яка називається **маркуючим кодом**; в інших розрядах записані нулі. В окремих випадках маркуючий код складається з двох одиниць і називається парно-одиничним.

Лічильник з одиничним кодуванням — це ланцюг тригерів, в якому забезпечується зсув попередньо записаного маркуючого коду по “кільцю” в напрямку старших розрядів (прямий підрахунок) або молодших (обернений підрахунок). Такі лічильники часто називають кільцевими (за аналогією з кільцевими регістрами зсуву).

Лічильник з одиничним кодуванням характеризується модулем $K_{\text{сч}} = n$ і ємністю лічби $N_{\text{max}} = n - 1$. Таким чином, число станів кільцевого лічильника дорівнює його розрядності і є значно меншим порівняно з іншими типами лічильників.

В кільцевих лічильниках кожний розряд має вагу, яка дорівнює номеру стану 0, 1, 2, ..., $(n - 1)$. Із стану $(n - 1)$ після надходження чергового імпульсу лічильник утворює на виході сигнал закінчення циклу (переповнення) і повертається в початковий стан за допомогою кола оберненого зв'язку з виходу старшого розряду Q_n на вхід молодшого розряду Q_1 . Схема чотирирозрядного кільцевого лічильника показана на рис. 3.24, а.

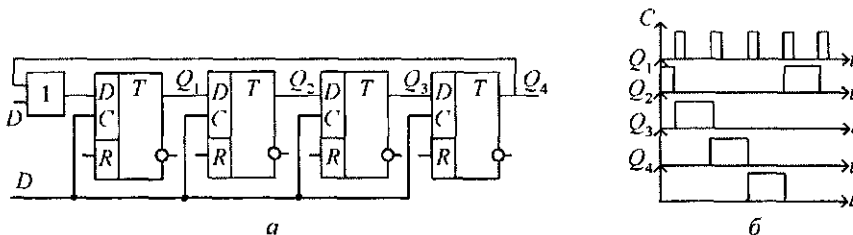


Рис. 3.24. Кільцевий лічильник: а — схема; б — часові діаграми

Перед початком роботи по входу D схеми ЧИ в молодший розряд лічильника записується одиниця і встановлюється початковий код $Q_4Q_3Q_2Q_1 = 0001$. З надходженням кожного лічильного імпульсу по входу синхронізації одиничний код послідовно зсувається в бік старших розрядів; при цьому молодші розряди, виконані на D -тригерах з динамічним керуванням, обнуляються. Після надходження четвертого імпульсу лічильник повертається в початковий стан за допомогою сигналу з виходу Q_4 на вхід схеми ЧИ.

Практичне використання кільцевих лічильників пояснюється такими його перевагами:

- не потребує вихідного дешифратора, оскільки всі стани відрізняються наявністю одиниці лише в одному якому-небудь тригері;
- в процесі лічби завжди переключається в одиничний стан лише один тригер, що забезпечує мінімальне значення $t_{\text{вст}}$;
- спрощується побудова схеми контролю лічильника.

Схема лічильника Джонсона будується на основі кільцевого, в якому обернений зв'язок реалізується підключенням інверсного виходу старшого розряду до входу молодшого.

Лічильник Джонсона (рис. 3.25) характеризується модулем лічби $K_{\text{лч}} = 2n$ і ємністю лічби $N_{\text{max}} = 2n - 1$. Таким чином, число станів лічильника Джонсона в два рази більше аналогічного параметра кільцевого лічильника. Проте інформація на виходах лічильника Джонсона представляється не в двійковій позиційній системі числення, яка потребує додаткового перетворення. Як видно із часових діаграм (рис. 3.25, б), в процесі лічби спочатку рухається “хвиля” одиниць, а потім — “хвиля” нулів. Дешиф-

рація станів лічильника Джонсона здійснюється простіше порівняно з двійковими позиційними лічильниками.

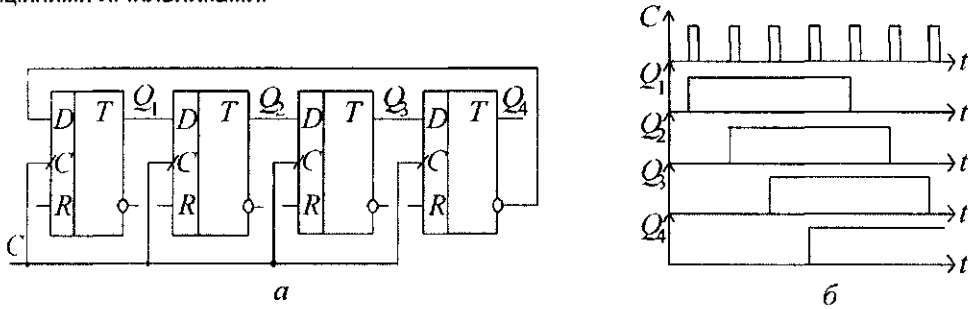


Рис. 3.25. Лічильник Джонсона: а — схема; б — часові діаграми

Контрольні запитання

1. Що таке регістр?
2. Які області застосування регістрів?
3. Сформулюйте ознаки класифікації регістрів.
4. Охарактеризуйте мікрооперації, які може виконувати регістр.
5. Що таке зсуви? Які основні види зсувів?
6. Які логічні операції реалізуються в регістрах?
7. Що таке лічильник?
8. Які області застосування лічильників?
9. Сформулюйте ознаки класифікації лічильників.
10. Охарактеризуйте два режими роботи лічильника — керування і ділення.
11. Як можна збільшити швидкодію лічильника в режимі керування?
12. Чим відрізняється двійковий лічильник від десяткового?
13. Які обмеження накладаються на роботу реверсивного лічильника?
14. За якими правилами організуються зв'язки між тригерами підсумовуючого і віднімального лічильників?
15. Які особливості роботи лічильника на двоступеневих тригерах порівняно з D -тригерами з динамічним керуванням?
16. Які способи побудови лічильників з довільним модулем лічби?
17. Охарактеризуйте лічильник Джонсона.

Розділ 4

Комбінаційні функціональні вузли комп'ютерної схемотехніки

4.1. Дешифратори

4.1.1. Загальна характеристика дешифраторів

Дешифратором називається функціональний вузол комп'ютера, призначений для перетворення кожної комбінації вхідного двійкового коду в керуючий сигнал лише на одному із своїх виходів. У загальному випадку дешифратор має n однофазних входів (іноді $2n$ парафазних) і $m = 2^n$ виходів, де n — розрядність (довжина) коду, який дешифрується. Дешифратор з максимально можливим числом виходів $m = 2^n$ називається **повним**. Функціонування повного дешифратора описується системою логічних виразів вигляду:

$$\begin{aligned} F_0 &= \bar{X}_n \bar{X}_{n-1} \dots \bar{X}_2 \bar{X}_1; \\ F_1 &= \bar{X}_n \bar{X}_{n-1} \dots \bar{X}_2 X_1, \\ &\dots \dots \dots \\ F_{m-1} &= X_n X_{n-1} \dots X_2 X_1, \end{aligned}$$

де X_1, \dots, X_n — вхідні двійкові змінні; F_0, F_1, \dots, F_{m-1} — вихідні логічні функції, що являють собою мінтерми (конституенти 1) n змінних.

Індекс функції F_i визначає номер обраного виходу і відповідає десятковому еквіваленту вхідного коду. Вихід, на якому з'являється керуючий сигнал, називається **активним**. Якщо значення сигналу на активному виході відображається лог. 1, то на решті пасивних виходів встановлюється лог. 0. Двійковий код, який вміщує завжди тільки одну одиницю, а інші — нулі, називається **унітарним**. Тому дешифратор є перетворювачем вхідного позиційного коду в унтарний вихідний код.

У дешифраторах в інтегральному виконанні стан активного виходу часто відображається значенням лог. 0, а на інших пасивних виходах устанавлюється лог. 1. Функціонування повного дешифратора з інверсними виходами представляється системою виду:

$$\begin{aligned} L_0 &= X_n \vee X_{n-1} \vee \dots \vee X_2 \vee X_1; \\ L_1 &= X_n \vee X_{n-1} \vee \dots \vee X_2 \vee \bar{X}_1, \\ &\dots \dots \dots \\ L_{m-1} &= \bar{X}_n \vee \bar{X}_{n-1} \vee \dots \vee \bar{X}_2 \vee \bar{X}_1, \end{aligned}$$

де L_0, L_1, \dots, L_{m-1} — вихідні логічні функції, що є макстермами (конституенти 0) n змінних.

Індекс функції L_i визначає номер вибраного виходу і відповідає десятковому еквіваленту вхідного коду. Між двома видами вихідних функцій існує простий зв'язок. $F_i = \bar{L}_i$.

Дешифратори класифікують за такими ознаками:

- способом структурної організації — одноступеневі (лінійні) і багатоступеневі, в тому числі пірамідальні та прямокутні (матричні);
- форматом вхідного коду — двійкові, двійково-десяткові;
- розрядністю коду, який дешифрується — 2, 3, ..., n ;
- формою подачі вхідного коду — з однофазними і парафазними входами;
- кількістю виходів — повні й неповні дешифратори;
- видом вхідних стробуючих сигналів — в прямому або інверсному значеннях;
- типом використовуваних логічних елементів — І, НЕ, ЧИ, НЕ І, НЕ ЧИ і т.д.

До основних характеристик дешифратора відносять: число ступенів (каскадів) дешифрації, кількість використаних логічних елементів або мікросхем, загальне число входів логічних елементів, час дешифрації і споживану потужність. Умовні графічні позначення дешифраторів на електричних схемах показані на рис. 4.1.

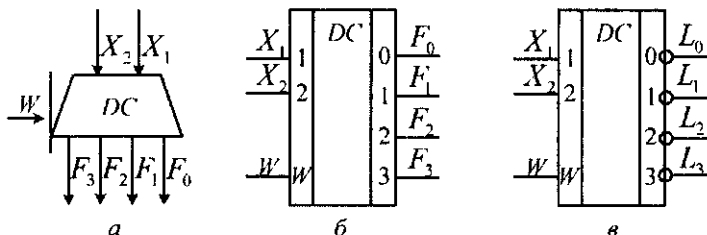


Рис. 4.1. Умовні графічні позначення дешифратора:

а — на функціональних схемах; *б, в* — на принципіальних схемах

Логічна функція дешифратора позначається буквами *DC* (decoder). Мітки лівого додаткового поля в умовному позначенні відображають десяткові ваги вхідних змінних, а мітки правого додаткового поля відповідають десятковим еквівалентам вхідних комбінацій двійкових змінних. У схему дешифраторів вбудовуються один або два стробуючих (дозволяючих) входи, наприклад, *W* (рис. 4.1, б). За допомогою сигналу на вході *W* визначається момент спрацювання дешифратора; крім того, вхід *W* використовується для нарощування розрядності вхідного коду. На практиці повний дешифратор на n входів і m виходів для стислості називають дешифратором "з n в m " або " $n \rightarrow m$ ". Наприклад, дешифратор "з 3 у 8" — активізується одна з восьми вихідних ліній.

В комп'ютерах дешифратори використовують для виконання таких операцій:

- дешифрації коду операції, записаного в регістр команд процесора, що забезпечує вибір потрібної мікропрограми;
- перетворення коду адреси операнда в команді в керуючі сигнали вибору заданої комірки пам'яті в процесі записування або читання інформації;
- забезпечення візуалізації на зовнішніх пристроях;
- реалізації логічних операцій та побудови мультіплексорів і демультіплексорів.

Використання дешифраторів для дешифрації коду операції і адреси операнда, розташованих в регістрі команд процесора, показано на рис. 4.2. Дешифрація коду операції в пристрої керування (ПК) визначає тип машинної команди. Дешифрація адреси операнда в оперативній пам'яті (ОП) забезпечує доступ до вказаної комірки пам'яті для записування або зчитування даних.

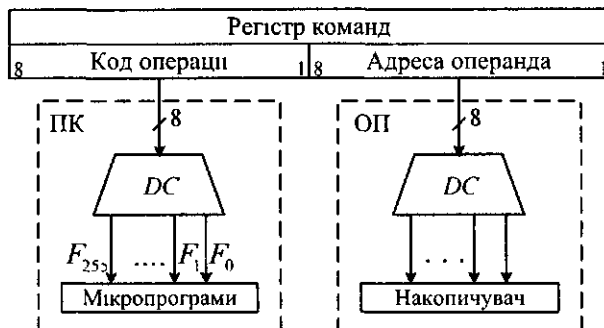


Рис. 4.2. Ілюстрація використання дешифраторів

4.1.2. Лінійні дешифратори на два входи і чотири виходи

У лінійному дешифраторі "з n в m " кожна вихідна функція F , реалізується повністю окремим n -вхідним логічним елементом при використанні парафазного вхідного коду. Логіка роботи повних дешифраторів на два входи X_1, X_2 і чотири прямих виходи F_0, F_1, F_2, F_3 і чотири інверсних виходи L_0, L_1, L_2, L_3 наведена в табл. 4.1 і 4.2 відповідно.

Таблиця 4.1

X_2	X_1	F_0	F_1	F_2	F_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Таблиця 4.2

X_2	X_1	L_0	L_1	L_2	L_3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

За даними табл. 4.1 отримують систему логічних функцій в ДДНФ

$$F_0 = X_2 X_1; F_1 = \overline{X_2} X_1; F_2 = X_2 \overline{X_1}; F_3 = X_2 X_1 \overline{X_2} \quad (4.1)$$

Для лінійного дешифратора зі стробуючим входом W система рівнянь (4.1) набуває вигляду

$$F_0 = \overline{X_2} \overline{X_1} W; F_1 = \overline{X_2} X_1 W; F_2 = X_2 \overline{X_1} W; F_3 = X_2 X_1 W \quad (4.2)$$

Схеми лінійних дешифраторів на основі рівнянь (4.1) і (4.2) показані на рис. 4.3

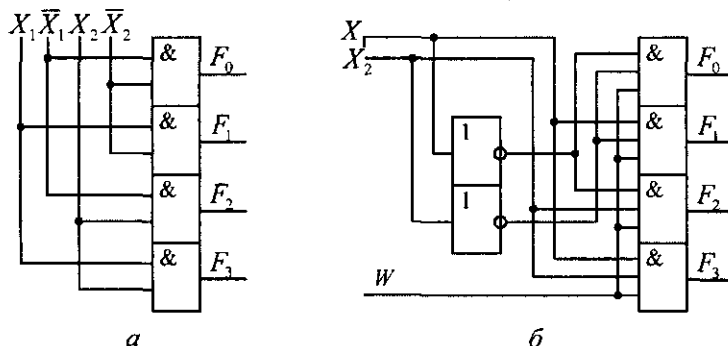


Рис. 4.3. Схеми лінійних дешифраторів на елементах І:

а — з парафазними входами; б — з однофазними входами і стробуванням

У схемі, зображеній на рис. 4.3, б використовується однофазний вхідний код, оскільки інверсії змінних утворюються елементами НЕ. Якщо сигнал на стробуючому вході $W = 0$, то робота дешифратора блокується — на всіх виходах устанавлюються логічні нулі незалежно від значень вхідних змінних. При $W = 1$ дешифратор функціонує згідно з табл. 4.1.

За даними табл. 4.2 записується система логічних функцій в ДКНФ:

$$L_0 = X_2 \vee X_1; L_1 = X_2 \vee \overline{X_1}; L_2 = \overline{X_2} \vee X_1; L_3 = \overline{X_2} \vee \overline{X_1}. \quad (4.3)$$

Схема лінійного дешифратора з парафазним вхідним кодом та інверсними виходами, побудована згідно з рівнянням (4.3) на елементах ЧИ, показана на рис 4.4, а.

Для лінійного дешифратора із стробуючим W входом система керування (4.3) набуває вигляду:

$$\begin{aligned} L_0 &= X_2 \vee X_1 \vee W; & L_1 &= X_2 \vee \overline{X_1} \vee W; \\ L_2 &= \overline{X_2} \vee X_1 \vee W; & L_3 &= \overline{X_2} \vee \overline{X_1} \vee W. \end{aligned} \quad (4.4)$$

Схема лінійного дешифратора на основі рівнянь (4.4) показана на рис. 4.4, б.

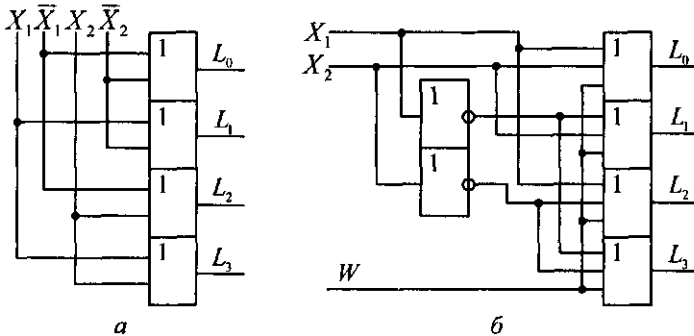


Рис. 4.4. Схема лінійних дешифраторів на елементах ЧИ:

а — з парафазними входами; б — з однофазними входами і стробуванням

Тут використовується однофазний вхідний код, оскільки інверсії змінних утворюються елементами НЕ. Якщо сигнал на стробуючому вході $W = 1$, то робота дешифратора блокується — на всіх виходах встановлюються лог. 1 незалежно від значень вхідних змінних. При $W = 0$ дешифратор функціонує згідно з табл. 4.2.

4.1.3. Пірамідальні дешифратори

У пірамідальному дешифраторі число ступенів на одиницю менше розрядності вхідного коду, тобто $K = n - 1$. В усіх ступенях використовуються тільки двовходові логічні елементи. На першому ступені використовуються лінійні дешифратори на два входи і чотири виходи. Число логічних елементів у кожному ступені дорівнює $M_i = 2^{i-1}$, де $i = 1, 2, \dots, k$. Це означає, що кожен подальший ступінь має в два рази більше елементів, ніж попередній. Вихід елемента i -го ступеня підключається до входів тільки двох елементів $(i+1)$ -го ступеня.

Пірамідальна структура для реалізації повного дешифратора "з 3 в 8" описується системою мінтермів виду:

$$F_0 = \overline{X_3} \overline{X_2} \overline{X_1}; F_1 = \overline{X_3} \overline{X_2} X_1; \dots F_7 = X_3 X_2 X_1.$$

Схема пірамідального дешифратора з парафазним вхідним кодом на три входи і вісім виходів показана на рис.4.5.

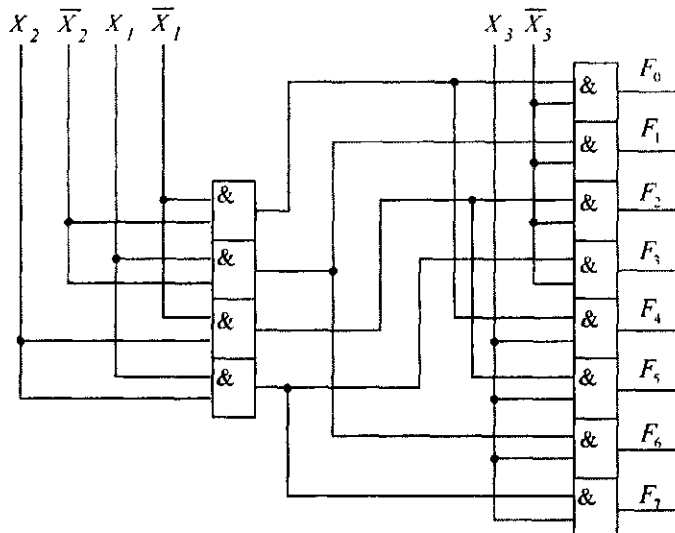


Рис. 4.5. Схема пірамідального дешифратора на три входи і вісім виходів

На першому ступені дешифруються змінні X_2 і X_1 , на другому ступені добавляється розряд X_3 . При більшому числі розрядів дешифрованого коду, наприклад, $n > 10$, дешифратор в $n/4$ економічніше лінійного.

Основним недоліком пірамідального дешифратора є велике число ступенів, що суттєво збільшує час дешифрації коду.

4.1.4. Прямокутні дешифратори

Прямокутний дешифратор будується за двоступеневою схемою. При цьому вхідний код розбивається на дві групи по $n/2$ розрядів при парному n ; при непарній розрядності групи вміщують нерівне число змінних. Дві групи змінних декодуються на першому ступені двома повними лінійними (можливо і пірамідальними) дешифраторами, а на другому ступені формуються вихідні функції.

Умовно вважають, що один з дешифраторів першого ступеня формує адреси рядків матриці, а другий — адреси стовпчиків матриці. На перетині ліній рядків і стовпчиків підключається $m = 2^n$ двохходових схем збігу, які утворюють другий, вихідний ступінь дешифратора. При парному n матриця вентилів квадратна, при непарному n — прямокутна. Тому такі дешифратори називаються матричними або прямокутними.

Запишемо систему вихідних функцій повного дешифратора "з 4 в 16" у вигляді таких скорочених значень:

$$\begin{aligned}
 F_0 &= a_0 b_0; & F_4 &= a_1 b_0; & F_8 &= a_2 b_0; & F_{12} &= a_3 b_0; \\
 F_1 &= a_0 b_1; & F_5 &= a_1 b_1; & F_9 &= a_2 b_1; & F_{13} &= a_3 b_1; \\
 F_2 &= a_0 b_2; & F_6 &= a_1 b_2; & F_{10} &= a_2 b_2; & F_{14} &= a_3 b_2; \\
 F_3 &= a_0 b_3; & F_7 &= a_1 b_3; & F_{11} &= a_2 b_3; & F_{15} &= a_3 b_3.
 \end{aligned}
 \tag{4.5}$$

де введені дворозрядні функції a_i і b_i , які реалізуються дешифраторами рядків і стовпчиків відповідно:

$$\begin{aligned} b_0 &= \bar{X}_2\bar{X}_1; & b_1 &= \bar{X}_2X_1; & b_2 &= X_2\bar{X}_1; & b_3 &= X_2X_1; \\ a_0 &= \bar{X}_4\bar{X}_3; & a_1 &= \bar{X}_4X_3; & a_2 &= X_4\bar{X}_3; & a_3 &= X_4X_3. \end{aligned} \quad (4.6)$$

Схема прямокутного дешифратора на основі рівнянь (4.5) і (4.6) показана на рис. 4.6.

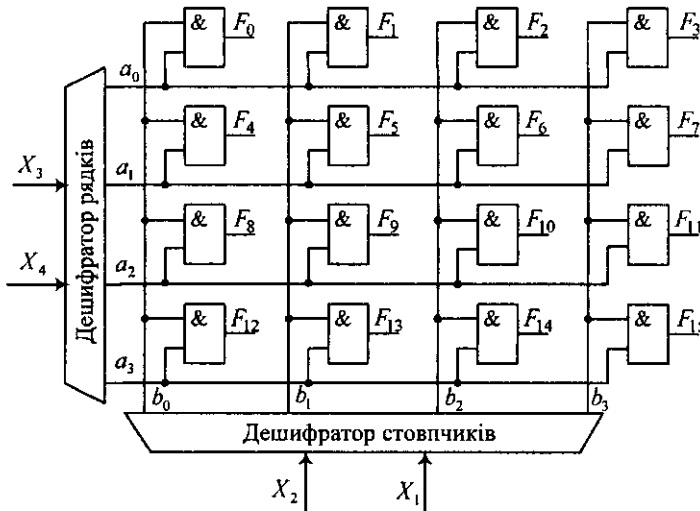


Рис. 4.6. Схема прямокутного дешифратора

При великому числі розрядів прямокутний дешифратор майже у $n/2$ рази економічніший лінійного і у два рази — пірамідального.

4.1.5. Багатоступеневі дешифратори. Каскадування дешифраторів

Принцип побудови багатоступеневих дешифраторів полягає у послідовному розбитті вхідного багаторозрядного коду до отримання у кожній групі двох-трьох розрядів. Як приклад на рис. 4.7 показано розбиття коду, який дешифрується для $n = 10$ і $n = 13$. Після цього багатоступенева схема дешифратора зображується у вигляді з'єднання ряду лінійних схем.

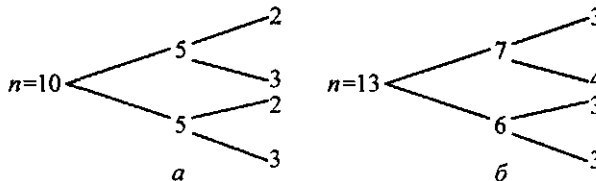


Рис. 4.7. Розбиття вхідного коду, який дешифрується на групи:

а — при $n = 10$; б — при $n = 13$

Під каскадуванням (нарощуванням) розуміють спосіб з'єднання дешифраторів у вигляді мікросхем середнього ступеня інтеграції для одержання більшої розрядності

вхідного коду. З'єднання двох трирозрядних дешифраторів для декодування чотирирозрядного коду показано на рис. 4.8.

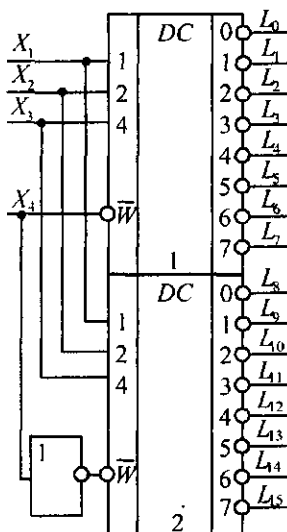


Рис. 4.8. Каскадування дешифраторів

Вхідні змінні X_1 , X_2 і X_3 подаються паралельно на входи обох дешифраторів: змінна X_4 подається безпосередньо на вхід стробування \overline{W} першого дешифратора, через інвертор — на вхід стробування другого дешифратора.

Ця каскадна схема працює так. Якщо значення старшого розряду вхідного коду $X_4 = 0$, то в роботу включається перший дешифратор з інверсними вісьмома виходами L_0, \dots, L_7 , при цьому другий дешифратор блокуваний (вимкнений) і на його виходах L_8, \dots, L_{15} встановлюються високі рівні. При $X_4 = 1$ блокується перший дешифратор і включається в роботу друга мікросхема.

Таким чином, через наявність стробуючого входу два трирозрядних дешифратори утворюють схему дешифрації чотирирозрядного коду.

4.1. Шифратори

4.2.1. Загальна характеристика шифратора

Шифратором називається функціональний вузол комп'ютера, призначений для перетворення вхідного m -розрядного унітарного коду у вихідний n -розрядний двійковий позиційний код. Двійкові шифратори виконують функцію, обернену функції дешифратора. При активізації однієї з вхідних ліній дешифратора на його виходах формується код, який відображає номер активного входу. Повний двійковий шифратор має $m = 2^n$ входів і n виходів. Умовні графічні позначення шифраторів на схемах показані на рис. 4.9.

Функція шифратора позначається буквами CD (coder). Входи шифратора нумеруються послідовними десятковими цифрами $0, 1, \dots, m-1$, а позначки виходів відображають ваги вихідних двійкових змінних $1, \dots, 2^{n-1}$.

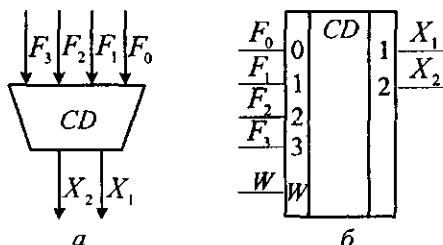


Рис. 4.9. Умовні графічні позначення шифратора: а — на функціональних схемах; б — на принципових схемах

У цифрових пристроях шифратори використовуються для таких операцій: перетворення унітарного вхідного коду у вихідний двійковий позиційний код; введення десяткових даних з клавіатури; показання старшої одиниці в слові; передачі інформації між різними пристроями при обмеженому числі ліній зв'язку.

4.2.2. Пріоритетний шифратор клавіатури

Одне з основних застосувань шифратора — введення даних з клавіатури, наприклад, десяткових цифр. Натискання клавіші з десятковою цифрою 0, 1, ..., 9 мають приводити до передачі в цифровий пристрій двійково-десяткового коду цієї цифри. Для цього використовується неповний шифратор "з 10 в 4".

Шифратори, які при одночасному натисканні декількох клавіш виробляють код тільки старшої цифри, називаються **пріоритетними**. Пріоритетні шифратори, які призначені для пошуку старшої (лівої) одиниці в слові та формування на виході двійкового номера шуканого розряду, називаються **показчиками старшої одиниці**. Їх застосовують у пристроях нормалізації чисел з плаваючою комою, в системах з пріоритетним обслуговуванням запитів на переривання роботи комп'ютера.

Логіка роботи пріоритетного шифратора на вісім входів наведена в табл. 4.3, де прийняті такі позначення: $\overline{F_0}, \overline{F_1}, \dots, \overline{F_7}$ — вхідні інверсні сигнали, записані в порядку зростання пріоритету: $\overline{F_0}$ — найнижчий, $\overline{F_7}$ — найвищий; $\overline{X_3}, \overline{X_2}, \overline{X_1}$ — вихідний інверсний позиційний код; \overline{W} — сигнал стробування, \overline{P} — функція, яка вказує на надходження вхідного сигналу; \overline{V} — функція, яка вказує на відсутність вхідних сигналів.

Таблиця 4.3

\overline{W}	$\overline{F_7}$	$\overline{F_6}$	$\overline{F_5}$	$\overline{F_4}$	$\overline{F_3}$	$\overline{F_2}$	$\overline{F_1}$	$\overline{F_0}$	$\overline{X_3}$	$\overline{X_2}$	$\overline{X_1}$	\overline{P}	\overline{V}
1	x	x	x	x	x	x	x	x	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0	1	1	1	0	1
0	1	1	1	1	1	1	0	x	1	1	0	0	1
0	1	1	1	1	1	0	x	x	1	0	1	0	1
0	1	1	1	1	0	x	x	x	1	0	0	0	1
0	1	1	1	0	x	x	x	x	0	1	1	0	1
0	1	1	0	x	x	x	x	x	0	1	0	0	1
0	1	0	x	x	x	x	x	x	0	0	1	0	1
0*	0	x	x	x	x	x	x	x	0	0	0	0	1

У табл. 4.3 значення вхідних змінних праворуч від діагоналі, утвореної цифрами 1, не повинні визначати вихідний код (вони позначені хрестиком). Це пояснюється тим, що сигнал з більшим пріоритетом блокує запити з меншими пріоритетами.

Із табл. 4.3 отримуємо вирази для вихідного коду шифратора $\overline{X_3}, \overline{X_2}, \overline{X_1}$ і функцій \overline{V} та \overline{P} , які відповідно визначають відсутність інформаційних сигналів на всіх виходах та наявність сигналу хоч би на одному вході. Для спрощення виразів використовуємо тотожність $F_i \vee \overline{F_i} F_k = F_i \vee F_k$ та закони де Моргана:

$$\overline{X_3} = \overline{W} \vee W Y_1;$$

$$\overline{X_2} = \overline{W} \vee W Y_1 \overline{F_3} \overline{F_2} \vee W \overline{F_7} \overline{F_6} F_5 \vee W \overline{F_7} \overline{F_6} F_4;$$

$$\overline{X_1} = \overline{W} \vee W Y_1 \overline{F_3} F_2 \vee W Y_1 \overline{F_3} \overline{F_1} \vee W \overline{F_7} \overline{F_6} \vee W F_7 F_5 F_4;$$

$$\overline{P} = \overline{W} \vee W Y_1 Y_2; \quad \overline{V} = W \vee \overline{Y_1} Y_2 W \vee \overline{Y_1} \vee \overline{Y_2};$$

$$Y_1 = \overline{F_7} \overline{F_6} \overline{F_5} \overline{F_4}; \quad Y_2 = \overline{F_3} \overline{F_2} \overline{F_1} \overline{F_0}.$$

На основі цих виразів побудована (рис. 4.10) схема пріоритетного шифратора "8 → 3".

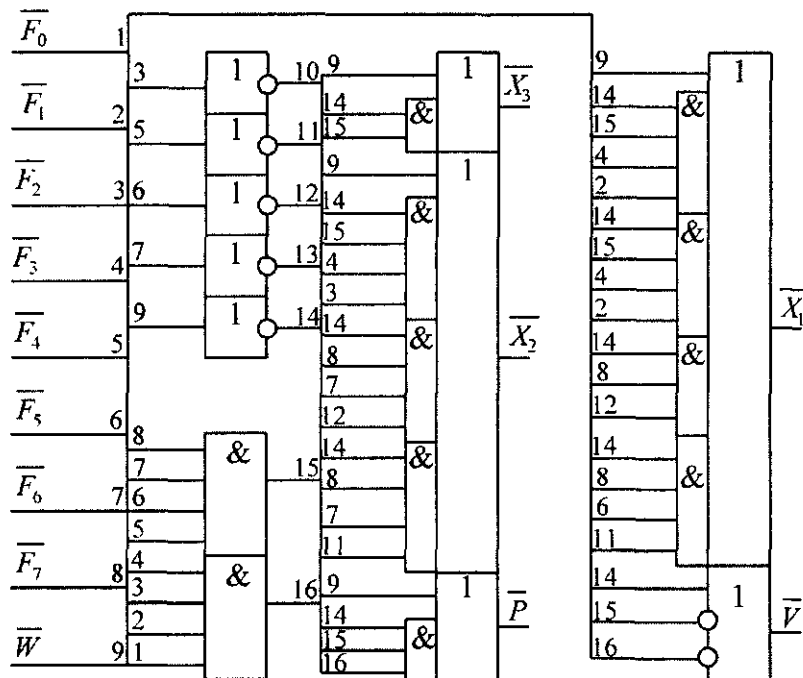


Рис. 4.10. Схема пріоритетного шифратора "8 → 3"

При $\overline{W} = 1$ робота схеми блокується і незалежно від сигналів на входах маємо на інверсних виходах: $\overline{X_3}\overline{X_2}\overline{X_1} = 111$, $\overline{P} = 1$, $\overline{V} = 1$. Якщо, наприклад, $\overline{F_6} = 0$ і $\overline{F_2} = 0$, то схема формує на виходах код номера входу із старшим пріоритетом: $\overline{X_3}\overline{X_2}\overline{X_1} = 001$ або в прямому коді $X_3X_2X_1 = 110_2 = 6_{10}$. Активний стан виходу відображається значеннями функцій $\overline{P} = 0$ і $\overline{V} = 1$, які передаються в процесор, а також використовуються при каскадуванні шифраторів. Схема, зображена на рис. 4.10, є аналогом шифратора K555IB1.

4.2.3. Каскадування шифраторів

Каскадування шифраторів використовується для збільшення розрядності вхідного слова. Схема каскадування двох восьмивходових шифраторів K555IB1 для пріоритетного обслуговування 16-розрядного слова $\overline{F_{15}} - \overline{F_0}$ показана на рис. 4.11.

Розряди $\overline{F_{15}} - \overline{F_8}$ старшого байта вхідного слова поступають на перший шифратор CD1, а розряди $\overline{F_7} - \overline{F_0}$ молодшого байта подаються на другий шифратор CD2. Вхід $\overline{F_{15}}$ має найвищий пріоритет, а $\overline{F_0}$ — найнижчий. Інформаційні виходи обох шифраторів об'єднуються за допомогою логічних елементів ЧИ, утворюючи трирозрядний інверсний код $\overline{Z_3}\overline{Z_2}\overline{Z_1}$. Значення старшого розряду $\overline{Z_4}$ забезпечується безпосередньо сигналом $\overline{P_1}$.

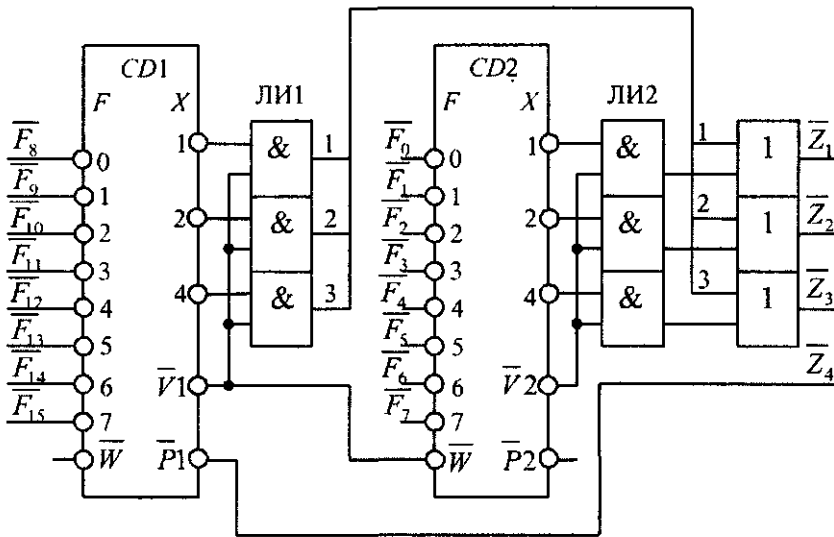


Рис. 4.11. Каскадування двох восьмивходових пріоритетних шифраторів

Інформація з виходів першого шифратора подається на входи елементів ЧИ за допомогою схем збігу ЛІІ1 при $\bar{V}1 = 1$ (мікросхема $CD1$ сприймає вхідні дані, а $CD2$ — блокувана). Інформація з виходів другого шифратора подається на входи елементів ЧИ за допомогою схем збігу ЛІІ2 при $V2 = 1$ (мікросхема $CD2$ сприймає вхідні дані, $CD1$ — блокується).

Схема працює так: коли на вході є активний сигнал із старшого байта вхідного слова, наприклад, $\bar{F}_{14} = 0$ ($\bar{P}1 = 0$, $\bar{V}1 = 1$), то працює шифратор $CD1$ і на виходах елементів ЧИ формується інверсний код $\bar{Z}_4\bar{Z}_3\bar{Z}_2\bar{Z}_1 = 0001$, що відповідає прямому значенню $Z_4Z_3Z_2Z_1 = 1110 = 14_{10}$. Якщо активний вхідний сигнал відноситься до молодшого байта слова, наприклад, $\bar{F}_6 = 0$, то працює шифратор $CD2$ ($\bar{P}1 = 1$, $\bar{V}1 = 0$, $\bar{V}2 = 1$) і на інверсних виходах формується код $\bar{Z}_4\bar{Z}_3\bar{Z}_2\bar{Z}_1 = 1001$, що відповідає прямому числу $Z_4Z_3Z_2Z_1 = 0110 = 6_{10}$.

4.3. Мультиплексори і демультимплексори

4.3.1. Загальна характеристика мультиплексорів

Мультиплексором називається функціональний вузол комп'ютера, призначений для почергової комутації (перемикання) інформації від одного з n входів на загальний вихід. Номер конкретної вхідної лінії, що підключається до виходу в кожний такт машинного часу, визначається адресним кодом A_0, A_1, \dots, A_{m-1} . Зв'язок між числом інформаційних n і адресних m входів визначається співвідношенням $n = 2^m$. Таким чином, мультиплексор реалізує керовану передачу даних від кількох вхідних ліній в одну вихідну.

Умовне графічне позначення мультиплексорів показано на рис. 4.12. Функція мультиплексорів записується буквами MUX (multiplexor).

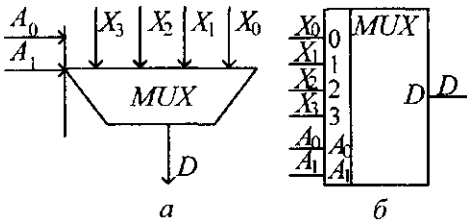


Рис. 4.12. Умовне позначення мультиплексора: а – на функціональних схемах; б – на принципових схемах

виходи внутрішнього дешифратора; X_0, X_1, X_2, X_3 – вхідна інформація; D – загальний інформаційний вихід.

Таблиця 4.4

A_1	A_0	F_0	F_1	F_2	F_3	D
0	0	1	0	0	0	F_0X_0
0	1	0	1	0	0	F_1X_1
1	0	0	0	1	0	F_2X_2
1	1	0	0	0	1	F_3X_3

На основі табл. 4.4. вираз для вихідної функції D можна представити з використанням виходів F_0-F_3 внутрішнього дешифратора у вигляді:

$$D = F_0X_0 \vee F_1X_1 \vee F_2X_2 \vee F_3X_3, \tag{4.7}$$

або з мінтермами адресного коду:

$$D = \overline{A_1} \overline{A_0} X_0 \vee \overline{A_1} A_0 X_1 \vee A_1 \overline{A_0} X_2 \vee A_1 A_0 X_3. \tag{4.8}$$

Схеми мультиплексорів, відповідні рівнянням (4.7) і (4.8), показані на рис. 4.13.

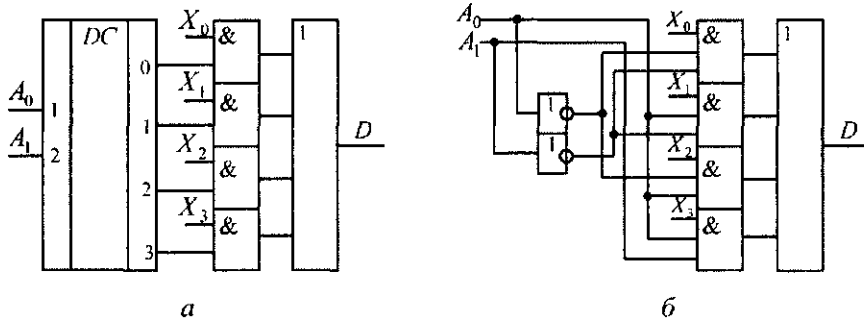


Рис. 4.13. Схеми мультиплексорів:

а – з внутрішнім дешифратором; б – з адресними мінтермами

При побудові схеми мультиплексорів з внутрішнім лінійним дешифратором потрібні логічні елементи з меншим числом входів, проте при цьому збільшується час встановлення вихідного сигналу. При виключенні дешифратора швидкодія підвищується, однак потрібні схеми збігу з більшим числом входів.

4.3.2. Каскадування мультиплексорів

В інтегральному виконанні мультиплексори випускають на чотири, вісім або шістнадцять входів. Каскадування дозволяє реалізувати комутацію довільного числа вхідних ліній на базі серійних мікросхем мультиплексорів меншої розрядності.

Приклад побудови схеми мультиплексора на 16 входів на основі типових чотиривходових мультиплексорів показаний на рис. 4.14.

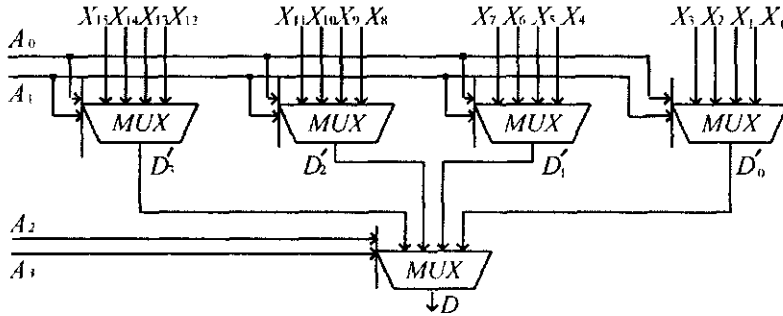


Рис. 4.14. Каскадування мультиплексорів

Молодші розряди адреси A_1, A_0 підключаються до адресних входів усіх мультиплексорів першого рівня, на виходах яких виробляються такі функції:

$$D'_0 = F_0 X_0 \vee F_1 X_1 \vee F_2 X_2 \vee F_3 X_3;$$

$$D'_1 = F_0 X_4 \vee F_1 X_5 \vee F_2 X_6 \vee F_3 X_7;$$

$$D'_2 = F_0 X_8 \vee F_1 X_9 \vee F_2 X_{10} \vee F_3 X_{11};$$

$$D'_3 = F_0 X_{12} \vee F_1 X_{13} \vee F_2 X_{14} \vee F_3 X_{15},$$

де F_0-F_3 — виходи внутрішніх дешифраторів: $F_0 = \overline{A_1} \overline{A_0}$; $F_1 = \overline{A_1} A_0$; $F_2 = A_1 \overline{A_0}$; $F_3 = A_1 A_0$; $X_{15}-X_0$ — вхідні змінні.

Старші розряди адреси A_3, A_2 подаються на адресні входи мультиплексора другого рівня, на виході якого формується остаточна функція

$$D = F'_0 D'_0 \vee F'_1 D'_1 \vee F'_2 D'_2 \vee F'_3 D'_3,$$

де внутрішні виходи дешифратора визначаються такими мінтермами:

$$F'_0 = \overline{A_3} \overline{A_2}; F'_1 = \overline{A_3} A_2; F'_2 = A_3 \overline{A_2}; F'_3 = A_3 A_2.$$

Нехай, наприклад, значення адреси $A_3 A_2 A_1 A_0 = 1011_2 = 11_{10}$. При цьому функція молодшої частини адреси приймає значення $F_3 = A_1 A_0 = 1$ і на виходах мультиплексорів першого рівня одночасно формуються сигнали $D'_0 = F_3 X_1$, $D'_1 = F_3 X_7$, $D'_2 = F_3 X_{11}$, $D'_3 = F_3 X_{15}$. Функція старшої частини адреси $F'_2 = A_3 \overline{A_2} = 1$ забезпечує передачу на вихід значення сигналу D'_2 , тобто

$$D = F'_2 D'_2 = \overline{A_3} A_2 A_1 A_0 X_{11}.$$

4.3.3 Реалізація логічних функцій мультиплексорами

За допомогою мультиплексорів реалізуються логічні функції з числом змінних m , що дорівнює розрядності адресного коду. Функція, що виконується, має бути представлена в ДДНФ. При цьому змінні поступають на адресні входи, а інформаційні

входи використовуються як настроювальні — на них подаються константи нуля і одиниці залежно від функції, яка реалізується.

Вихідна функція триадресного мультиплексора на вісім входів описується рівнянням:

$$D(A) = \bar{A}_2 \bar{A}_1 \bar{A}_0 X_0 \vee \bar{A}_2 \bar{A}_1 A_0 X_1 \vee \bar{A}_2 A_1 \bar{A}_0 X_2 \vee \bar{A}_2 A_1 A_0 X_3 \vee \\ \vee A_2 \bar{A}_1 \bar{A}_0 X_4 \vee A_2 \bar{A}_1 A_0 X_5 \vee A_2 A_1 \bar{A}_0 X_6 \vee A_2 A_1 A_0 X_7.$$

Якщо потрібно отримати логічну функцію з десятковими еквівалентами мінтермів 1, 3, 5 і 7, то на парні входи X_0, X_2, X_4 і X_6 необхідно подати константу "0", а на непарні X_1, X_3, X_5 і X_7 — константу "1". У результаті отримуємо (рис. 4.15):

$$D(A) = \bar{A}_2 \bar{A}_1 A_0 \vee \bar{A}_2 A_1 A_0 \vee A_2 \bar{A}_1 A_0 \vee A_2 A_1 A_0.$$

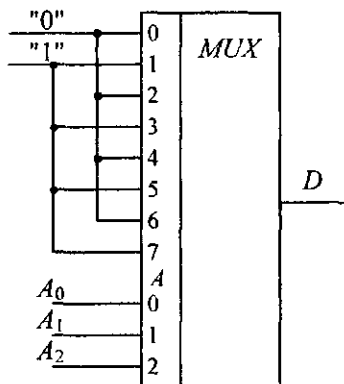


Рис. 4.15. Схема мультиплексора для реалізації логічної функції трьох змінних з десятковими еквівалентами мінтерміє 1,3,5 і 7

За допомогою додаткових логічних перетворень можна реалізувати логічні функції з числом змінних $m+1$, тобто на одиницю більше розрядності адресного коду мультиплексора.

Можливі й інші схемотехнічні застосування мультиплексорів. Мультиплексор може виконувати функції перетворювача n -розрядного двійкового паралельного коду в послідовний.

Для цього потрібно подати паралельний код на інформаційні входи мультиплексора і потім змінювати код на адреси в необхідній послідовності.

4.3.4 Мультиплексування шин

Мультиплексування шин — це почергове перемикання шин (груп ліній) від кількох джерел інформації до одного приймача. Такі мікрооперації реалізуються схемами на основі мультиплексорів одиночних ліній. При виборі кількості й типу мультиплексорів враховують:

- число комутованих шин дорівнює 2^m , де m — довжина адресного коду;
- i -й номер входу всіх мультиплексорів служить для підключення розрядів певної однієї шини.

Схема мультиплексора чотирьох $X(n), Y(n), Z(n)$ і $S(n)$ шин показана на рис. 4.16. Для її побудови потрібно n двоадресних чотиривходових мультиплексорів, де n — довільна розрядність шин, що комутуються.

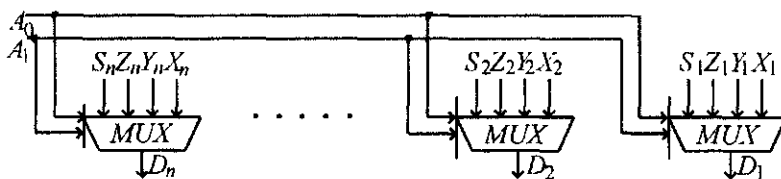


Рис. 4.16. Мультиплексор шин

4.3.5. Загальна характеристика демультимплексорів

Демультимплексором називається функціональний вузол комп'ютера, призначений для комутації (перемикання) сигналу з одного інформаційного входу D на один з n інформаційних виходів. Номер виходу, на який в кожний такт машинного часу передається значення вхідного сигналу, визначається адресним кодом $A_0, A_1, A_2, \dots, A_{m-1}$. Адресні входи m та інформаційні виходи n пов'язані співвідношенням $n = 2^m$ або $m = \log_2 n$.

Демультимплексор виконує функцію, обернену функції мультиплексора. Стосовно мультиплексорів і демультимплексорів користуються також терміном "селектори" даних.

В умовних графічних позначеннях (рис. 4.17) функція демультимплексора позначається буквами DMX .

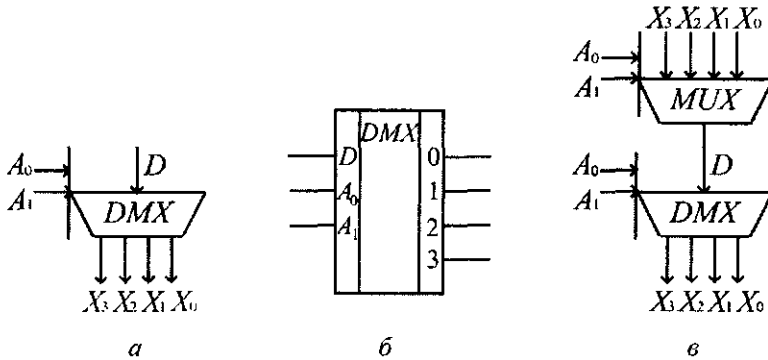


Рис. 4.17. Умовні графічні позначення демультимплексорів: а — на функціональних схемах; б — на принципових схемах; в — типове з'єднання з мультиплексором

Демультимплексори використовують для таких операцій:

- комутації як окремих ліній, так і багаторозрядних шин;
- перетворення послідовного коду в паралельний;
- реалізації логічних функцій та інших.

Демультимплексори часто позначають: " $1 \rightarrow n$ ".

Логіка роботи двоадресного демультимплексора на мові мікрооперацій наведена в табл. 4.5, де D — інформаційний вхід; F_0, F_1, F_2 і F_3 — виходи внутрішнього дешифратора адреси.

Таблиця 4.5

A_1	A_0	F_0	F_1	F_2	F_3	X_0	X_1	X_2	X_3
0	0	1	0	0	0	$F_0 D$	—	—	—
0	1	0	1	0	0	—	$F_1 D$	—	—
1	0	0	0	1	0	—	—	$F_2 D$	—
1	1	0	0	0	1	—	—	—	$F_3 D$

За даними табл. 4.5 запишемо систему рівнянь для інформаційних виходів:

$$\begin{aligned} X_0 &= F_0 D = \bar{A}_1 \bar{A}_0 D; & X_1 &= F_1 D = \bar{A}_1 A_0 D; \\ X_2 &= F_2 D = A_1 \bar{A}_0 D; & X_3 &= F_3 D = A_1 A_0 D; \end{aligned} \quad (4.9)$$

На основі рівнянь (4.9) побудовані схеми демультимплексорів із внутрішнім дешифратором (рис. 4.18, а) і з поєднанням адресних і вхідних змінних на тривходових елементах І (рис. 4.18, б).

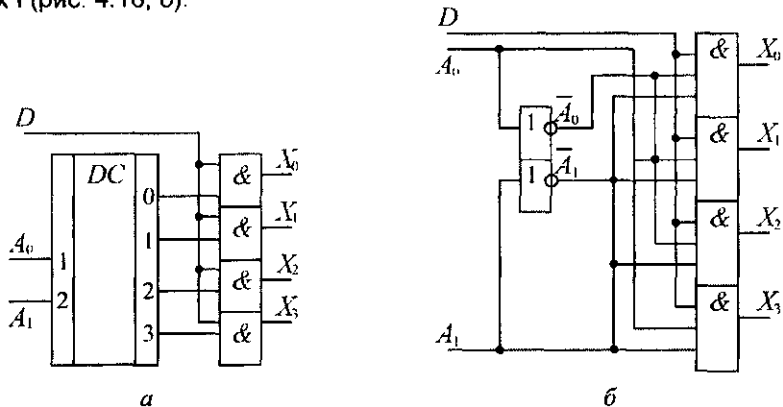


Рис. 4.18. Схема демультимплексорів:

а — з внутрішнім дешифратором; б — з поєднанням адресних і вхідних змінних

Схема демультимплексора з поєднанням адресних і вхідних змінних забезпечує високу швидкодію, проте вимагає застосування логічних елементів з більшим числом входів.

4.3.6. Каскадування демультимплексорів

Каскадування дозволяє реалізувати комутацію одного вхідного сигналу на довільне число вихідних ліній на базі серійних мікросхем меншої розрядності. Нехай потрібно реалізувати демультимплексування вхідного сигналу на n вихідних ліній, що визначаються m -розрядним адресним кодом, на базі типових мікросхем меншої розмірності виду " $1 \rightarrow n_1$ ".

Для цього потрібно використати $L = n/n_1$ типових демультимплексорів з числом адресних входів $m_1 = \log_2 n_1$ кожен. Число старших адресних розрядів, що дорівнює різниці $m - m_1$, використовується додатковим "ведучим" демультимплексором, який розташовується у першому рівні схеми каскадування. Ведучий демультимплексор визначає по черговому увімкненню одного з L демультимплексорів мікросхем другого рівня. Каскадування демультимплексорів виду " $1 \rightarrow 4$ " для реалізації комутатора " $1 \rightarrow 16$ " показано на рис. 4.19.

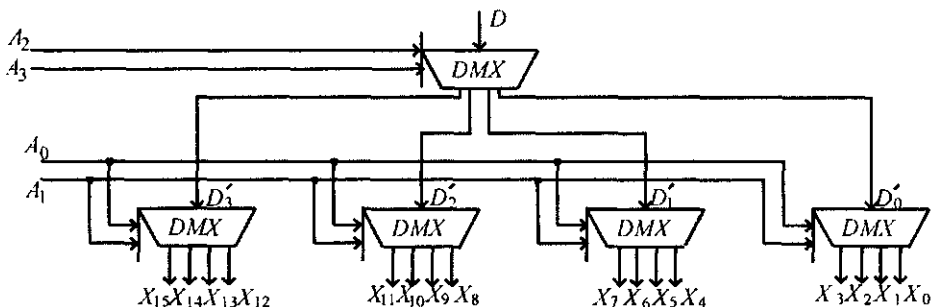


Рис. 4.19. Каскадування демультимплексорів

Нехай для схеми (рис. 4.19) адресний код $A_3A_2A_1A_0 = 1010$ і значення вхідного сигналу $D = 1$. Тоді на виході ведучого демультимплексора $D'_2 = A_3 \bar{A}_2 D = 1$, а на інших виходах встановлюються нульові значення. Одиначне значення сигналу D'_2 передається на вихід X_{10} веденого демультимплексора згідно зі співвідношенням

$$X_{10} = A_1 \bar{A}_0 D'_2 = A_3 \bar{A}_2 A_1 \bar{A}_0 D.$$

Демультимплексори не випускають як самостійні вироби на інтегральних мікросхемах. Функцію демультимплексора звичайно реалізують на дешифраторах, що мають входи стробування (дешифратори-демультимплексори).

4.3.7. Демультимплексування шин

Під демультимплексуванням шин розуміється почергове перемикання груп ліній від одного джерела інформації до багатьох приймачів. Такі мікрооперації реалізуються звичайно на основі демультимплексорів одиночних ліній. При виборі кількості і типу демультимплексорів враховують:

- число шин, які комутуються, дорівнює 2^m , де m — довжина адресного коду;
- кількість демультимплексорів, які використовуються, визначається розрядністю n шин, які демультимплексуються;
- адресні входи всіх мультиплексорів паралельно об'єднуються.

Схема демультимплексора вхідної шини $D(n)$ на чотири вихідні шини $X(n)$, $Y(n)$, $Z(n)$ і $S(n)$ показана на рис. 4.20.

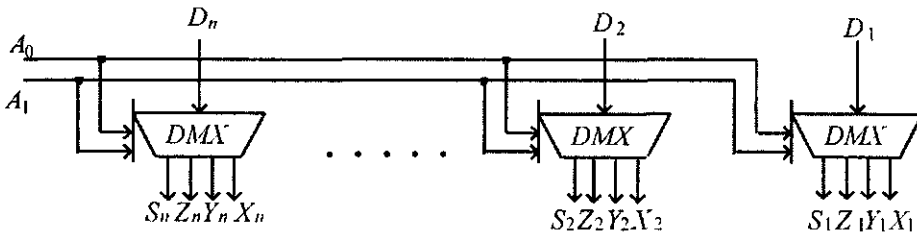


Рис. 4.20. Демультимплексор шин

Типове включення мультиплексорів і демультимплексорів для комутації вхідних і вихідних шин n -розрядних регістрів A , B , C і D показано на рис. 4.21.

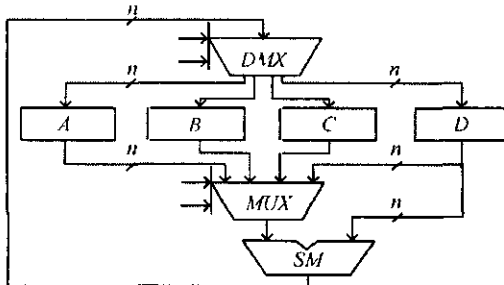


Рис. 4.21. Типова комутація вхідних і вихідних шин регістрів АЛП

У АЛП така комутація забезпечує використання як першого операнда суматора вмісту будь-якого регістра і запису результату операції в будь-який регістр, вказаний мікропрограмою команди, що виконується.

4.4. Схеми порівняння і контролю

4.4.1. Загальна характеристика схем порівняння

Схемою порівняння (компаратором) називається функціональний вузол комп'ютера, призначений для вироблення ознак відношень між двійковими словами (числами). Ознаки відношень записуються у вигляді

$$F_i = A * K \quad \text{або} \quad F_{i, A * K} \quad \text{або} \quad F_{A * K},$$

$$F_i = A * B \quad \text{або} \quad F_{i, A * B} \quad \text{або} \quad F_{A * B},$$

де A і B — двійкові або двійково-десяткові числа, K — двійкова константа, i — номер відношення (часто пропускається), $*$ — операція відношення вигляду $=, \neq, <, >, \leq, \geq$ і т. ін., F_i — функція, що задає результат відношення лог 1 — якщо відношення виконується, тобто істинне, і лог 0 — якщо відношення не виконується, тобто помилкове. Функція компаратора позначається буквами *COMP* (*comparator*) або знаками $=$.

Основними відношеннями вважаються "рівне" $F_{A=B}$, "більше" $F_{A>B}$ і "менше" $F_{A<B}$. Часто схеми, що реалізують відношення $F_{A=B}$ або $F_{A>B}$, називають схемами порівняння "на більше" або "на менше". Маючи в своєму розпорядженні основні ознаки відношень, можна на їхній основі отримати ряд додаткових ознак, наприклад

$$F_{A \neq B} = \overline{F_{A=B}}, \quad F_{A \leq B} = \overline{F_{A>B}}; \quad F_{A \leq B} = F_{A=B} \vee F_{A<B}$$

Ознаки відношення використовуються як логічні умови (повідомляючі сигнали) в мікропрограмах, командах передачі керування, а також у пристроях контролю і діагностики. Після виконання кожної команди в машині автоматично формуються ознаки результатів операції. Ці ознаки, які називаються **прапорами** (прапорцями), вміщуються в спеціальний регістр прапорів. До прапорів звичайно відносять ознаки нульового результату, переповнення розрядної сітки, знак результату, наявність перенесень із старшого розряду суматора, парне або непарне число одиниць в результаті та ін.

Зазначимо, що формування і використання ознак (прапорців) — це основна відмінність комп'ютера від калькулятора. Тільки за допомогою прапорців машина приймає рішення про хід обчислювального процесу, тобто володіє інтелектуальними властивостями.

4.4.2. Схеми порівняння слів з константою

Приймемо, що потрібно отримати ознаки відношень двійкового слова $A = A_2 A_1 A_0$ з наступними заданими константами

$$F_1 = (A=000), \quad F_2 := (A=111) \quad \text{і} \quad F_3 = (A \leq 011)$$

На основі табл. 4.6 значення ознак відношення слова A з константами запишуться у вигляді

$$F_1 = \overline{A_2} \overline{A_1} \overline{A_0}; \quad F_2 = A_2 A_1 A_0, \quad F_3 = \overline{A_3} \quad (4.10)$$

Схема порівняння слова з константою згідно з виразами (4.10) показана на рис. 4.22

Таблиця 4.6

A_2	A_1	A_0	F_1	F_2	F_3
0	0	0	1	0	1
0	0	1	0	0	1
0	1	0	0	0	1
0	1	1	0	0	1
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	0	1	0

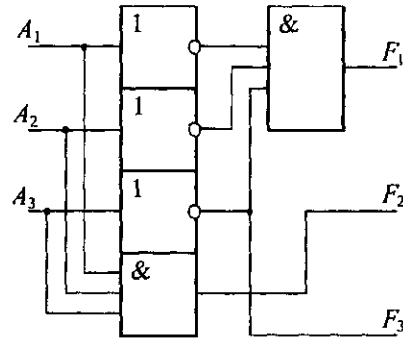


Рис. 4.22. Схеми порівняння слова з константою

4.4.3. Схеми порівняння двійкових слів A і B

Багаторозрядні двійкові слова рівні, коли одночасно попарно рівні всі їхні розряди, тобто $A(n) = B(n)$, якщо $A_i = B_i$, $i = 1, 2, \dots, n$. На основі табл. 4.7, яка задає умову рівності r_i двох i -х розрядів A і B , отримаємо:

$$r_i = \overline{A_i} \overline{B_i} \vee A_i B_i = \overline{A_i \oplus B_i} = \overline{M_i}, \quad (4.11)$$

де M_i — функція додавання по модулю два ("Виключальне ЧИ").

Схемна реалізація функції (4.11) показана на рис. 4.23.

Таблиця 4.7

A_i	B_i	r_i
0	0	1
0	1	0
1	0	0
1	1	1

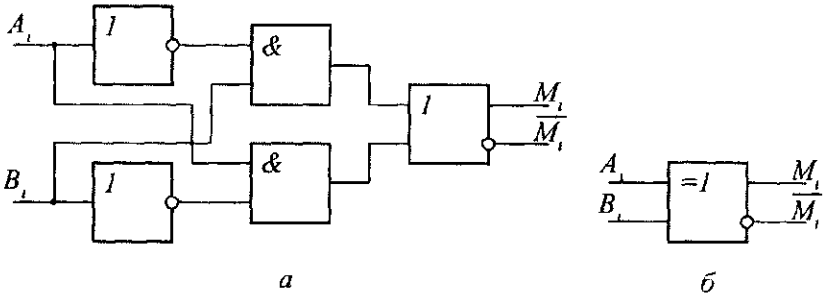


Рис. 4.23. Елемент "Виключальне ЧИ": а — схема; б — умовне позначення

Ознака рівності двох n -розрядних слів $P_{A=B}$ визначається логічним добутком порозрядних умов r_i :

$$F_{A=B} = r_n r_{n-1} \dots r_1 = \overline{M_n} \cdot \overline{M_{n-1}} \dots \overline{M_1}. \quad (4.12)$$

Схеми порівняння двох чотирирозрядних слів A і B згідно з виразом (4.12) показана на рис. 4.24. Схеми вміщують чотири логічних елементи "Виключальне ЧИ" і один кон'юнктор.

При великій розрядності слів, які порівнюються, можна на першому рівні отримати ознаки для чотирирозрядних груп і на другому рівні реалізувати загальний прапор логічним множенням групових ознак. Наприклад, при

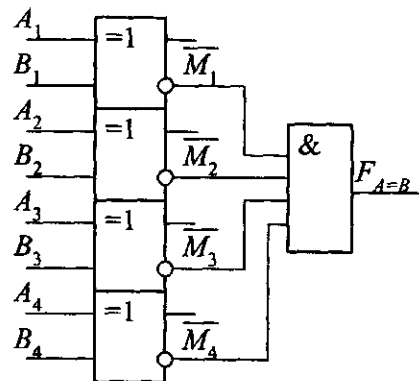


Рис. 4.24. Схеми порівняння двох чотирирозрядних слів A і B

розрядності порівнюваних слів $n = 16$ отримуємо чотири групові ознаки порівняння

$$F_{A=B}^{1,4}; F_{A=B}^{5,8}; F_{A=B}^{9,12}; F_{A=B}^{13,16},$$

де верхні індекси означають номери розрядів у групах. Тоді ознака порівняння двох 16-розрядних слів запишеться у вигляді:

$$F_{A=B} = F_{A=B}^{1,4} \cdot F_{A=B}^{5,8} \cdot F_{A=B}^{9,12} \cdot F_{A=B}^{13,16}.$$

Схема порівняння двох 16-розрядних слів показана на рис. 4.25, а.

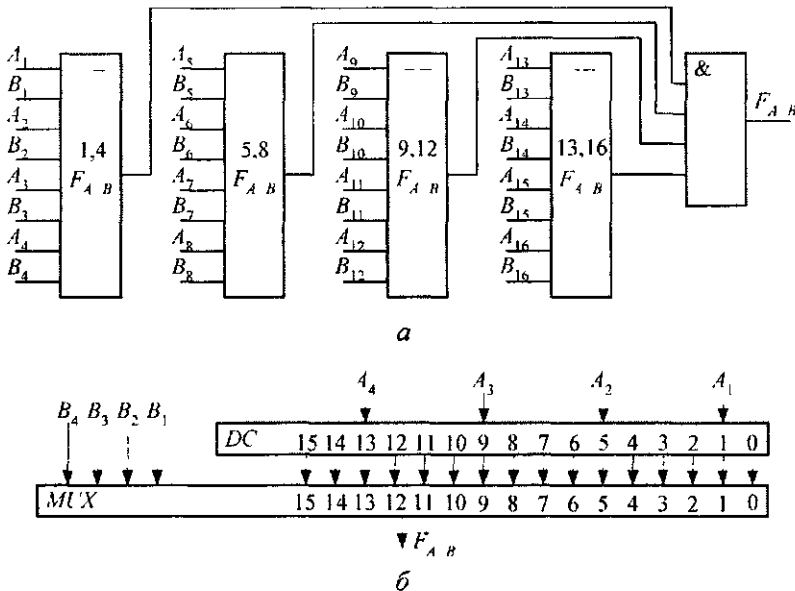


Рис. 4.25. Схема порівняння двох слів на рівність:

а — групова структура; б — на основі дешифратора і мультиплексора

Порівняння може бути реалізоване і на інших схемотехнічних принципах. Схема порівняння двох чотирирозрядних чисел A і B на основі дешифратора і мультиплексора показана на рис. 4.25, б. Дешифратор виробляє одиничне значення сигналу на тому виході, номер якого також визначається десятковим еквівалентом вхідного коду. Наприклад, при $A_4 A_3 A_2 A_1 = 0111$ логічна одиниця з'явиться на виході з номером сім. Мультиплексор підключає до виходу той вхід, номер якого також визначається десятковим еквівалентом вхідної комбінації. Якщо $B_4 B_3 B_2 B_1 = 0111$, то дозволяється проходження на вихід сигналу із сьомого входу. Таким чином, якщо слова A і B рівні, то формується прапор $F_{A=B} = 1$.

4.4.4. Схеми порівняння двох слів “на більше”

Схема порівняння двох слів A і B “на більше” за абсолютним значенням виробляє ознаку $F_{A>B}$ і будується за наступним алгоритмом:

- аналіз нерівності слів A і B виконується послідовно в напрямку від старших розрядів до молодших;
- молодші розряди включаються в аналіз в тому випадку, коли старші розряди рівні (еквівалентні);

• для отримання ознаки $P_{A>B}$ будується диз'юнктивна сума порозрядних умов.

Логіка порівняння розрядів A і B наведена в табл. 4.8, де C_i — ознака $A_i > B_i$; r_i — умова підключення до аналізу сусідніх молодших розрядів обох слів.

На основі табл. 4.8 отримуємо такі вирази:

$$C_i = A_i \bar{B}_i; r_i = \bar{A}_i \bar{B}_i \vee A_i B_i = \bar{A}_i \oplus \bar{B}_i = \bar{M}_i. \quad (4.13)$$

З урахуванням виразу (4.13) і алгоритму аналізу функцію ознаки $F_{A>B}$ представляємо у вигляді:

$$F_{A>B} = C_n \vee r_n C_{n-1} \vee \dots \vee r_n r_{n-1} \dots r_2 C_1. \quad (4.14)$$

Для порівняння двох чотирирозрядних слів "на більше" ознаку нерівності згідно з виразом (4.14) представляємо таким чином:

$$\begin{aligned} F_{A>B} &= C_4 \vee r_4 C_3 \vee r_4 r_3 C_2 \vee r_4 r_3 r_2 C_1 = \\ &= A_4 \bar{B}_4 \vee \bar{M}_4 A_3 \bar{B}_3 \vee \bar{M}_4 \bar{M}_3 A_2 \bar{B}_2 \vee \bar{M}_4 \bar{M}_3 \bar{M}_2 A_1 \bar{B}_1. \end{aligned} \quad (4.15)$$

Схема порівняння "на більше" двох чотирирозрядних слів A і B згідно із співвідношенням (4.15) показана на рис. 4.26.

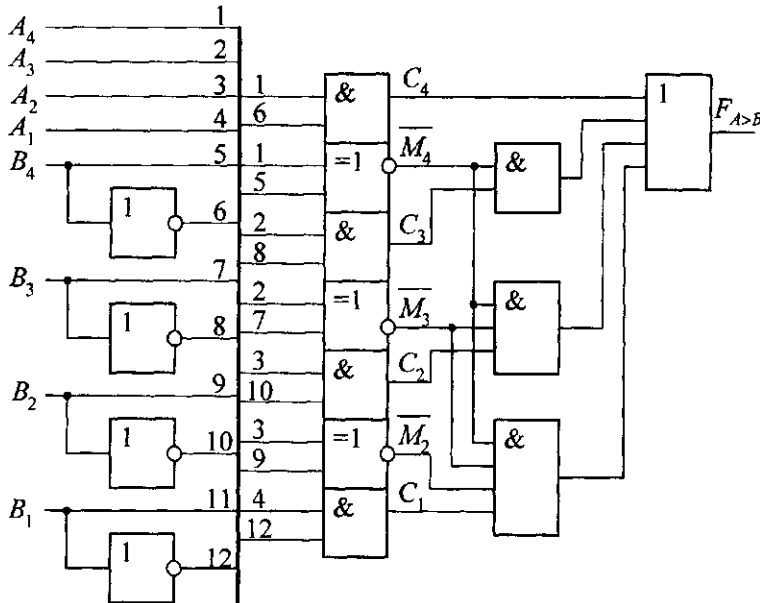


Рис. 4.26. Схема порівняння двох слів "на більше"

4.4.5. Багаторозрядні схеми порівняння "на більше"

При реалізації схем порівняння багаторозрядних слів "на більше" виникають технічні труднощі, пов'язані з необхідністю використання вентилів з великою кількістю входів. Тому слова, що порівнюються, розбиваються на групи, які складаються, наприклад, з чотирьох розрядів. Кожна група виробляє свою ознаку нерівності $F_{A>B}^i$ і умову підключення до аналізу $M_{i,r}^i$ молодшої групи згідно з виразом (4.15) і схемою

(рис 4 26) Наприклад, для $n = 16$ маємо чотири групи, які об'єднуються згідно із співвідношенням

$$F_{A>B} = F_{A>B}^4 \vee M_{гр}^4 F_{A>B}^3 \vee M_{гр}^4 M_{гр}^3 F_{A>B}^2 \vee M_{гр}^4 M_{гр}^3 M_{гр}^2 F_{A>B}^1 \quad (4 16)$$

де $F_{A>B}^4$ — прапор порівняння 'на більше' в найстаршій групі з розрядами $A_{16} - A_{13}$, $B_{16} - B_{13}$, $M_{гр}^4 = M_{16}M_{15}M_{14}M_{13}$ — умова для підключення до аналізу сусідньої молодшої групи, $F_{A>B}^3$ — прапор порівняння "на більше" у групі з розрядами $A_{12} - A_{9}$, $B_9 - B_9$, $M_{гр}^3 = M_{12}M_{11}M_{10}M_9$ — умова аналізу молодшої групи $F_{A>B}^2$ — прапор порівняння "на більше" у групі з розрядами $A_8 - A_5$, $B_8 - B_5$, $M_{гр}^2 = M_8M_7M_6M_5$ — умова підключення молодшої групи, $F_{A>B}^1$ — прапор порівняння "на більше" у групі з розрядами $A_4 - A_1$, $B_4 - B_1$

Схема порівняння 'на більше' двох 16-розрядних слів A і B на основі рівняння (4 16) показана на рис 4 27

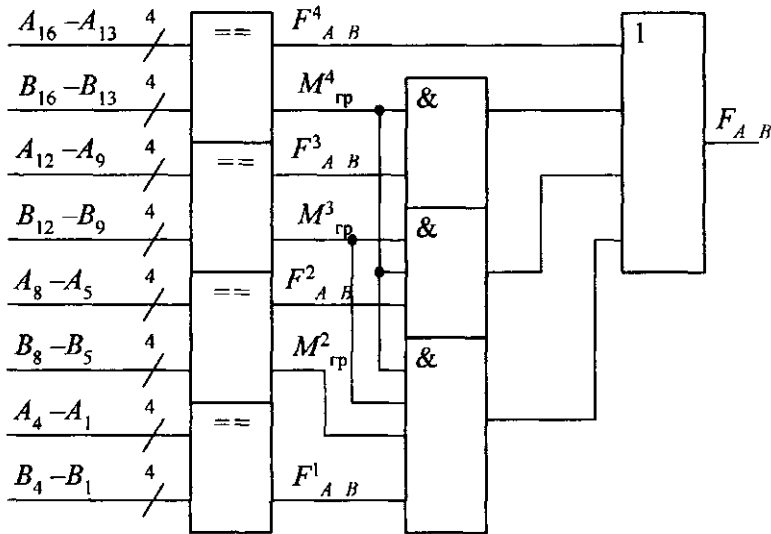


Рис 4 27 Схема порівняння "на більше" двох 16-розрядних слів

4.4.6. Застосування компараторів

Контроль (виявлення) і корекція (виправлення) результатів операцій є важливою умовою грамотної експлуатації машин. Контроль може бути програмним або апаратним. До апаратних методів відносяться дублювання операцій і відновлення вхідних сигналів.

Контроль операцій додавання методом дублювання реалізується двома однаковими суматорами (SM), на входи яких одночасно поступають доданки $A(n)$ і $B(n)$. Обидва результати $S_1(n)$ і $S_2(n)$ поступають на входи схеми порівняння (рис 4 28, а). Якщо обидва результати рівні, то на виході схеми порівняння значення ознаки $F_{S_1=S_2} = 1$ і помилок немає. При нульовому значенні ознаки операцію потрібно повторити або зупинити роботу ЕОМ.

Схема контролю методом відновлення вхідних сигналів показана на рис 4 28 б. Дворозрядне слово A_2A_1 декодується і значення унітарного коду з виходів дешифратора поступає на входи шифратора. При правильній роботі дешифратора і шифра-

тора вхідний код A_2A_1 має збігатися з вихідним кодом шифратора B_2B_1 . При цьому на виході схеми порівняння встановиться одиничне значення ознаки $F_{A B}$.

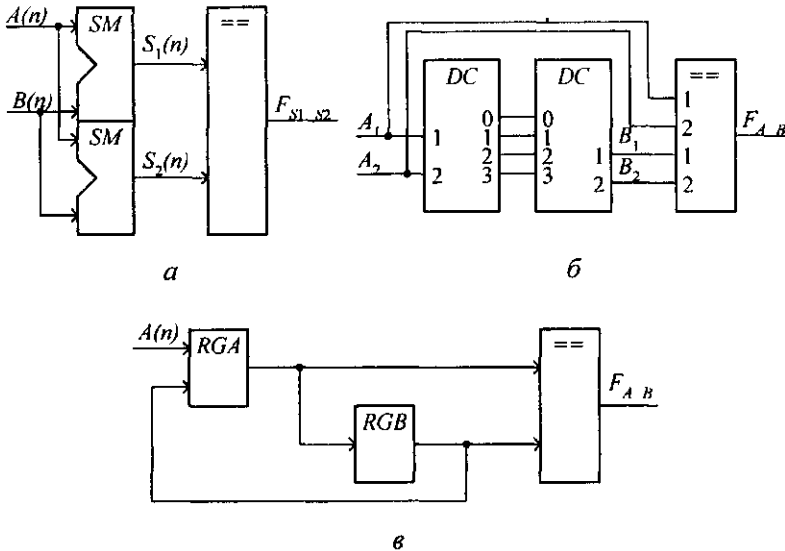


Рис. 4.28. Застосування схеми порівняння для контролю операцій

При передачі інформації з одного регістра в інший контроль правильності пересилки може здійснюватися порозрядним порівнянням вмісту цих двох регістрів. На рис. 4.28, в показаний один з варіантів контролю пересилок слів між регістрами. Після передачі інформації з регістра A в регістр B (або навпаки) проводиться порівняння їхнього вмісту. Якщо значення двох слів збігаються, то значення ознаки рівності набуває одиничного значення, інакше — виробляється сигнал помилки.

4.4.7. Загальна характеристика схем контролю парності

У комп'ютерах широко використовується **контроль парності** (синонім — за паритетом або відповідністю). Цей спосіб заснований на допущенні, що в двійковому числі найчастіше виникають одиничні помилки — втрата або поява зайвої одиниці. У обох випадках число одиниць зміниться на одну. Якщо двійкове число мало непарне число одиниць, то після одиничної помилки воно виявиться парним і навпаки.

На практиці контроль парності здійснюється таким чином. Для підвищення ефективності контролю двійкове слово розбивається на частини, як правило, байти. До кожного байта додається додатковий контрольний розряд. Вміст контрольного розряду залежить від вибраного способу контролю (за парністю або непарністю). При контролі за парністю значення контрольного розряду вибирається таким, щоб загальне число одиниць у байті й контрольному біті було парним. У цьому випадку значення контрольного (паритетного) біта визначається додаванням за модулем два значень розрядів байта (рис. 4.29, а):

$$F_{к.п} = A_1 \oplus A_2 \oplus A_3 \oplus A_4 \oplus A_5 \oplus A_6 \oplus A_7 \oplus A_8. \quad (4.17)$$

Внаслідок операції додавання за модулем два значень розрядів байта з парним числом одиниць одержуємо значення контрольного байта $F_{кп} = 0$. При додаванні за модулем два значень розрядів байта з непарним числом одиниць значення контрольного байта $F_{кп} = 1$.

$F_{кп}$	8	7	6	5	4	3	2	1
0	1	0	1	0	1	1	1	1
1	1	1	1	0	1	1	0	0

a

$F_{кн}$	8	7	6	5	4	3	2	1
1	1	0	1	0	1	1	1	1
0	1	1	1	0	1	1	0	0

б

Рис. 4.29. Контроль байта: *a* — за парністю; *б* — за непарністю

При контролі за непарністю значення контрольного біта вибирається з умови, щоб кількість одиниць у байті з урахуванням вмісту контрольного розряду була непарною. У цьому випадку значення контрольного біта набуває такого виразу:

$$F_{кп} = A_1 \oplus A_2 \oplus A_3 \oplus A_4 \oplus A_5 \oplus A_6 \oplus A_7 \oplus A_8 = \overline{F_{кп}}. \quad (4.18)$$

На практиці контроль непарності використовується частіше, оскільки фіксує повне пропання інформації. Контроль парності (непарності) передбачає формування значень контрольних розрядів до виконання операції та перевірку байта після виконання операції з урахуванням контрольних розрядів. Наприклад, при записуванні байта в пам'ять комп'ютера одночасно автоматично формується (генерується) значення його контрольного розряду. При зчитуванні байта, що зберігається, здійснюється додавання за модулем два значень його розрядів спільно з контрольним бітом згідно з визначеним способом контролю парності або непарності. Таким чином, контроль за паритетом вимагає використання додаткових розрядів.

Схеми, що забезпечують отримання значення контрольного розряду і перевірку двійкового числа за ознакою парності або непарності, називаються схемами контролю парності. Їх часто називають схемами згортки, схемами контролю за модулем два, схемами контролю за паритетом. Для отримання умови парності (4.17) потрібне складання за модулем два восьмирозрядного слова, що реалізується за допомогою ступінчатого включення двохходових елементів "Виключальне ЧИ":

- на першому рівні отримують функції $F_1 - F_4$:

$$F_1 = A_1 \oplus A_2; \quad F_2 = A_3 \oplus A_4; \quad F_3 = A_5 \oplus A_6; \quad F_4 = A_7 \oplus A_8; \quad (4.19)$$

- на другому і третьому рівнях реалізуються функції:

$$F_5 = F_1 \oplus F_2; \quad F_6 = F_3 \oplus F_4; \quad M = F_5 \oplus F_6. \quad (4.20)$$

Функція M згідно з виразами (4.19) і (4.20) набуває значення лог. 1 при непарному числі одиниць у вхідному байті та значення лог. 0 — при парному числі одиниць у вхідному байті.

Для задання ознаки контролю вводиться керуючий сигнал V , який разом з сигналом M поступає на входи схеми "Виключальне ЧИ" в четвертому рівні; на прямому й інверсному виходах цього рівня формуються пряме й інверсне значення контрольного розряду:

$$F = M \oplus V; \quad \overline{F} = \overline{M \oplus V}.$$

Логіка роботи схеми контролю, показаної на рис. 4.30, *a*, наведена в табл. 4.9.

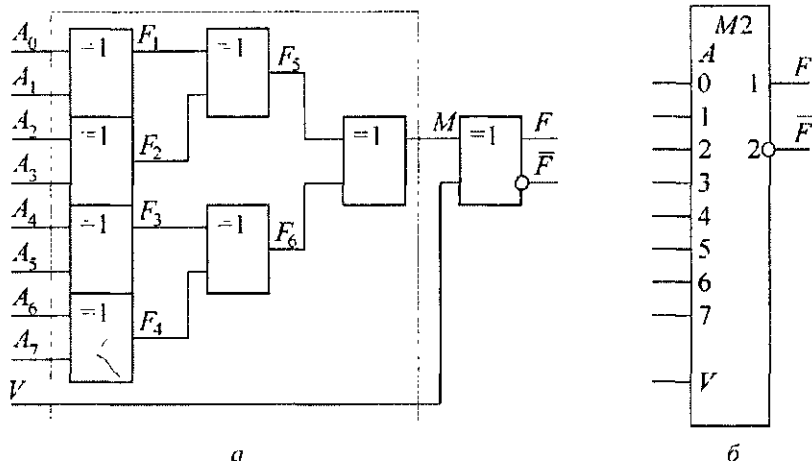


Рис. 4.30. Схеми контролю за парністю:

а — ступінчасте включення елемента "Виключальне ЧИ"; б — умовне позначення

Таблиця 4.9

Входи $A_8 - A_1$		V	F	\bar{F}
На входах:	Парне число одиниць	0	0	1
	Непарне число одиниць	0	1	0
На входах:	Парне число одиниць	1	1	0
	Непарне число одиниць	1	0	1

Із табл. 4.9 видно, що при $V = 0$ на виході F генерується значення контрольного розряду для контролю парності, при $V = 1$ на виході F генерується значення контрольного розряду для контролю непарності.

Як приклад схема контролю непарності пересилок байта від джерела інформації (ДІ) до приймача інформації (ПІ) показана на рис. 4.31.

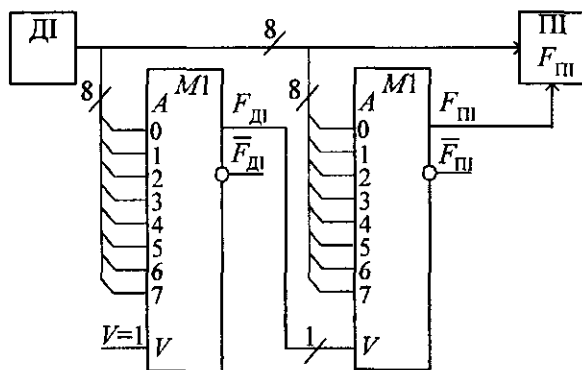


Рис. 4.31. Контроль пересилок байта

Схема контролю з боку джерела інформації виступає як генератор значення контрольного розряду непарності $F_{дл}$. Схема контролю з боку приймача інформації забезпечує додавання за модулем два значень розрядів визначеного байта спільно з визначеним контрольним бітом непарності. Прийом інформації можливий тільки при виконанні умови непарності $F_{пл} = 1$ з боку приймача.

4.5. Перетворювачі кодів

4.5.1. Загальна характеристика перетворювачів кодів

Перетворювачем коду називається функціональний вузол комп'ютера, призначений для перетворення двійкового коду з однієї форми в іншу.

Для подання інформації використовують різноманітні двійкові та двійково-десяткові коди: прямий, обернений, доповняльний і їхні модифікації, циклічний з лиском три та інші. Існує велика кількість кодів, які забезпечують

- простоту виконання арифметико-логічних операцій,
- зручність переведення чисел з десятикової системи в двійковий код,
- надійність виконання заданих алгоритмів функціонування і ефективний контроль результатів обчислень,
- зменшення апаратних витрат при побудові цифрових пристроїв.

Найбільш поширеними є прямий, обернений і доповняльний коди, які забезпечують представлення знака числа і заміну операції віднімання додаванням (табл. 4.10). До перетворювачів коду відносяться шифратори і дешифратори, однак за традицією ці функціональні вузли виділені в окремі самостійні класи.

Таблиця 4.10

Коди для додатних чисел				Коди для від'ємних чисел			
десятьковий	прямий	обернений	доповняльний	десятьковий	прямий	обернений	доповняльний
+0	0,000	0,000	0,000	-0	1,000	1,111	0,000
+1	0,001	0,001	0,001	-1	1,001	1,110	1,111
+2	0,010	0,010	0,010	-2	1,010	1,101	1,110
+3	0,011	0,011	0,011	-3	1,011	1,100	1,101
+4	0,100	0,100	0,100	-4	1,100	1,011	1,100
+5	0,101	0,101	0,101	-5	1,101	1,010	1,011
+6	0,110	0,110	0,110	-6	1,110	1,001	1,010
+7	0,111	0,111	0,111	-7	1,111	1,000	1,001

Прямий, обернений і доповняльний коди використовуються для записування знака числа, заміни операції віднімання чисел додаванням їхніх кодів, а також для визначення переповнення розрядної сітки. Для представлення знака числа у них відводиться знаковий розряд, який розташовується зліва від числа і відділяється комою. У знаковий розряд записується нуль — для позитивного числа і одиниця — для негативного.

4.5.2. Перетворювач прямого коду в обернений

У прямому двійковому коді $X_{\text{пр}} = X_{3\text{н}} X_{n-1} \dots X_1$ один розряд, звичайно старший, відображає знак числа, інші — значення цифрових розрядів; при цьому для додатного числа $X_{3\text{н}} = 0$, а для від'ємного $X_{3\text{н}} = 1$. Обернений код додатного двійкового числа збігається з прямим кодом, а для від'ємного числа цифрові розряди прямого коду інвертуються.

У процесі перетворення прямого коду в обернений значення знакового розряду $X_{3\text{н}}$ використовується як керуючий сигнал, що забезпечує отримання такого виразу:

$$Y_i = \bar{X}_{3\text{н}} X_i \vee X_{3\text{н}} \bar{X}_i = X_{3\text{н}} \oplus X_i, \quad (4.21)$$

де Y_i — значення i -го розряду оберненого коду; X_i — значення i -го розряду додатного вхідного числа ($X_{3\text{н}} = 0$); \bar{X}_i — значення i -го розряду від'ємного вхідного числа ($\bar{X}_{3\text{н}} = 1$).

Схема п'ятирозрядного перетворювача прямого коду в обернений, побудована на елементах "Виключальне ЧИ" відповідно до виразу (4.21), показана на рис. 4.32.

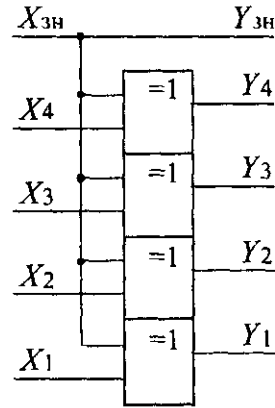


Рис. 4.32. Схема перетворювача прямого коду в обернений

4.5.3. Перетворювач прямого коду в доповняльний

Доповняльний код додатного двійкового числа збігається з його прямим і оберненим кодами. Доповняльний код від'ємного двійкового числа утворюється з його оберненого коду додаванням до молодшого розряду одиниці. Таким чином, операція перетворення прямого коду в доповняльний не є порозрядною і виконується значно складніше, ніж отримання оберненого коду.

Відповідність між прямим і доповняльним кодами на прикладі чотирьох цифрових розрядів (беззнакових) наведена в табл. 4.11.

Таблиця 4.11

Прямий код				Доповняльний код				Прямий код				Доповняльний код			
X_4	X_3	X_2	X_1	Y_4	Y_3	Y_2	Y_1	X_4	X_3	X_2	X_1	Y_4	Y_3	Y_2	Y_1
0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
0	0	0	1	1	1	1	1	1	0	0	1	0	1	1	1
0	0	1	0	1	1	1	0	1	0	1	0	0	1	1	0
0	0	1	1	1	1	0	1	1	0	1	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	0	1	0	0
0	1	0	1	1	0	1	1	1	1	0	1	0	0	1	1
0	1	1	0	1	0	1	0	1	1	1	0	0	0	1	0
0	1	1	1	1	0	0	1	1	1	1	1	0	0	0	1

Знаковий розряд прямого коду використовується як керуючий сигнал: якщо $X_{3\text{н}} = 0$, то вихідний код повторює значення вхідного; при $X_{3\text{н}} = 1$ реалізується перетворення згідно з табл. 4.11.

Карта Карно відповідно до табл. 4.11 для отримання мінімальних форм функцій перетворення прямого коду в доповняльний показана на рис. 4.33.

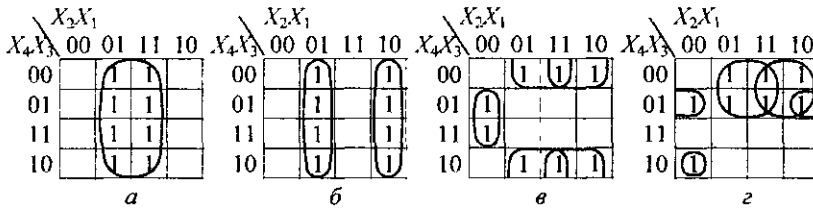


Рис. 4.33. Карта Карно для функцій перетворювача прямого коду в доповняльний:
а — Y_1 ; б — Y_2 ; в — Y_3 ; г — Y_4

На основі карт Карно з врахуванням знакового розряду $X_{3Н}$ прямого коду для функцій Y_1, Y_2, Y_3, Y_4 , що представляють виходи перетворювача, отримуємо:

$$\begin{aligned} Y_{3Н} &= X_{3Н}; & Y_1 &= X_1; & Y_2 &= X_2 \oplus X_1 X_{3Н}; \\ Y_3 &= X_3 \oplus (X_2 \vee X_1) X_{3Н}; & Y_4 &= X_4 \oplus (X_3 \vee X_2 \vee X_1) X_{3Н}. \end{aligned} \quad (4.22)$$

У загальному вигляді для Y_i справедливе рівняння:

$$Y_i = X_i \oplus (X_{i-1} \vee X_{i-2} \vee \dots \vee X_1) X_{3Н}. \quad (4.23)$$

Схема перетворювача прямого коду в доповняльний на основі виразів (4.22) і (4.23) показана на рис. 4.34, а. Даний перетворювач характеризується високою швидкістю. Час встановлення вихідного коду визначається трьома затримками поширення сигналу, однак в міру зростання номера розряду лінійно зростає й необхідне число входів використовуваних елементів ЧИ.

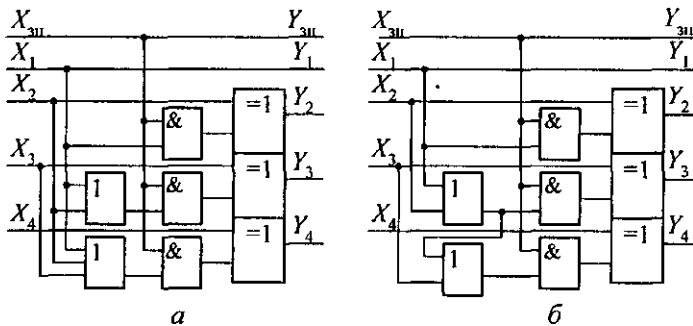


Рис. 4.34. Схеми перетворювачів прямого коду в доповняльний

Другий варіант схеми перетворювача (рис. 4.34, б) використовує тільки двовходові елементи ЧИ, при цьому диз'юнктивна сума змінних утворюється послідовним способом. У такій реалізації схема перетворювача спрощується, однак час встановлення вихідного коду істотно збільшується.

Практичне правило отримання доповняльного коду полягає в тому, що праворуч від першої одиниці (враховуючи і саму одиницю) в прямому коді числа значення розрядів — незмінні, а зліва від одиниці (крім знакового) — інвертуються. Наприклад, для прямого коду 10100100 доповняльним буде код 11011100.

Для перетворення в доповняльний код багаторозрядних двійкових чисел часто використовують переведення числа в обернений код і подальшого додавання одиниці до його молодшого розряду за допомогою суматора.

4.5.4. Перетворювач двійкових чисел у код Грея

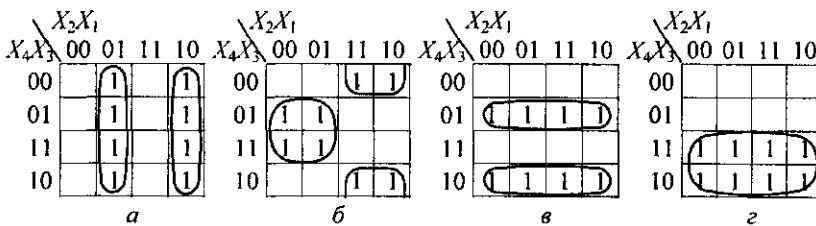
Код Грея утворений послідовністю двійкових чисел, в яких два будь-яких сусідніх числа відрізняються тільки одним розрядом (табл. 4.12). Перше і останнє числа вважаються сусідніми. Код Грея, який називають циклічним, відноситься до незважених двійкових кодів.

Достоїнствами коду Грея є: зручність кодування кутових переміщень; простота кодувальної логіки; скорочення часу перетворення у зв'язку зі зміною значення тільки одного розряду; висока ефективність захисту від збоїв. Недоліками коду Грея є ускладнення при виконанні арифметичних операцій і цифро-аналогових перетворень. Тому при необхідності код Грея перетворюють у двійковий код.

Таблиця 4.12

X_4	X_3	X_2	X_1	I_4	I_3	I_2	I_1	X_4	X_3	X_2	X_1	I_4	I_3	I_2	I_1
0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
0	0	0	1	0	0	0	1	1	0	0	1	1	1	0	1
0	0	1	0	0	0	1	1	1	0	1	0	1	1	1	1
0	0	1	1	0	0	1	0	0	0	1	1	1	1	1	0
0	1	0	0	0	1	1	0	1	1	0	0	1	0	1	0
0	1	0	1	0	1	1	1	1	1	0	1	1	0	1	1
0	1	1	0	0	1	0	1	1	1	1	0	1	0	0	1
0	1	1	1	0	1	0	0	1	1	1	1	1	0	0	0

За даними табл. 4.12 в клітинки карт Карно (рис. 4.35) внесено значення розрядів I_1, I_2, I_3, I_4 коду Грея.

Рис. 4.35. Карта Карно для коду Грея: а — I_1 , б — I_2 , в — I_3 , г — I_4

За допомогою карт Карно отримуємо такі вирази для розрядів коду Грея:

$$\begin{aligned}
 I_1 &= \bar{X}_2 X_1 \vee X_2 \bar{X}_1 = X_1 \oplus X_2; & I_2 &= \bar{X}_3 X_2 \vee X_3 \bar{X}_2 = X_2 \oplus X_3; \\
 I_3 &= \bar{X}_4 X_3 \vee X_4 \bar{X}_3 = X_3 \oplus X_4; & I_4 &= X_4.
 \end{aligned}
 \tag{4.24}$$

Схема перетворювача прямого коду в код Грея на основі співвідношень (4.24) показана на рис. 4.36.

За аналогічною методикою, використовуючи табл. 4.12 і нові заповнення карт Карно, отримуємо обернене перетворення коду Грея в прямий код:

$$X_1 = I_1 \oplus I_2 \oplus I_3 \oplus I_4; \quad X_2 = I_2 \oplus I_3 \oplus I_4; \quad X_3 = I_3 \oplus I_4; \quad X_4 = I_4.
 \tag{4.25}$$

Схема перетворення коду Грея в прямий код на основі співвідношень (4.25) показана на рис. 4.37.

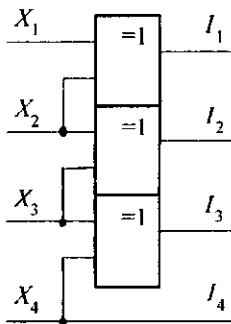


Рис. 4.36. Перетворювач прямого коду в код Грея

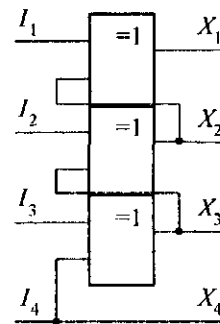


Рис. 4.37. Перетворювач коду Грея в прямий код

4.5.5. Перетворювач двійково-десяткових чисел в код семисегментного індикатора

Візуальне відображення двійково-десяткових чисел часто виконується за допомогою семисегментних індикаторів на основі електролюмінісcentних приладів, рідких кристалів або світлодіодних матриць. Кількість семисегментних індикаторів визначається розрядністю чисел, що відображаються на світловому табло — звичайно шість і більше десяткових цифр.

Десятковий код відображуваної цифри, що виводиться з комп'ютера, поступає на вхід двійково-десятьового перетворювача, виходи якого a, b, c, \dots, g підключаються до відповідних сегментів індикатора (рис. 4.38, а).

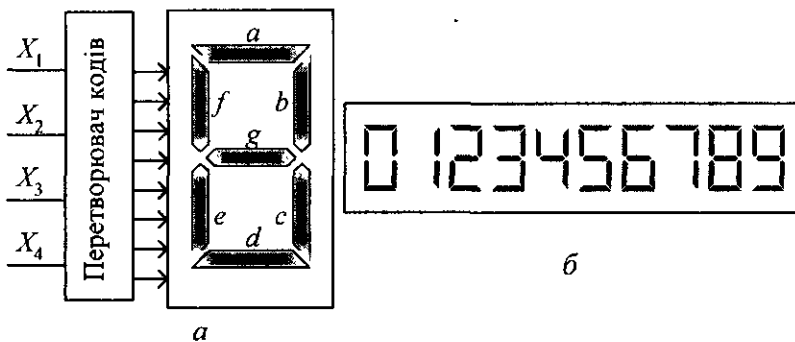


Рис. 4.38. Підключення перетворювача до індикатора (а) і відображення цифр (б)

Одиничне значення вихідного сигналу перетворювача викликає світіння сегмента, підключеного до цього виходу. Комбінації одиничних сигналів на виходах перетворювача утворюють зображення десяткової цифри в своєму розряді (рис. 4.38, б).

Відповідність між двійково-десятьовим числом і необхідними для відображення десяткової цифри наборами сегментів наведена в табл. 4.13.

Таблиця 4.13

X_4	X_3	X_2	X_1	a	b	c	d	e	f	g	X_4	X_3	X_2	X_1	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0	0	1	0	1	1	0	1	1	0	1	1
0	0	0	1	0	1	1	0	0	0	0	0	1	1	0	1	0	1	1	1	1	1
0	0	1	0	1	1	0	1	1	0	1	0	1	1	1	1	1	0	0	0	0	0
0	0	1	1	1	1	1	1	0	0	1	1	0	0	0	1	1	1	1	1	1	1
0	1	0	0	0	1	1	0	0	1	1	1	0	0	1	1	1	1	1	1	0	1

На основі даних табл. 4.13 і після їхньої мінімізації за допомогою карт Карно отримуємо систему логічних рівнянь для сегментів перетворювачів кодів

$$a = X_2 \vee X_4 \vee X_1 X_3 \vee \bar{X}_1 \bar{X}_3,$$

$$b = X_1 X_2 \vee \bar{X}_1 \bar{X}_2 \vee \bar{X}_3 \vee X_4,$$

$$c = X_1 \vee \bar{X}_2 \vee X_3 \vee X_4,$$

$$d = \bar{X}_1 X_2 \vee X_2 \bar{X}_3 \vee \bar{X}_1 \bar{X}_3 \vee X_1 \bar{X}_2 X_3 \vee X_4,$$

$$e = \bar{X}_1 X_2 \vee \bar{X}_1 \bar{X}_3,$$

$$f = \bar{X}_1 \bar{X}_2 \vee \bar{X}_1 X_3 \vee \bar{X}_2 X_3 \vee X_4,$$

$$g = \bar{X}_1 X_2 \vee X_2 \bar{X}_3 \vee \bar{X}_2 X_3 \vee X_4$$

Схема перетворювача двійково-десятичного коду в керуючі сигнали семисегментного індикатора показана на рис. 4.39

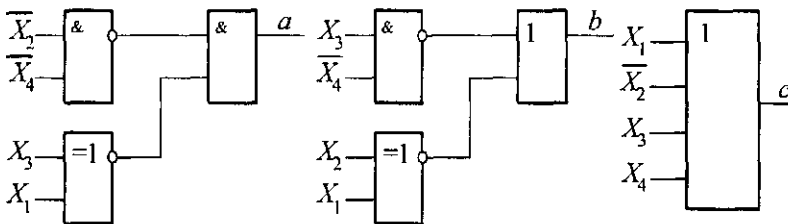


Рис. 4.39. Схема перетворювача коду "8421" в код семисегментного індикатора (для виходів a , b , c)

4.5.6. Двійково-десятикові перетворювачі

У комп'ютерах широко використовується двійково-десятикове кодування, в якому кожна десяткова цифра зображується чотирирозрядним двійковим кодом, тобто тетрадою двійкових символів. Число різних двійково-десятикових кодів визначається числом можливих комбінацій по десять із 16 комбінацій, які допускаються тетрадою. Десяткові числа можуть представлятися в кодї "з лишком 3", в кодї з вагою "5421" або "2421" (код Айкена) та ін.

Найбільш поширеним є код прямого заміщення "8421", в якому кожна десяткова цифра 0, 1, ..., 9 замінюється її двійковим еквівалентом 0000, 0001, ..., 1001. Такий код називають також зваженим D -кодом. Наприклад, число 729_{10} у двійково-десятиковому кодї записується у вигляді трьох тетрад 011100101001_{2-10} .

Для D -кодів розроблені машинні алгоритми операцій додавання, віднімання, множення, ділення та інші. Операції над десятковими числами (десятькова арифметика) входять до складу команд комп'ютерів різних класів.

Особливістю *D*-кодів є наявність десяти дозволених і шести заборонених комбінацій двійкових символів в тетраді. Поява забороненої комбінації при виконанні операцій над числами свідчить про виникнення помилки або ж про необхідність корекції результату.

У розрядній сітці машини двійково-десятьові коди представляються у формі з плаваючою або фіксованою комою. При цьому від'ємні числа відображаються в прямому, оберненому або доповняльному кодах. Для *D*-кодів не виконується умова отримання оберненого коду інвертуванням розрядів тетради.

Застосування *D*-кодів у комп'ютерах не вимагає виконання різного роду перетворень двійково-десятьових чисел у двійковий та навпаки. Наприклад, за допомогою шифратора забезпечується порівняно простий спосіб введення в машину десятикових цифр двійково-десятьовим кодом.

Значення однієї тетради *D*-коду, доповнення тетради до дев'яти "9-*D*" (обернений код) і до десяти "10-*D*" (доповняльний код), а також коди з "лишком 3" і з вагою "5421" наведені в табл. 4.14.

Таблиця 4.14

Код <i>D</i>				Код "9- <i>D</i> "				Код "10- <i>D</i> "				Код " <i>D</i> +3"				Код "5421"			
<i>X</i> ₄	<i>X</i> ₃	<i>X</i> ₂	<i>X</i> ₁	<i>Y</i> ₄	<i>Y</i> ₃	<i>Y</i> ₂	<i>Y</i> ₁	<i>F</i> ₄	<i>F</i> ₃	<i>F</i> ₂	<i>F</i> ₁	<i>Z</i> ₄	<i>Z</i> ₃	<i>Z</i> ₂	<i>Z</i> ₁	<i>E</i> ₄	<i>E</i> ₃	<i>E</i> ₂	<i>E</i> ₁
0	0	0	0	1	0	0	1	1	0	1	0	0	0	1	1	0	0	0	0
0	0	0	1	1	0	0	0	1	0	0	1	0	1	0	0	0	0	0	1
0	0	1	0	0	1	1	1	1	0	0	0	0	1	0	1	0	0	1	0
0	0	1	1	0	1	1	0	0	1	1	1	0	1	1	0	0	0	1	1
0	1	0	0	0	1	0	1	0	1	1	0	0	1	1	1	0	1	0	0
0	1	0	1	0	1	0	0	0	1	0	1	1	0	0	0	0	1	0	1
0	1	1	0	0	0	1	1	0	1	0	0	1	0	0	1	0	1	1	0
0	1	1	1	0	0	1	0	0	0	1	1	1	0	1	0	0	1	1	1
1	0	0	0	0	0	0	1	0	0	1	0	1	0	1	1	1	0	0	0
1	0	0	1	0	0	0	0	0	0	0	1	1	1	0	0	1	1	0	0

Систему мінімальних логічних виразів оберненого двійково-десятьового коду отримуємо за допомогою карт Карно (рис. 4.40), в які вносяться значення розрядів *Y*₁–*Y*₄ за даними табл. 4.14.

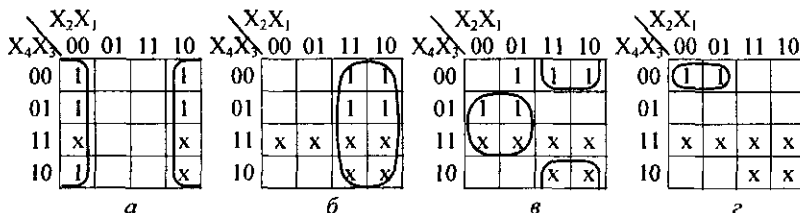


Рис. 4.40. Карты Карно для отримання оберненого *D*-коду;

знаком "х" помічені невизначені набори: а — *Y*₁, б — *Y*₂, в — *Y*₃, г — *Y*₄.

На основі карт Карно отримуємо систему рівнянь для розрядів *Y*₁ – *Y*₄ оберненого *D*-коду

$$Y_1 = \bar{X}_1; \quad Y_2 = X_2; \quad Y_3 = X_2 \oplus X_3; \quad Y_4 = \bar{X}_4 \bar{X}_3 \bar{X}_2 = \overline{X_4 \vee X_3 \vee X_2}. \quad (4.26)$$

Із виразів (4.26) випливає, що значення другого розряду тетради прямого і оберненого D -кодів збігаються. Схема перетворювача прямого D -коду в обернений на основі співвідношень (4.26) показана на рис. 4.41, а.

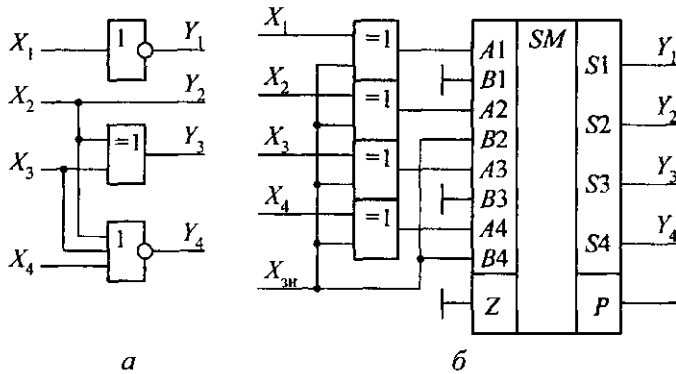


Рис. 4.41. Схеми перетворювачів D -коду в обернений:

а — на основі логічних рівнянь; б — з використанням елемента "виключальне ЧИ"

Можливий і інший спосіб побудови перетворювача D -коду в обернений: спочатку інвертуються цифри всіх тетрад (виходить код з лишком шість) і потім відбувається віднімання з кожної тетради числа мінус 0110, що еквівалентне додаванню в доповняльному коді плюс 1010 (без урахування перенесень між тетрадами). Наприклад, $A = -148$, прямий D -код $[A]_{пр} = 1\ 0001\ 0100\ 1000$; після інвертування маємо: $[A]_{пр}^* = 1\ 1110\ 1011\ 0111$; після додавання плюс 1010 до кожної тетради одержуємо значення оберненого D -коду: $[A]_{об} = 1\ 1000\ 0101\ 0001$.

Схема перетворювача на основі інвертування і корекції тетрад показана на рис. 4.41, б. В даній схемі інвертування значень розрядів тетрад здійснюється логічними елементами "виключальне ЧИ", а корекція реалізується типовою мікросхемою чотирирозрядного комбінаційного суматора. Доповняльний D -код тетради виходить з оберненого додаванням одиниці до молодшого розряду.

4.6. Двійкові суматори

4.6.1. Загальна характеристика суматорів

Суматором називається функціональний вузол комп'ютера, призначений для додавання двох n -розрядних слів (чисел). Операція віднімання замінюється додаванням слів в оберненому або доповняльному коді. Операції множення та ділення зводяться до реалізації багаторазового додавання та зсування. Тому суматор є важливою частиною арифметико-логічного пристрою. Функція суматора позначається буквами SM або Σ .

Суматор складається з окремих схем, які називаються **однорозрядними суматорами**; вони виконують усі дії з додавання значень однойменних розрядів двох чисел (операндів). Суматори класифікуються за такими ознаками:

- способом додавання — паралельні, послідовні та паралельно-послідовні;

- числом входів — напівсуматори, однорозрядні та багаторозрядні суматори,
- організацією зберігання результату додавання — комбінаційні, накопичувальні, комбіновані,
- організацією перенесення між розрядами — з послідовним, наскрізним, паралельним або комбінованим перенесеннями (з груповою структурою),
- системою числення — позиційні (двійкові, двійково-десяткові, трійкові) та непозиційні, наприклад, у системі залишкових класів,
- розрядністю (довжиною) операндів — 8-, 16-, 32-, 64-розрядні,
- способом представлення від'ємних чисел — в оберненому або доповняльному кодах, а також в їхніх модифікаціях,
- часом додавання — синхронні, асинхронні

У паралельних n -розрядних суматорах значення всіх розрядів операндів поступають одночасно на відповідні входи однорозрядних підсумовуючих схем. У послідовних суматорах значення розрядів операндів та перенесення, що запам'ятовувалися в минулому такті, поступають послідовно в напрямку від молодших розрядів до старших на входи одного однорозрядного суматора. В паралельно-послідовних суматорах числа розбиваються на частини, наприклад, байти, розряди байтів поступають на входи восьмирозрядного суматора паралельно (одночасно), а самі байти — послідовно, в напрямку від молодших до старших байтів з урахуванням запам'ятованого перенесення.

У комбінаційних суматорах результат операції додавання запам'ятовується в регістрі результату. В накопичувальних суматорах процес додавання поєднується із зберіганням результату. Це пояснюється використанням T -тригерів як однорозрядних схем додавання.

Організація перенесення практично визначає час виконання операції додавання. Послідовні перенесення схемно створюються просто, але є повільнодіючими. Паралельні перенесення схемно організуються значно складніше, але дають високу швидкодію.

Розрядність суматорів знаходиться в широких границях 4–16 — для мікро- та міні-комп'ютерів та 32–64 і більше — для універсальних машин.

Суматори з постійним інтервалом часу для додавання називаються **синхронними**. Суматори, в яких інтервал часу для додавання визначається моментом фактичного закінчення операції, називаються **асинхронними**. В асинхронних суматорах є спеціальні схеми, які визначають фактичний момент закінчення додавання і повідомляють про це в пристрій керування. На практиці переважно використовуються синхронні суматори.

Суматори характеризуються такими параметрами:

- швидкістю — часом виконання операції додавання t_{Σ} , який відраховується від початку подачі операндів до одержання результату, часто швидкодія характеризується кількістю додавання в секунду $F_{\Sigma} = 1/t_{\Sigma}$, тут маються на увазі операції типу реєстр-реєстр (тобто числа зберігаються в реєстрах АЛП),
- апаратними витратами вартість однорозрядної схеми додавання визначається загальним числом логічних входів використаних елементів, вартість багаторозрядного суматора визначається загальною кількістю використаних мікросхем,
- споживаною потужністю суматора

4.6.2. Однорозрядні суматори

Однорозрядним суматором називається логічна схема, яка виконує додавання значень i -х розрядів X_i та Y_i двійкових чисел з урахуванням перенесення Z_i з молодшого сусіднього розряду та виробляє на виходах функції результат S_i і перенесення P_i в старший сусідній розряд. На основі однорозрядних схем додавання на три входи та два виходи будуються багаторозрядні суматори будь-якого типу. Алгоритм роботи однорозрядного суматора відображається таблицею істинності (табл. 4.15).

На основі табл. 4.15 записується система логічних функцій для результату S_i та перенесення P_i у ДДНФ:

$$S_i = \bar{X}_i \bar{Y}_i Z_i \vee \bar{X}_i Y_i \bar{Z}_i \vee X_i \bar{Y}_i \bar{Z}_i \vee X_i Y_i Z_i; \quad (4.27)$$

$$P_i = \bar{X}_i Y_i Z_i \vee X_i \bar{Y}_i Z_i \vee X_i Y_i \bar{Z}_i \vee X_i Y_i Z_i. \quad (4.28)$$

Мінімізація функцій (4.27) та (4.28) за допомогою карт Карно показана на рис. 4.42.

Таблиця 4.15

X_i	Y_i	Z_i	S_i	P_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

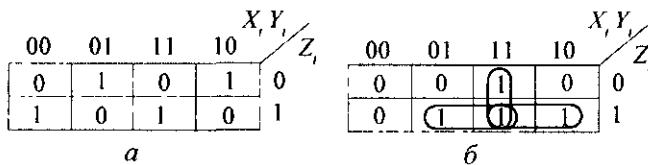


Рис. 4.42. Карты Карно для мінімізації функцій: а — S_i ; б — P_i .

Як видно з карт Карно, функція результату S_i не мінімізується, а функція P_i мінімізується зі зниженням рангу кон'юнкції та використовує тільки прямі значення змінних.

$$P_i = X_i Y_i \vee X_i Z_i \vee Y_i Z_i = X_i Y_i \vee (X_i \vee Y_i) Z_i. \quad (4.29)$$

При проектуванні комбінаційних однорозрядних суматорів враховують такі чинники:

- схема має характеризуватися регулярністю (подібністю) структури та мінімальною вартістю, тобто мати по можливості найменше число логічних входів всіх елементів;
- з метою підвищення швидкодії багаторозрядного суматора потрібен мінімальний час одержання функції перенесення $t_{\Pi} = k t_p$, де k — число послідовно увімкнених елементів від входів до виходів P_i або \bar{P}_i ; t_p — середня затримка розповсюдження сигналу одним логічним елементом в обраній серії інтегральних мікросхем; параметр k часто називають каскадністю (поверховістю) схем. Таким чином, для мінімізації часу одержання перенесення необхідно зменшити каскадність схеми та використати інтегральні мікросхеми з малим часом затримки розповсюдження сигналу;
- для схем однорозрядних суматорів на основі рівнянь (4.27) і (4.28) необхідно виробляти як прямі P_i , так й інверсні \bar{P}_i значення функції перенесення. Така організація перенесень називається парафазною.

Для побудови схеми однорозрядного суматора на універсальних логічних елементах НЕ і рівняння (4.27) і (4.28) перетворюються на основі правил подвійної інверсії та де Моргана до такого вигляду:

$$S_i = \overline{\bar{X}_i \bar{Y}_i Z_i \cdot \bar{X}_i Y_i \bar{Z}_i \cdot X_i \bar{Y}_i \bar{Z}_i \cdot X_i Y_i Z_i}; \quad P_i = \overline{\bar{X}_i Y_i \cdot X_i Z_i \cdot Y_i Z_i}. \quad (4.30)$$

Схема однорозрядного суматора, побудована на елементах НЕ І відповідно до рівнянь (4.30), показана на рис. 4.43, а; її вартість, яка вимірюється числом логічних входів всіх елементів, становить 27, каскадність $k = 3$.

Рівняння (4.27) та (4.28) можуть бути виражені через функцію "Виключальне ЧИ":

$$S_i = (X_i \oplus Y_i) \bar{Z}_i \vee (\overline{X_i \oplus Y_i}) Z_i = X_i \oplus Y_i \oplus Z_i; \quad (4.31)$$

$$P_i = X_i Y_i \vee (\bar{X}_i Y_i \vee X_i \bar{Y}_i) Z_i = X_i Y_i \vee (X_i \oplus Y_i) Z_i. \quad (4.32)$$

Схема однорозрядного суматора на елементах "Виключальне ЧИ" згідно з рівняннями (4.31) і (4.32) показана на рис. 4.43, б; її вартість становить вісім входів і каскадність $k = 2$.

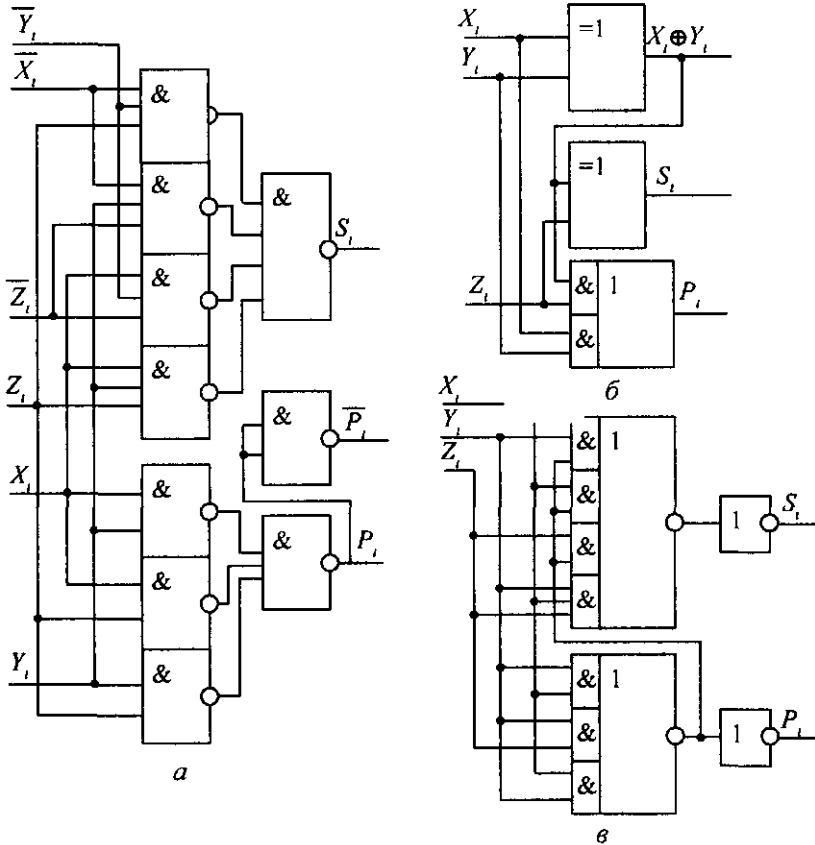


Рис. 4.43. Схеми однорозрядних суматорів: а — на елементах НЕ І;

б — на елементах "Виключальне ЧИ"; в — з використанням власного перенесення

Функції однорозрядного суматора — самоподвійні, тобто їхні інверсії утворюються інвертуванням значень аргументів без зміни місцезнаходження знаків диз'юнкції та кон'юнкції, наприклад, для перенесення з рівняння (4.28):

$$\bar{P}_i = X_i \bar{Y}_i \bar{Z}_i \vee \bar{X}_i Y_i \bar{Z}_i \vee \bar{X}_i \bar{Y}_i Z_i \vee X_i Y_i Z_i. \quad (4.33)$$

Помножуючи ліві та праві частини співвідношення (4.33) на макстерм $(X_i \vee Y_i \vee Z_i)$, одержують:

$$\bar{P}_i X_i \vee \bar{P}_i Y_i \vee \bar{P}_i Z_i = X_i \bar{Y}_i \bar{Z}_i \vee \bar{X}_i Y_i \bar{Z}_i \vee \bar{X}_i \bar{Y}_i Z_i. \quad (4.34)$$

Після підстановки лівої частини співвідношення (4.34) в праву частину виразу (4.27) одержують рівняння для функції S_i , з використанням власного перенесення:

$$S_i = \overline{P}_i X_i \vee \overline{P}_i Y_i \vee \overline{P}_i Z_i \vee X_i Y_i Z_i. \quad (4.35)$$

Схема однорозрядного суматора відповідно до рівнянь (4.35) і (4.29) показана на рис. 4.43, в; її вартість дорівнює 17 входів, каскадність $k = 2$. Важливою властивістю цієї схеми є використання тільки прямих значень вхідних змінних і однофазного ланцюга формування перенесення P_i в старший розряд.

Напівсуматором називається логічна схема, яка виконує додавання значень i -х розрядів X_i і Y_i двійкових чисел X і Y та реалізує на виході значення результату M_i і перенесення в старший сусідній розряд R_i :

$$M_i = \overline{X}_i Y_i \vee X_i \overline{Y}_i = X_i \oplus Y_i; \quad R_i = X_i Y_i. \quad (4.36)$$

Таким чином, напівсуматор виконує лише частину завдання підсумовування в i -му розряді, оскільки не враховує перенесення з сусіднього молодшого розряду. Схема напівсуматора, побудована на основі рівнянь (4.36), показана на рис. 4.44. З рівнянь (4.31) і (4.32) виходить, що схема однорозрядного суматора може бути побудована на основі двох напівсуматорів і додаткового логічного елемента ЧИ, як показано на рис. 4.44, в.

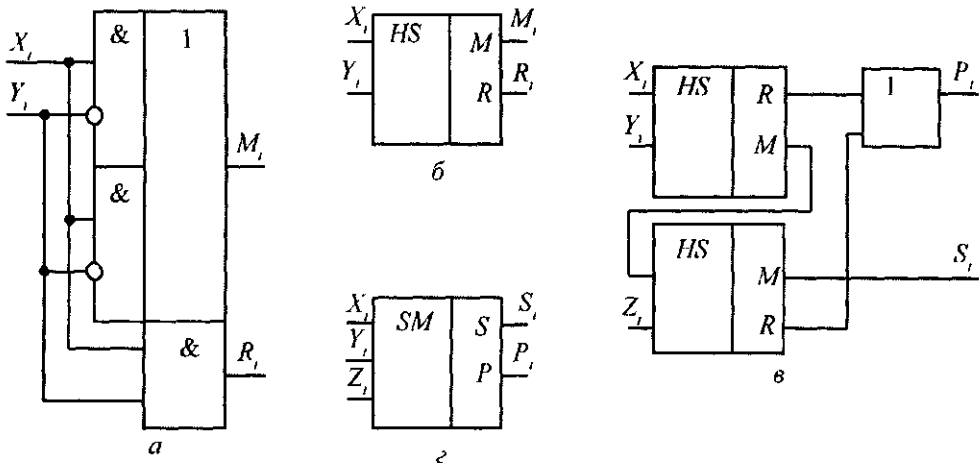


Рис. 4.44. Схеми підсумовування: а, б — напівсуматор і його умовне позначення; в, г — однорозрядний суматор і його умовне позначення

4.6.3. Послідовний багаторозрядний суматор

Послідовний двійковий багаторозрядний суматор містить: n -розрядні зсуваючі регістри операндів X і Y , регістр результату S , однорозрядний суматор SM і двоступеневий D -тригер для запам'ятовування перенесення. Усі регістри забезпечують одночасне зсування праворуч, у бік молодших розрядів (рис. 4.45).

У послідовному суматорі попарна подача значень розрядів X_i і Y_i починається з молодших розрядів. Утворюються значення суми S_i і перенесення P_i , які записуються відповідно в регістр результату та в тригер запам'ятовування перенесення на один такт T_c .

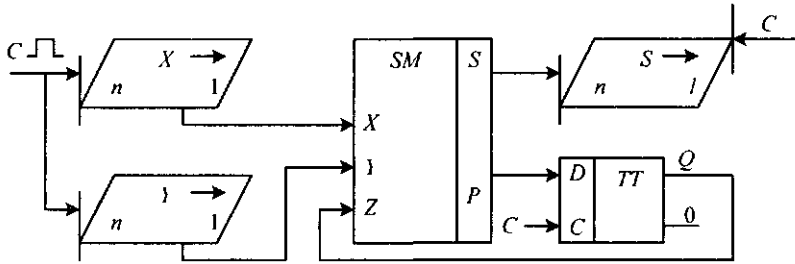


Рис. 4.45 Схема послідовного багаторозрядного суматора

Послідовне додавання виконується за стільки тактів, скільки розрядів у числі. Тому час додавання t_{Σ} визначається співвідношенням $t_{\Sigma} = nT_c$, де T_c — тривалість машинного такту.

Від'ємні числа рекомендується представляти в доповняльному коді.

Послідовний суматор потребує мінімальних апаратних витрат, однак тривалість операції додавання пропорційна розрядності операндів. Тому послідовний суматор можна використовувати у відносно повільнодіючих цифрових пристроях.

4.6.4. Паралельні багаторозрядні суматори

Паралельний багаторозрядний суматор містить n однорозрядних схем додавання, наприклад, чотири, як показано на рис. 4.46.

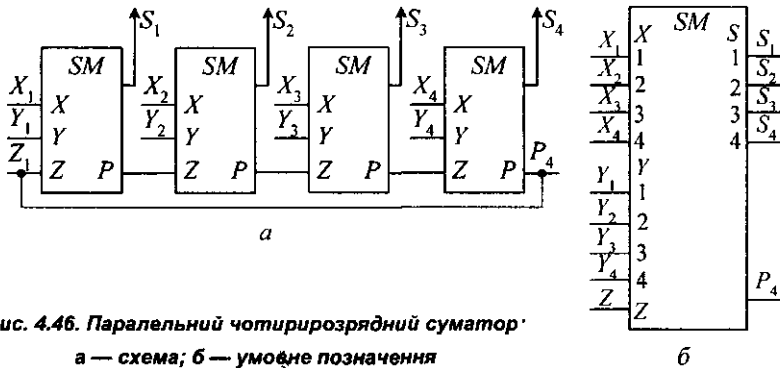


Рис. 4.46. Паралельний чотирирозрядний суматор:
а — схема; б — умовне позначення

Значення всіх розрядів двох чисел X та Y поступають на входи відповідних однорозрядних суматорів паралельно (одночасно). В паралельних суматорах з послідовним перенесенням значення сигналу перенесення P передається від розряду до розряду послідовно в часі (асинхронно). При застосуванні оберненого коду перенесення з найстаршого розряду подається на вхід перенесення молодшого розряду по ланцюгу циклічного перенесення (рис. 4.46, а). При застосуванні доповняльного коду ланцюг циклічного перенесення розривається, а на вхід перенесення молодшого розряду подається логічний нуль.

У паралельних суматорах з послідовним перенесенням час додавання визначається співвідношенням

$$t_{\Sigma} = (n-1) t_{\Pi} + t_{\Sigma},$$

де t_{Π} — час формування перенесення в кожному розряді, t_{Σ} — час додавання в най-

старшому розряді. У гіршому випадку можливий варіант, коли сигнал перенесення послідовно розповсюджується від першого до n -го розряду.

4.6.5. Паралельні суматори з паралельними перенесеннями

Для підвищення швидкодії суматорів використовують n -розрядні суматори з паралельними перенесеннями. При цьому суматор розбивається на дві частини: власне схему додавання і схему вироблення прискорених перенесень (СПП). Схема додавання містить n спрощених однорозрядних суматорів SM на три входи і тільки на один вихід суми S_i ; вони можуть будуватися за виразами типу (4.27), (4.30) та ін.

Схема прискорених переносів виробляє сигнали перенесення P_i одночасно у всіх розрядах на основі рівнянь, які враховують значення змінних X_i та Y_i , як у даному розряді, так і в попередніх молодших. Наприклад, перенесення P_i в першому (молодшому) розряді є функцією змінних X_1 і Y_1 та зовнішнього перенесення Z_1 :

$$P_1 = f_1(X_1, Y_1, Z_1) = Z_2.$$

Для другого розряду функція перенесення P_2 записується у вигляді:

$$P_2 = f_2(X_2, Y_2, Z_2) = f_2(X_2, Y_2, X_1, Y_1, Z_1) = Z_3.$$

Аналогічно для i -го розряду

$$P_i = f_i(X_i, Y_i, X_{i-1}, Y_{i-1}, \dots, X_1, Y_1, Z_1) = Z_{i+1},$$

а в останньому (старшому) розряді маємо:

$$P_n = f_n(X_n, Y_n, \dots, X_1, Y_1, Z_1).$$

Для зменшення апаратних витрат при побудові схеми та скорочення запису функцій перенесення P_i , вводять дві допоміжні функції: генерації (породження) перенесення в i -му розряді $R_i = X_i Y_i$ та розповсюдження (транзиту або передачі) перенесення через i -й розряд. Функція транзиту має дві еквівалентні за результатами застосування формули: $M_i = X_i \oplus Y_i$ — з немінімізованого рівняння перенесення (4.30) та $F_i = X_i \vee Y_i$ — з мінімізованого рівняння перенесення (4.29). Перенесення з i -го розряду в старший генерується за умови, що $R_i = X_i Y_i = 1$, тобто $P_i = 1$ при $X_i = Y_i = 1$. Сигнал перенесення з молодшого розряду передається транзитом в старший за умови, що $Z_i = 1$ та $M_i = X_i \oplus Y_i = 1$ або $F_i = X_i \vee Y_i = 1$. З урахуванням функцій перенесення і транзиту рівняння (4.29) і (4.30) запишуться відповідно у вигляді:

$$\begin{aligned} P_i &= X_i Y_i \vee (X_i \vee Y_i) Z_i = R_i \vee F_i Z_i; \\ P_i &= X_i Y_i \vee (X_i \oplus Y_i) Z_i = R_i \vee M_i Z_i. \end{aligned} \quad (4.37)$$

Для утворення в суматорі паралельних (одночасних) перенесень необхідно, щоб рівняння (4.37) будувалися з урахуванням значень змінних всіх молодших розрядів. Це досягається послідовною підстановкою в кожну функцію перенесення P_i замість змінної Z_i її значення P_{i-1} в напрямку від молодших до старших розрядів:

$$\begin{aligned} P_1 &= R_1 \vee F_1 Z_1 = Z_2; \\ P_2 &= R_2 \vee F_2 Z_2 = R_2 \vee F_2 P_1 = R_2 \vee F_2 R_1 \vee F_2 F_1 Z_1 = Z_3; \\ P_3 &= R_3 \vee F_3 Z_3 = R_3 \vee F_3 P_2 = R_3 \vee F_3 R_2 \vee F_3 F_2 R_1 \vee F_3 F_2 F_1 Z_1 = Z_4; \\ P_4 &= R_4 \vee F_4 Z_4 = R_4 \vee F_4 P_3 = \\ &= R_4 \vee F_4 P_3 \vee F_4 F_3 R_2 \vee F_4 F_3 F_2 R_1 \vee F_4 F_3 F_2 F_1 Z_1 = Z_5. \end{aligned} \quad (4.38)$$

Для i -го розряду:

$$P_i = R_i \vee F_i R_{i-1} \vee F_i F_{i-1} R_{i-2} \vee \dots \vee F_i F_{i-1} \dots F_1 Z_1.$$

Схема чотирирозрядного суматора з паралельними перенесеннями між розрядами на основі системи рівнянь (4.38) показана на рис. 4.47.

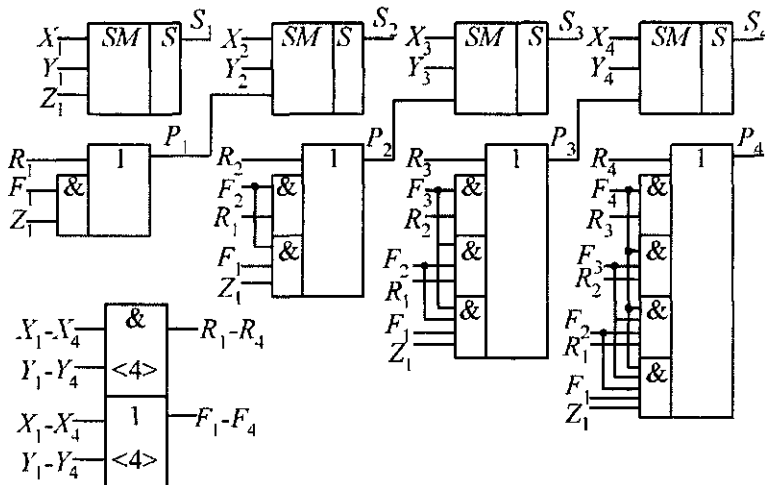


Рис. 4.47. Схема чотирирозрядного суматора з паралельними перенесеннями

Для вироблення функцій генерації R_1-R_4 і транзиту F_1-F_4 використано по чотири логічних елементи відповідно І та ЧИ, які показані на схемі пакетним зображенням.

Час додавання в схемі (рис.4.47) обчислюється за формулою $t_{\Sigma} = t_r + t_n$, де враховані затримки при виробленні сигналів генерації $t_r = t_p$, перенесення $t_n = t_p$ і додавання $t_s = 2t_p$. З урахуванням цього $t_{\Sigma} = 4t_p$. Наприклад, для $t_p = 5$ нс маємо: $t_{\Sigma} = 20$ нс. Таким чином, тривалість операції додавання в даному суматорі не залежить від числа розрядів n , якщо не враховувати погіршення часових характеристик логічних елементів при збільшенні числа входів і навантаження на виході.

В чотирирозрядному суматорі елементи І ЧИ в схемі перенесень разом мають 30 входів збігу. В 32-розрядному суматорі загальне число входів збігу в елементі І ЧИ буде дорівнювати 6550. Тому побудова багаторозрядних суматорів з паралельними перенесеннями утруднена з конструктивно-технологічних причин. В схемотехніці часто використовують частково паралельні перенесення: n -розрядний суматор розбивається на k груп по m розрядів у групі (звичайно $m = 4$) Перенесення між групами може бути послідовним, паралельним або наскрізним.

Для побудови n -розрядних суматорів з паралельними перенесеннями в групі для $m = 4$ (перший рівень) та універсальними способами перенесень між групами в рівнянні (4.38) виділяють допоміжні функції генерації перенесення з групи

$$G_1 = R_4 \vee F_4 R_3 \vee F_4 F_3 R_2 \vee F_4 F_3 F_2 R_1 \quad (4.39)$$

і транзиту через групу

$$H_1 = F_4 F_3 F_2 F_1. \quad (4.40)$$

Із врахуванням формул (4.39) і (4.40) рівняння для паралельного перенесення з першої групи має вигляд:

$$L_1 = P_4 = G_1 \vee H_1 Z_1.$$

За аналогією записують рівняння для G_i , H_i та L_i , де $i = 1, 2, \dots, k$.

Для організації універсальних способів перенесень у схему (див. рис. 4.47) включають додаткові схеми вироблення функцій G_i , H_i та L_i (рис. 4.48).

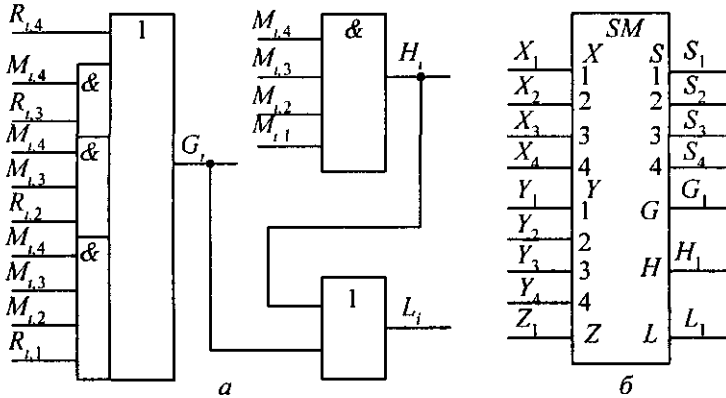


Рис. 4.48. Суматор з універсальними виходами перенесень:

а — схема для G_i ; б — умовне позначення першої групи суматора

За допомогою функцій G_i , H_i , L_i реалізують такі способи перенесень між групами:

- послідовний, якщо використовується вихід L_i (рис.4.49);
- паралельний, якщо використовуються виходи G_i та H_i ;
- наскрізний з використанням виходу H_i .

У всіх цих випадках в групах використовують паралельне перенесення.

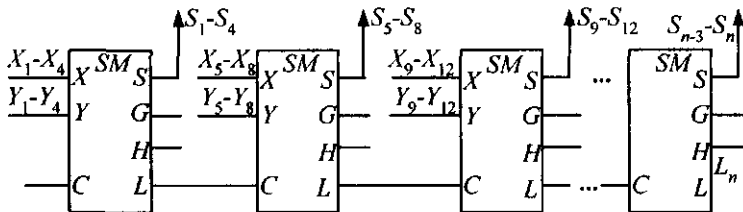


Рис. 4.49. Суматор з паралельними перенесеннями в групах і послідовними — між групами

Час додавання в суматорі (рис. 4.49) визначається затримками сигналів послідовного перенесення $2(k-1)t_p$ та виконанням дії в останньому розряді групи $t, = 2t_p$, тобто $t_{\Sigma} = 2(k-1)t_p + 2t_p$. Наприклад, для $k = 8$, $t_p = 5$ нс маємо: $t_{\Sigma} = 14t_p + 2t_p = 16t_p = 80$ нс.

Паралельні перенесення між групами суматора (другий рівень паралелізму) реалізують на основі системи рівнянь, які описують функцію перенесення C_i в кожну старшу групу з урахуванням рівнянь (4.39) і (4.40) (за аналогією з паралельними перенесеннями першого рівня — між розрядами в групі):

$$\begin{aligned}
 C_2 &= G_1 \vee H_1 C_1; \quad C_1 = Z_1; \\
 C_3 &= G_2 \vee H_2 C_2 = G_2 \vee H_2 G_1 \vee H_2 H_1 C_1; \\
 C_4 &= G_3 \vee H_3 C_3 = G_3 \vee H_3 G_2 \vee H_3 H_2 G_1 \vee H_3 H_2 H_1 C_1; \\
 C_4 &= G_4 \vee H_4 C_4 = G_4 \vee H_4 G_3 \vee H_4 H_3 G_2 \vee H_4 H_3 H_2 C_1 \vee H_4 H_3 H_2 H_1 C_1.
 \end{aligned} \tag{4.41}$$

У рівняннях (4.41) виділяють функції генерації перенесення D та транзиту V для забезпечення третього рівняння паралелізму:

$$D = G_4 \vee H_4 G_3 \vee H_4 H_3 G_2 \vee H_4 H_3 H_2 G_1; V = H_4 H_3 H_2 H_1. \quad (4.42)$$

Реалізацію функцій C_2 – C_4 та D і H на логічних елементах І ЧИ, І показано на рис. 4.50, а На їхній основі проектується СПП (рис. 4.50, б)

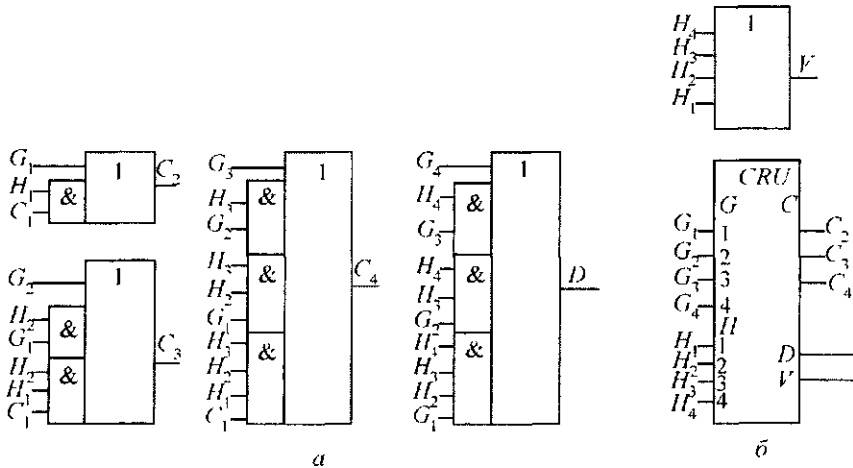


Рис. 4.50. Реалізація функцій C_2, C_3, C_4, D і V :
а — схеми; б — умовне позначення мікросхеми СПП

Мікросхема CRU (символ функції СПП) у серіях ТТЛШ позначається як ИПЧ, наприклад, КР153ЗИПЧ. Вона має входи G_i та $H_i, i = 1, 2, 3, 4$, на які поступають сигнали генерації та транзиту від чотирирозрядних суматорів з паралельними перенесеннями між розрядами в групах (див. рис. 4.49). З виходів мікросхеми CRU сигнали прискореного перенесення C_2, C_3 і C_4 подаються відповідно на входи перенесення C другої, третьої та четвертої (старшої) груп суматорів.

Таким чином, за допомогою мікросхеми CRU будується 16-розрядний суматор з паралельними перенесеннями в чотирирозрядних групах і між групами (рис. 4.51).

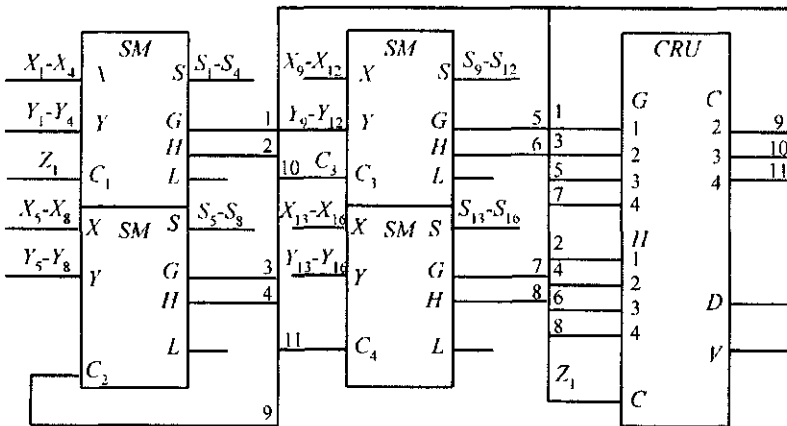


Рис. 4.51. Схема блока 16-розрядного суматора SM16 з паралельними перенесеннями в чотирьох групах та між групами

Час додавання в цьому суматорі визначається затримками сигналів: генерації (чи транзиту) $t_G = 2t_p$ в групі; перенесення із CRU $t_c = 2t_p$ та часу виконання дії в останньому розряді четвертої групи $t_s = 2t_p$. Таким чином, $t_\Sigma = t_G + t_c + t_s = 6t_p$. Для $t_p = 5$ нс маємо час додавання $t = 30$ нс.

Схема 64-розрядного суматора з третім рівнем паралелізму будується на основі системи рівнянь, одержаних за аналогією зі співвідношеннями (4.41) і (4.42):

$$\begin{aligned} K_2 &= D_1 \vee V_1 C_1; \quad C_1 = Z_1; \\ K_3 &= D_2 \vee V_2 C_2 = D_2 \vee V_2 D_1 \vee V_2 V_1 C_1; \\ K_4 &= D_3 \vee V_3 C_3 = D_3 \vee V_3 D_2 \vee V_3 V_2 D_1 \vee V_3 V_2 V_1 C_1 \end{aligned} \tag{4.43}$$

Схема 64-розрядного суматора з третім рівнем паралелізму показана на рис 4.52. Вона містить чотири блоки суматорів $SM16$ і додатково одну мікросхему CRU .

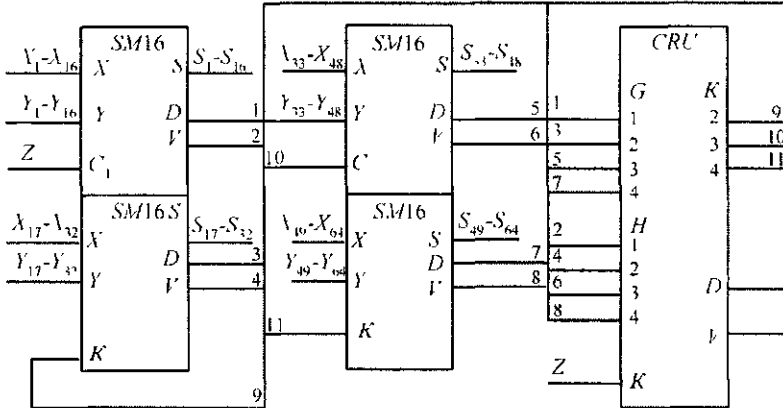


Рис. 4.52. Схема 64-розрядного суматора з третім рівнем паралелізму

Час додавання в цьому суматорі визначається затримками формування сигналів генерації (чи транзиту) в групах $t_G = 2t_p$, затримкою в двох схемах CRU $t_c = 4t_p$ та затримкою часу виконання дії в останньому розряді старшої групи. $t_\Sigma = t_G + t_c + t_s = 7t_p$. Для $t_p = 5$ нс маємо $t_\Sigma = 35$ нс.

4.6.6. Схеми для виконання логічних мікрооперацій

Усі комп'ютери можуть виконувати певний список порозрядних логічних операцій. Для їхньої реалізації використовують окремий блок логічних операцій (рис. 4.53)

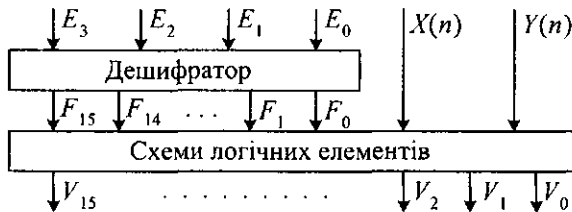


Рис. 4.53. Структура блока логічних операцій

У простих комп'ютерах цей блок суміщується із схемою паралельного суматора. Кожну логічну операцію в блоці задають кодом E_3-E_0 , який подається на входи де-

шифратора. Сигнали $F_{15}-F_0$ з виходів дешифратора безпосередньо керують вибором визначеної логічної операції. Складемо список булевих виразів для шести порозрядних логічних мікрооперацій:

$$\begin{aligned} L_{0i} &:= F_0(X_i \vee Y_i) < \text{логічне ЧИ} >; \\ L_{1i} &:= F_1(\overline{X_i} \vee Y_i) < \text{логічне НЕ ЧИ} >; \\ L_{2i} &:= F_2(X_i Y_i) < \text{логічне I} >; \\ L_{3i} &:= F_3(\overline{X_i} \overline{Y_i}) < \text{логічне НЕ I} >; \\ L_{4i} &:= F_4(X_i \oplus Y_i) < \text{виключальне ЧИ} >; \\ L_{5i} &:= F_5(\overline{X_i} \oplus Y_i) < \text{рівнозначність} >. \end{aligned}$$

Кожна порозрядна логічна операція виконується одним із визначених логічних елементів ЧИ, НЕ ЧИ, I, НЕ I та іншими. Функції $L_{5i}-L_{0i}$ в кожному розряді об'єднуються елементом ЧИ за формулою:

$$\begin{aligned} V_i = F_0(X_i \vee Y_i) \vee F_1(\overline{X_i} \vee Y_i) \vee F_2(X_i Y_i) \vee F_3(\overline{X_i} \overline{Y_i}) \vee \\ \vee F_4(X_i \oplus Y_i) \vee F_5(\overline{X_i} \oplus Y_i) \end{aligned} \quad (4.44)$$

Схема для реалізації порозрядних логічних мікрооперацій над i -ми розрядами операцій $X(n)$ і $Y(n)$ згідно з виразом (4.44) показана на рис. 4.54.

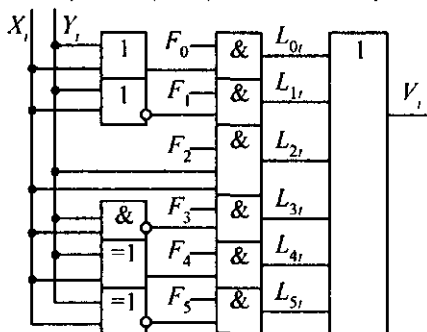


Рис. 4.54. Схема реалізації порозрядних логічних мікрооперацій над i -ми розрядами операндів

Блок логічних порозрядних операцій над n -розрядними двома операндами повинен мати n схем, аналогічних зображеним на рис. 4.54.

4.6.7. Мікросхеми ALU

Промисловість випускає мікросхеми із символом функції ALU для виконання 16 арифметичних та 16 порозрядних логічних мікрооперацій залежно від вхідних сигналів настройки. У серіях ТТЛШ 530, 531, 533, 555 та 1533 вони мають позначення ИПЗ; в серіях ЕЗЛ 100, 500 і 700 використовують позначення ИП179.

Мікросхема ALU в серіях ТТЛШ має (рис. 4.55):

- інформаційні входи для подання двох чотирирозрядних операндів X і Y ;
- входи настроювання E_3-E_0 для задання номера однієї з мікрооперацій;
- вхід M для задання типу мікрооперації: $M=0$ — арифметичні, $M=1$ — логічні;
- вхід перенесення $\overline{C1}$, необхідний тільки при виконанні арифметичних мікрооперацій;

- виходи: результату мікрооперації F_4-F_1 , послідовного перенесення \bar{L} , генерації G , транзитиву H , а також вихід з відкритим колектором від внутрішнього компаратора для вироблення ознаки рівності операндів $F_A \text{ } B$.

Перелік арифметичних і логічних операцій, які виконують ALU , наведений у табл. 4.16. При виконанні логічних операцій перенесення між розрядами не використовується. Арифметичні операції реалізуються з урахуванням перенесень і позик. В арифметичні операції включені фрагменти логічних дій. Наприклад, запис $(X \vee Y) + X\bar{Y}$ означає, що спочатку виконується операція інверсії (\bar{Y}), потім — логічного додавання $(X \vee Y)$ та логічного множення $(X \cdot \bar{Y})$, а потім одержані таким чином два числа додаються арифметично з урахуванням перенесень.

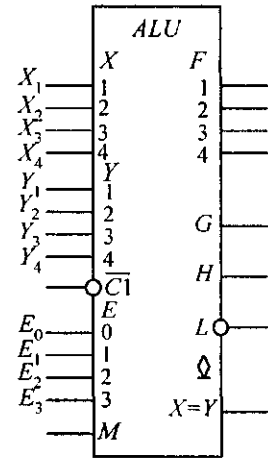


Рис. 4.55. Умовні позначення мікросхеми ALU 1533ИПЗ

Таблиця 4.16

E_3	E_2	E_1	E_0	Логіка $M=1$	Арифметика $M=0$
0	0	0	0	\bar{X}	X
0	0	0	1	$\overline{X \vee Y}$	$X \vee Y$
0	0	1	0	$\bar{X}Y$	$X \vee \bar{Y}$
0	0	1	1	0	-1
0	1	0	0	$\bar{X}Y$	$X + X\bar{Y}$
0	1	0	1	\bar{Y}	$(X \vee Y) + X\bar{Y}$
0	1	1	0	$X \oplus Y$	$X - \bar{Y} - 1$
0	1	1	1	$X\bar{Y}$	$X\bar{Y} - 1$
1	0	0	0	$\bar{X} \vee Y$	$X + X\bar{Y}$
1	0	0	1	$X \oplus \bar{Y}$	$X + Y$
1	0	1	0	Y	$(X + \bar{Y}) + X\bar{Y}$
1	0	1	1	$X \cdot Y$	$X\bar{Y} - 1$
1	1	0	0	1	$X + Y$
1	1	0	1	$X \vee \bar{Y}$	$(X \vee Y) + X$
1	1	1	0	$X \vee Y$	$(X \vee \bar{Y}) + X$
1	1	1	1	X	$X - 1$

Мікросхема ALU виконує операцію арифметичного додавання двох чотирирозрядних операндів X і Y , якщо на входи налаштування подані сигнали $E_3E_2E_1E_0 = 1001$ та $M = 0$. В цьому випадку мікросхема ALU виконує функцію суматора. Нарощування розрядності ALU здійснюється за допомогою мікросхеми CRU згідно рис. 4.51 та рис. 4.52.

4.7. Двійково-десяткові суматори

Двійково-десяткові суматори використовуються для обробки масивів десятикової інформації за порівняно простими алгоритмами, оскільки при цьому вилучаються витрати часу на переведення чисел з десятикової системи числення в двійкову і навпаки.

Кожна десятикова цифра X_i кодується двійковим кодом прямого заміщення "8421" (двійковою тетрадою), тобто $X_i = X_{i4}X_{i3}X_{i2}X_{i1}$ і $Y_i = Y_{i4}Y_{i3}Y_{i2}Y_{i1}$. Наприклад $X_i = 7_{10} = 0111_{2-10}$, $Y_i = 9_{10} = 1001_{2-10}$; для дворозрядних десятикових чисел:

$$X_i X_{i-1} = 16_{10} = 00010110_{2-10}; Y_i Y_{i-1} = 28_{10} = 00101000_{2-10}.$$

Один розряд двійково-десятьового суматора (декада) містить чотирирозрядний суматор $SM1$ для одержання попередньої суми в тетраді, чотирирозрядний суматор $SM2$ для корекції результату та логічний елемент І ЧИ для вироблення ознак корекції, як показано на рис. 4.56.

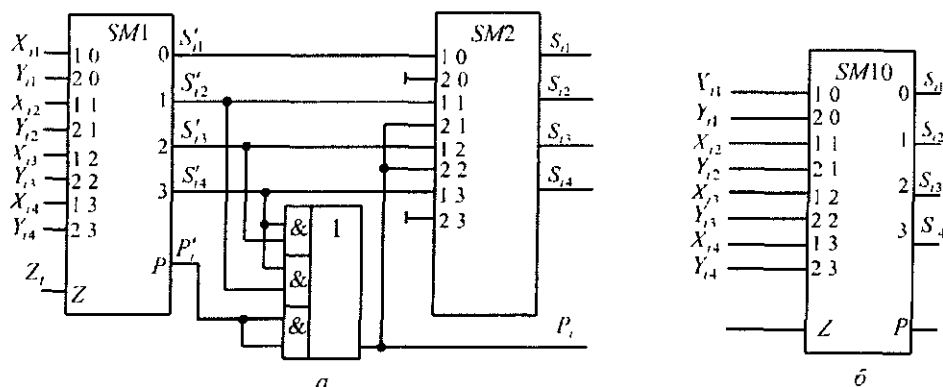


Рис. 4.56. Однорозрядний двійково-десятьовий суматор:

а — схема; б — умовне позначення

Декада працює таким чином. Двійкові тетради десятикових цифр $X_i = X_{i4}X_{i3}X_{i2}X_{i1}$ і $Y_i = Y_{i4}Y_{i3}Y_{i2}Y_{i1}$ разом із перенесенням поступають на входи суматора $SM1$ і на його виходах утворюється попередня сума $S'_{i7} = S'_{i4}S'_{i3}S'_{i2}S'_{i1}$, де S'_{i7} — десятиковий еквівалент тетради (табл. 4.17).

Таблиця 4.17

До корекції					Після корекції					Примітка
P'	S'_4	S'_3	S'_2	S'_1	P	S_4	S_3	S_2	S_1	
0	0	0	0	0	0	0	0	0	0	$0 \leq S_{i7} \leq 9$ Корекція не потрібна
0	0	0	0	1	0	0	0	0	1	
0	0	0	1	0	0	0	0	1	0	
0	1	0	0	1	0	1	0	0	1	$10 \leq S_{i7} \leq 15$ Корекція потрібна: мінус 10 і перенесення в старшу декаду
1	0	0	0	0	0	1	0	1	0	
1	0	0	0	1	0	1	0	1	1	
1	0	0	0	0	0	1	1	1	0	
1	0	1	0	1	0	1	1	1	1	

Таблиця 4.17. Продовження

До корекції					Після корекції					Примітка
1	0	1	1	0	1	0	0	0	0	
1	0	1	1	1	1	0	0	0	1	
1	1	0	0	0	1	0	0	1	0	
1	1	0	0	1	1	0	0	1	1	

При цьому можливі три випадки:

1. для значення $0 \leq S'_{i1} \leq 9$ корекція не потрібна;
2. для значень $10 \leq S'_{i1} \leq 15$ потрібно відняти з попередньої суми число 10 і здійснити перенесення в старшу сусідню декаду; віднімання числа 10 в доповняльному коді відповідає додаванню за допомогою суматора $SM2$ до попереднього результату числа шість, тобто плюс 0110_2 , ознакою такої корекції є одиничне значення функції корекції суми та перенесення

$$F'_{i1} = S'_{i4} S'_{i3} \vee S'_{i4} S'_{i2}, \quad (4.45)$$

яке реалізується елементом І ЧИ;

3. для значень $16 \leq S'_{i1} \leq 19$ на виході суматора $SM1$ виникає перенесення P'_{i1} з вагою 16_{10} . Однак у старшій декаді його значення сприймається як 10, тому потрібно додати до попереднього результату за допомогою суматора число шість, тобто 0110_2 .

З урахуванням рівняння (4.45) функцію корекції результату та перенесення можна записати у вигляді:

$$P_{i1} = P'_{i1} \vee F'_{i1} = P'_{i1} \vee S'_{i4} S'_{i3} \vee S'_{i4} S'_{i2}.$$

Таким чином, в усіх випадках, коли $P'_{i1} = 1$, до попередньої суми додається число плюс 0110_2 і формується перенесення у старший розряд.

Схема чотирирозрядного двійково-десятькового суматора з послідовним перенесенням в тетрадах і між декадами показана на рис. 4.57. Швидкодію таких суматорів розраховують за аналогією з двійковими послідовними перенесеннями. Для двійково-десятькових суматорів можна використовувати групві структури прискорених перенесень.

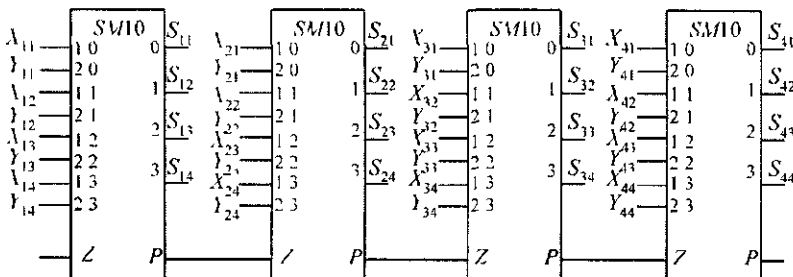


Рис. 4.57. Схема чотирирозрядного двійково-десятькового суматора.

Операція віднімання в двійково-десятьковому суматорі замінюється додаванням операндів у оберненому або доповняльному кодах. Обернений код від'ємних десяткових чисел одержують заміною кожної цифри її доповненням до дев'яти. Схема одного десяткового суматора з перетворювачами прямого коду операндів і результату в обернений код показана на рис. 4.58.

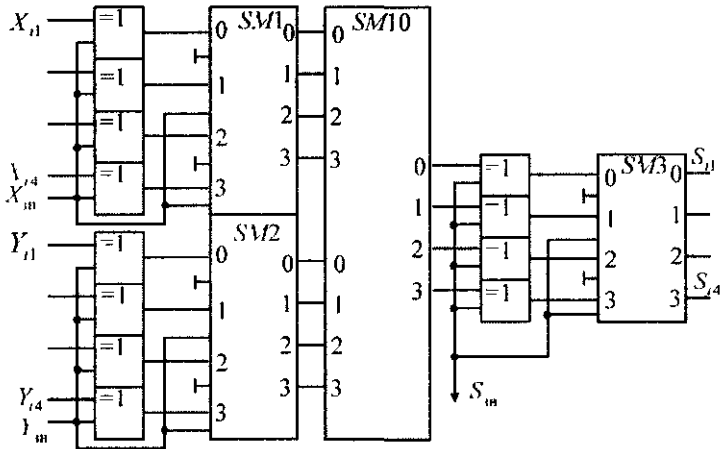


Рис. 4.58. Схема одного розряду десятичного суматора з перетворювачами прямого коду в обернений

Значення від'ємних чисел при $X_{zn} = 1$, $Y_{zn} = 1$, $S_{zn} = 1$ інвертується схемою "виключальне ЧИ"; при цьому утворюється двійковий код тетрадь з надлишком шість. Корекцію результату виконують суматорами $SM1$, $SM2$ і $SM3$, в яких віднімання замінюється додаванням двійкової тетради з оберненим кодом числа шість, тобто плюс 1010_2 .

Контрольні запитання

1. Назвіть області застосування дешифраторів і шифраторів.
2. Як будуються мультиплексори та демультимплексори шин?
3. Поясніть призначення компараторів.
4. Що таке контроль за модулем два?
5. Назвіть основні типи кодоперетворювачів.
6. За якими ознаками класифікують суматори?
7. Поясніть відмінність у побудові двійкового і двійково-десятичного суматорів
8. Назвіть способи збільшення швидкодії суматорів.
9. Поясніть функції мікросхем ALU .
10. Поясніть призначення мікросхеми CRU .
11. Поясніть спосіб перетворення прямого двійково-десятичного коду в обернений.
12. Як будується послідовний багаторозрядний суматор?
13. Що таке функція генерації перенесення?
14. Що таке функція транзиту перенесення?
15. В чому відміна синхронного суматора від асинхронного?
16. Поясніть побудову матричного дешифратора.
17. Поясніть принцип роботи пріоритетного шифратора.
18. Дайте визначення типового функціонального вузла комп'ютера.

Розділ 5

Цифро-аналогові і аналого-цифрові перетворювачі

5.1. Цифро-аналогові перетворювачі

5.1.1. Загальна характеристика цифро-аналогових перетворювачів

Цифро-аналогові перетворювачі (ЦАП) призначені для перетворення цифрової інформації в аналогову форму у вигляді напруги (іноді струму). Їх використовують у системах керування технологічними процесами, в аналогових мікропроцесорах, в дисплеях, графопобудовниках, робототехніці.

Цифро-аналогове перетворення полягає в тому, що для вхідного паралельного n -розрядного коду

$$X = X_1 2^{-1} + X_2 2^{-2} + \dots + X_i 2^{-i} + \dots + X_n 2^{-n},$$

де X_i — цифри 0 або 1, а 2^{-i} — вага i -го розряду, спочатку отримують струм I_X , пропорційний значенню числа X , а потім перетворюють його у вихідну напругу. Значення струму I_X визначається сумою еталонних струмів I_i , які створюються для кожного розряду числа

$$I_X = X_1 I_1 + X_2 I_2 + \dots + X_i I_i + \dots + X_n I_n,$$

причому підсумовуються струми тільки тих розрядів, для яких $X_i = 1$. Значення еталонів струму I_i пропорційні вазі позиції двійкового числа і зменшуються у два рази при переході від старшого i -го розряду до сусіднього молодшого з номером $i+1$.

5.1.2. Схеми цифро-аналогових перетворювачів

Структура ЦАП вміщує: резистивну або транзисторну матрицю для формування еталонних струмів; ключі для комутації еталонних струмів згідно з вхідним кодом до спільної точки підсумовування; операційний підсилювач (ОП) для перетворення струму I_X у вихідну напругу; допоміжні схеми для узгодження з вхідними рівнями сигналів; стабілізоване джерело опорної напруги $U_{оп}$.

Резистивні матриці будують або з набору двійково-зважених за номіналами резисторів, або у вигляді сходового (багатоланкового) ланцюжка резисторів лише двох номіналів $R-2R$. Схема ЦАП з резистивною матрицею на основі двійково-зважених опорів виду $R-2R-\dots-2^{n-1}R$ показана на рис. 5.1. У цій схемі опір резисторів матриці подвоюється при переході від старшого розряду до молодшого, а еталонні струми зменшуються у два рази. Наприклад, якщо для першого, найстаршого розряду взяти значення струму $I_1 = 1$ мА, то для другого розряду $I_2 = 0,5$ мА, для третього $I_3 = 0,25$ мА і т. д.

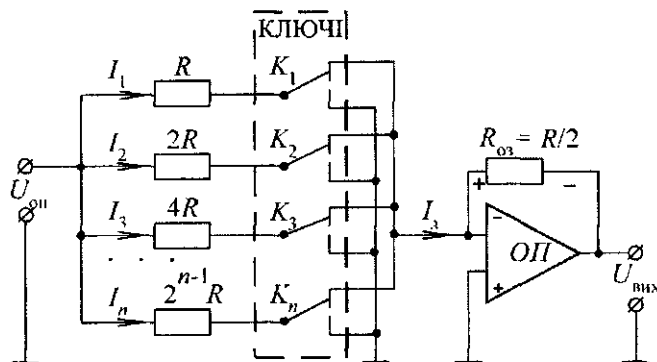


Рис. 5.1. Схема ЦАП із зваженими резисторами

Ключі $K_1 - K_n$ керуються рівнями напруги, які відображають цифри "нуль" і "один" відповідних розрядів вхідного коду. Джерело опорної напруги $U_{оп}$ найчастіше буває зовнішнім, але у деяких випадках його вбудовують у мікросхему ЦАП. На вході ОП завжди є практично нульовий потенціал, тому додавання розрядних струмів визначається співвідношенням

$$I_x = \frac{U_{оп}}{R} X_1 + \frac{U_{оп}}{2R} X_2 + \frac{U_{оп}}{4R} X_3 + \dots + \frac{U_{оп}}{2^{n-1}R} X_n = \\ = \frac{U_{оп}}{R} (X_1 + 2^{-1} X_2 + 2^{-2} X_3 + \dots + 2^{-(n-1)} X_n) = \frac{U_{оп}}{R} \sum_{i=1}^n X_i \cdot 2^{-(i-1)}. \quad (5.1)$$

Напругу на виході ЦАП розраховують за формулою

$$U_{вих} = -I_x R_{о3} = -U_{оп} \sum_{i=1}^n X_i \cdot 2^{-i}, \quad (5.2)$$

де $R_{о3} = R/2$ — опір у ланцюгу оберненого зв'язку підсилювача.

Приклад 5.1

Розрахувати напругу на виході ЦАП при перетворенні вхідного шестирозрядного цифрового коду $X = 101011$. Опорна напруга $U_{оп} = 10$ В.

На основі виразу (5.2) маємо:

$$U_{вих} = -10(1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} + 0 \cdot 2^{-4} + 1 \cdot 2^{-5} + 1 \cdot 2^{-6}) = \\ = -10(2^{-1} + 2^{-3} + 2^{-5} + 2^{-6}) = -6,72 \text{ В.}$$

Основний недолік розглянутої структури ЦАП — це величезний діапазон опорів матриці, особливо при великій розрядності вхідного коду. Схема ЦАП на основі сходової матриці $R-2R$ показана на рис. 5.2. Тут використані резистори тільки двох номіналів, за допомогою яких реалізуються співвідношення (5.1) і (5.2), тобто при переході від старшого розряду до молодшого еталонний струм зменшується у два рази.

Для швидкодіючого ЦАП струмові ключі будують на діодах і біполярних транзисторах; для перетворювачів середньої і низької швидкодії використовують ключі на КМОН-структурах. Схема діодного ключа, який можна використовувати в розглянутих схемах ЦАП, показана на рис. 5.3, а.

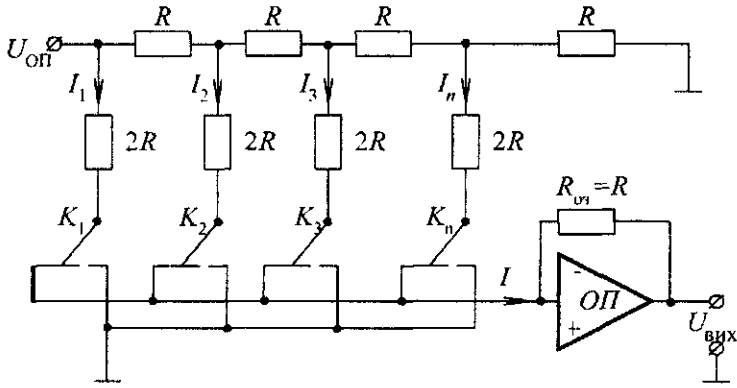


Рис. 5.2. Схема ЦАП з матрицею R-2R

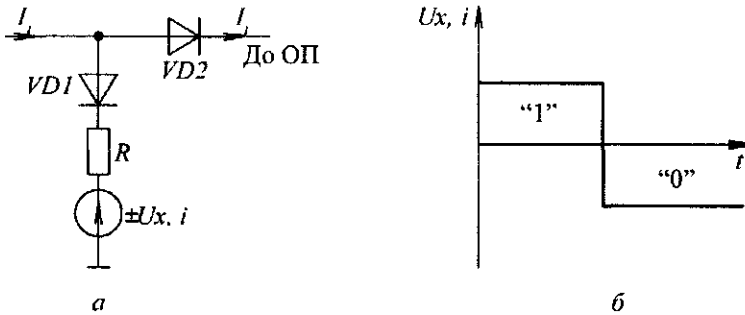


Рис. 5.3. Діодний ключ: а — схема; б — вхідний сигнал i-го розряду

Вхідний сигнал $U_{x,i}$, який відображає цифру в розряді вхідного коду, є двополярним: позитивний рівень (лог. 1) закриває діод $VD1$ і еталонний струм I , через діод $VD2$ поступає на вхід ОП; негативний рівень (лог. 0) закриває діод $VD2$ і струм I , замикається на джерело вхідного сигналу $U_{x,i}$.

У схемі ЦАП часто вбудовують генератори струмів (ГС), призначені для термо-стабілізації струмів. Один з варіантів ГС з оберненим зв'язком показаний на рис. 5.4.

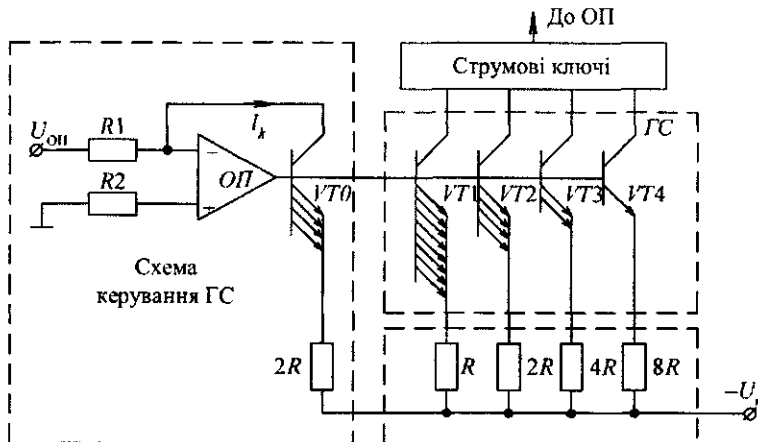


Рис. 5.4. Схема ГС з вузлом керування

У цій схемі транзистори $VT1-VT4$ стабілізують струми резистивної матриці $R-2R-4R-8R$. Допоміжний транзистор $VT0$ разом з підсилювачем утворюють схему керування, яка стабілізує струми транзисторів $VT1-VT4$. Зміна еталонних струмів від розрахункових значень контролюється за струмом колектора транзистора $VT0$, який ідентичний транзисторам $VT1-VT4$ і знаходиться з ними в однакових температурних умовах. Якщо через колектор транзистора $VT0$ протікає заданий струм I_K , то сигнал корекції з виходу ОП не буде поступати на бази транзисторів $VT0-VT4$ (баланс ОП) і режим роботи транзисторів не зміниться. При відхиленні I_K від заданого значення на бази транзисторів $VT1-VT4$ буде подаватися сигнал корекції.

5.1.3. Двійкові коди, використовувані у цифро-аналогових перетворювачах

Введення інформації в ЦАП здійснюється, в основному, в паралельному коді. У ЦАП використовують три основних двійкових коди: прямий, зміщений і доповняльний (рис. 5.5). Прямий код зручний при перетворенні сигналів систем стеження тому, що при переході через нуль не міняються старші розряди коду, а це дозволяє реалізувати лінійний перехід від малих позитивних до малих негативних вихідних напруг. Для перетворення позитивних і негативних кодів використовують знаковий розряд, який керує перемиканням вихідної напруги ЦАП (рис. 5.5, а).

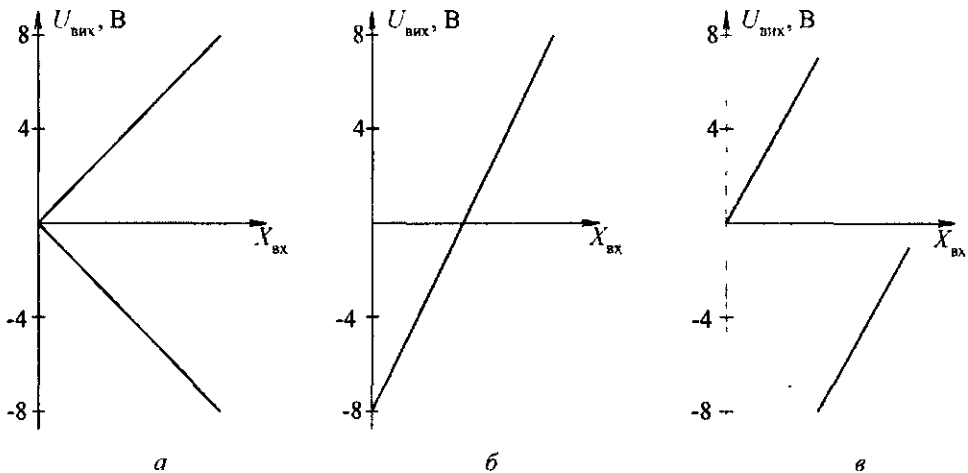


Рис. 5.5. Вихідні напруги ЦАП для кодів: а — прямого; б — зміщеного; в — доповняльного

Для виключення комутуючих елементів із схеми ЦАП використовують зміщений код, що є найпростішим (рис. 5.5, б). У доповняльному коді (рис. 5.5, в) позитивні числа перетворюються так, як і в прямому коді, а негативні — двійковим доповненням відповідного позитивного числа (інверсія всіх розрядів з подальшим додаванням одиниці в молодший розряд).

5.1.4. Перемножуючі цифро-аналогові перетворювачі

Працювати ЦАП може з постійною або змінною опорною напругою. Перетворювачі, які працюють зі змінною напругою $U_{оп}$, називаються перемножуючими. Операція множення виконується схемою ЦАП безпосередньо у вигляді:

$$U_{вих} = X_{вх1} X_{вх2}, \quad (5.3)$$

де $X_{вх1} = U_{оп}$; $X_{вх2} = X(X_1, X_2, \dots, X_n)$. На основі виразу (5.3) відтворюється операція множення аналогового сигналу $X_{вх1}$ на цифровий код $X_{вх2}$.

Продуктивність обчислювальних систем підвищується, якщо на входах використовуються ЦАП, які реалізують множення аналогових і цифрових сигналів. Цифро-аналогові перетворювачі можуть бути одно- і багатоканальними, що досягається об'єднанням в одній ВІС декількох ідентичних перетворювачів, які працюють незалежно один від одного, або використанням на входах і виходах ЦАП комутаторів аналогових і цифрових сигналів.

Структура багатоканального перемножуючого ЦАП показана на рис. 5.6.

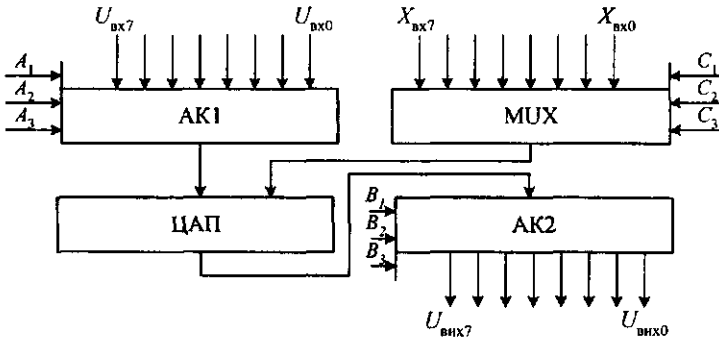


Рис. 5.6. Структура багатоканального перемножуючого ЦАП

Комутатори аналогових каналів $AK1$ і $AK2$ і мультиплексор цифрових каналів MUX керуються трирозрядними адресними кодами: $A_3A_2A_1$, $B_3B_2B_1$ і $C_3C_2C_1$ відповідно. Якщо $A_3A_2A_1 = 011$ і $C_3C_2C_1 = 110$, то отримуємо для виходу з адресою $B_3B_2B_1 = 100$ значення операції множення $U_{вих4} = U_{вх3} X_{вх6}$.

Змінюючи кодові комбінації на адресних входах, можна отримати будь-який добуток виду:

$$U_{вихi} = U_{вхj} U_{вхk}, \quad i, j, k = 1, 2, \dots, 8.$$

У перемножуючого ЦАП в основному застосовуються струмові ключі на МОП-транзисторах в режимах малих напруг і струмів, що дає їм можливість працювати з опорною напругою довільного знаку і форми. Вихідний сигнал такого ЦАП може розташовуватися у будь-якому із чотирьох квадрантів залежно від знаків, які приймаються співмножниками.

5.1.5. Основні параметри і характеристики цифро-аналогових перетворювачів

Основними параметрами ЦАП є число розрядів вхідного цифрового коду, роздільна здатність, похибки перетворення, діапазон вихідних сигналів, динамічні параметри (табл. 5.1).

Таблиця 5.1

Тип мікросхеми	Розрядність, n	Максимальне відхилення $\delta_{\text{лд}}$, %	Час установлення $t_{\text{уст}}$, мкс	Технологія
K427ПА4	16	0,0015	20	КМОН
K572ПА2	12	0,025	15	Біполярна
KM1118ПА1	8	0,75	0,02	Біполярна
K1108ПА3	8	0,5	0,05	Біполярна
KM1148ПА1	10	0,75	1	Біполярна

Число розрядів n вхідного коду для різних типів ЦАП дорівнює від восьми до вісімнадцяти. Число розрядів визначає максимальну кількість кодових комбінацій на вході ЦАП, що дорівнює 2^n .

Діапазон зміни вихідної напруги (без урахування знака) визначається із співвідношення (5.2) при $X_i = 1, i = 1, 2, \dots, n$:

$$U_{\text{вих max}} = U_{\text{оп}}(2^{-1} + 2^{-2} + \dots + 2^{-n}) = U_{\text{оп}}(1 - 2^{-n}),$$

якщо $n = 10, U_{\text{оп}} = 10 \text{ В}$, то $U_{\text{вих max}} \approx 10 \text{ В}$.

Роздільна здатність h характеризується мінімальним квантом вихідної напруги, який відповідає зміні вхідного коду на одиницю молодшого розряду: $h = U_{\text{оп}}/2^n \approx \approx 10 \text{ мВ}$ для попереднього прикладу.

Абсолютна похибка перетворення δ_A — відхилення вихідної напруги від розрахункової в кінцевій точці характеристики перетворення. Типова похибка ЦАП не перевищує $\pm 1/2$ молодшого розряду.

Нелінійність δ_n — максимальне відхилення реальної характеристики перетворення від теоретичної (прямої лінії, що з'єднує точку нуля і мінімального вихідного сигналу).

Диференціальна нелінійність δ_d — максимальне відхилення різниці двох аналогових сигналів сусідніх кодів від значення молодшого розряду.

Параметри δ_A, δ_n і δ_d виражаються в частках молодшого розряду або у відсотках від повної шкали вихідної напруги.

Час установлення $t_{\text{уст}}$ — інтервал часу від подачі вхідного коду до моменту досягнення вихідним сигналом сталого значення із заданою похибкою (зазвичай $\pm 1/2$ молодшого розряду). Цей час визначає загальну швидкодію ЦАП.

Залежно від значень параметрів виділяють прецизійні ($\delta_n < 0,1\%$) і швидкодіючі ($t_{\text{уст}} \leq 100 \text{ нс}$) ЦАП.

Розрізняються ЦАП структурою резистивної матриці (зважені резистори або $R - 2R$), елементною базою, функціонально повнотою, узгоджувальними пристроями, числом джерел живлення та ін.

Реалізуються ЦАП у вигляді гібридних і напівпровідникових мікросхем. Більшість мікросхем ЦАП — напівпровідникові, функціонально закінчені. Ряд ЦАП використовують зовнішні джерела опорної напруги і вихідні операційні підсилювачі.

Живлення ЦАП може здійснюватися від одного або від кількох джерел, що визначається елементною базою. Перспективами розвитку ЦАП є підвищення швидкодії та точності, зручність узгодження з мікропроцесорами, зниження споживаної потужності.

5.2. Аналого-цифрові перетворювачі

5.2.1. Загальна характеристика аналого-цифрових перетворювачів

Аналого-цифрові перетворювачі (АЦП) призначені для перетворення аналогової інформації (звичайно у вигляді напруги) у цифровий код. Застосовують АЦП у мікропроцесорних системах, у цифрових вимірювальних приладах. Області застосування їх багато в чому аналогічні ЦАП, оскільки вони часто використовуються спільно, наприклад, в автоматизованих системах керування (АСК) (рис. 5.7).



Рис. 5.7. Аналого-цифровий і цифро-аналоговий перетворювачі в контурі керування

Основними параметрами і характеристиками АЦП є:

- число розрядів n вихідного коду;
- роздільна здатність h — мінімальний квант вхідної напруги, за якої вихідний код змінюється на одиницю молодшого розряду;
- нелінійність δ_n — максимальне відхилення вихідного коду від розрахункового значення у всьому діапазоні шкали;
- абсолютна похибка δ_A — найбільше відхилення вихідного коду від розрахункового в кінцевій точці шкали;
- час перетворення $t_{пр}$ — інтервал від моменту початку перетворення до появи на виході сталого коду; часто замість $t_{пр}$ швидкодія АЦП характеризується частотою перетворення;
- діапазон і полярність вхідної напруги, число джерел живлення, струм споживання, можливість спільної роботи з мікропроцесорами.

У АЦП застосовуються такі методи перетворення:

- послідовної лічби (з використанням ЦАП або з двотактним інтегруванням);
- порозрядного кодування (послідовного двійкового наближення);
- паралельної дії (зчитування);
- паралельно-послідовні (комбіновані).

Аналого-цифрові перетворювачі послідовної лічби

Метод послідовної лічби із застосуванням АЦП заснований на урівноваженні вхідної напруги сумою еталонів, які підраховуються лічильником. Момент урівноваження визначається аналоговим компаратором.

Схема АЦП послідовної лічби показана на рис. 5.8, а. У ній за сигналом "Пуск" RS-тригер переключається в стан "1" і дозволяє проходження імпульсів від генератора G через елемент І на вхід підсумовування двійкового лічильника CT_2 .

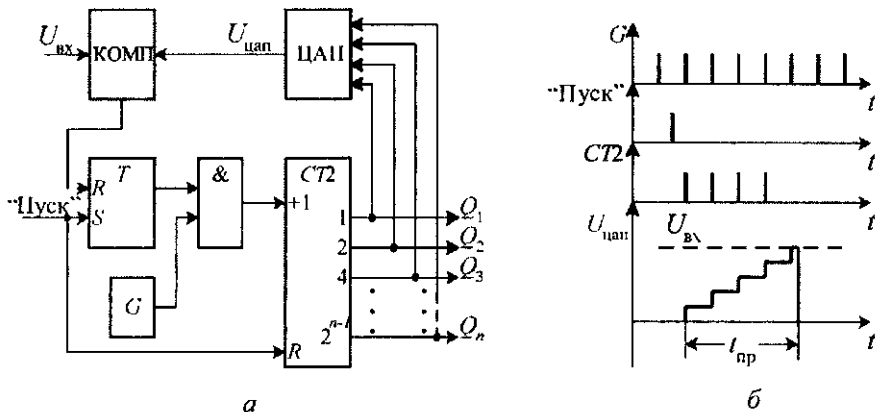


Рис. 5.8. Аналого-цифровий перетворювач послідовної лічби з ЦАП:
 а — схема; б — часові діаграми роботи

Наростаючий цифровий код з виходу лічильників $СТ2$ перетворюється за допомогою ЦАП в напругу, яка подається на вхід компаратора КОМП. На другий вхід КОМП поступає вимірювана напруга $U_{вх}$.

У момент рівності напруг $U_{вх} = U_{ЦАП}$ компаратор виробляє сигнал скидання тригера. Після цього рахунок імпульсів припиняється і на виході лічильника $СТ2$ фіксується цифровий еквівалент вхідної напруги. Час перетворення $t_{пр}$ залежить від значення напруги $U_{вх}$ (рис. 5.8, б).

У АЦП з двотактним інтегруванням (рис. 5.9, а) перетворення реалізується за два такти.

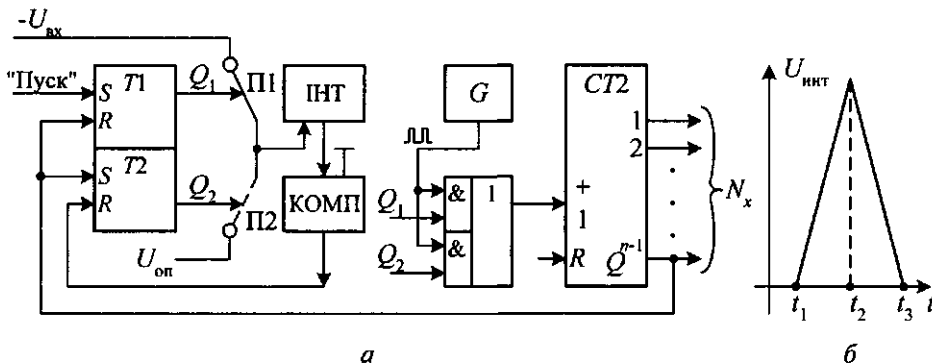


Рис. 5.9. Аналого-цифровий перетворювач з двотактним інтегруванням:
 а — схема; б — часова діаграма роботи

У першому такті сигнал "Пуск" переключає RS-тригер $T1$ у стан "1". Вхідна напруга мінус $U_{вх}$ через ключ $\Pi1$, керований виходом Q_1 , поступає на інтегратор ІНТ. Імпульси від генератора G через схему І ЧИ подаються на вхід підсумовування лічильника $СТ2$. Після підрахунку 2^n імпульсів (n — розрядність лічильника) сигнал з виходу Q_n старшого розряду лічильника $СТ2$ скидає тригер $T1$ і встановлює в стан "1" RS-тригер $T2$. На цьому перший етап інтегрування закінчується.

У другому такті на інтегратор поступає опорна напруга плюс $U_{оп}$ через ключ $\Pi2$,

керований виходом Q_2 . Одночасно високий рівень напруги з виходу Q_2 дозволяє проходження імпульсів від генератора G на підсумовуючий вхід лічильника CT_2 (починається новий рахунок).

У момент, коли напруга на виході ІНТ дорівнює нулю, компаратор обнуляє тригер T_1 , проходження імпульсів на вхід лічильника блокується, і на його виходах фіксується цифровий код N_X , пропорційний вхідній напрузі. Для інтегратора прирости напруг в першому і другому тактах рівні (рис. 5.9, б):

$$U_{\text{вх}}(t_2 - t_1) = U_{\text{он}}(t_3 - t_2).$$

Оскільки $t_2 - t_1 = T_0 2^n$, $t_3 - t_2 = N_X T_0$, де T_0 — період проходження імпульсів генератора G , то отримуємо:

$$U_{\text{вх}} = (U_{\text{он}} N_X) / 2^n. \quad (5.4)$$

Приклад 5.2

При вимірюванні десятирозрядний АЦП з опорною напругою $U_{\text{он}} = 10$ В, зафіксував у лічильнику число імпульсів $N_X = 455$. Знайти значення вхідної напруги $U_{\text{вх}}$.

На основі виразу (5.4) маємо:

$$U_{\text{вх}} = \frac{10 \cdot 455}{2^{10}} = \frac{4550}{1024} \approx 4,48 \text{ В.}$$

Аналого-цифрові перетворювачі з порозрядним кодуванням

У схемі АЦП з порозрядним кодуванням (урівноваженням) вхідна напруга $U_{\text{вх}}$ порівнюється стрибкоподібно з вихідною напругою $U_{\text{ЦАП}}$ за певною програмою (рис. 5.10, а).

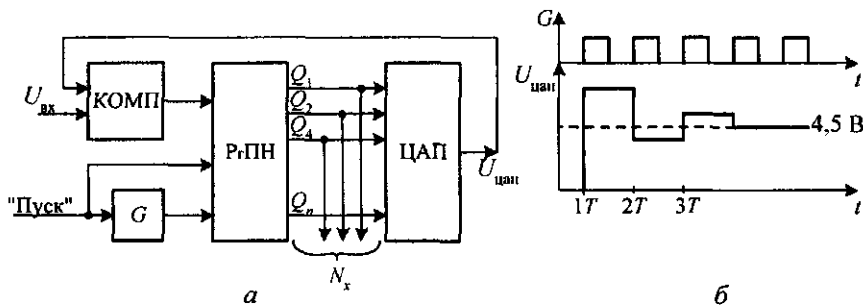


Рис. 5.10. Аналого-цифровий перетворювач порозрядного кодування:

а — схема; б — часові діаграми роботи

Сигнал "Пуск" включає генератор періодичних імпульсів G і одночасно записує одиницю в старший розряд зсувового регістра порозрядного наближення РгПН, а інші розряди обнуляє. У першому такті за допомогою компаратора КОМП порівнюються вхідна напруга $U_{\text{вх}}$ і напруга з виходу ЦАП, яка відповідає значенню старшого розряду РгПН. Алгоритм порівняння такий: якщо $U_{\text{ЦАП}} > U_{\text{вх}}$, то в наступному такті ця одиниця вилучається, а замість неї в РгПН додається наступна із сусіднього розряду. Якщо ж $U_{\text{ЦАП}} < U_{\text{вх}}$, то одиниця в старшому розряді зберігається і до неї додається одиниця із сусіднього розряду і т. д.

Перетворення закінчується за час $t_{пр} = n_0 T_0 + 1$, де n_0 — розрядність РГПН; T_0 — період проходження тактуючих імпульсів. Часові діаграми порозрядного урівноваження для $U_{вх} = 4,5 \text{ В}$ і $n = 4$ показані на рис. 5.10, б.

Аналого-цифровий перетворювач паралельної і паралельно-послідовної дії

Принцип роботи АЦП паралельної дії заснований на одночасному порівнянні вхідного сигналу з $2^n - 1$ еталонами напруг, де n — розрядність вихідного коду. Порівняння здійснюється за допомогою компараторів, результат порівняння запам'ятовується в регістрі й надалі шифрується в n -розрядний двійковий вихідний код (рис. 5.11).

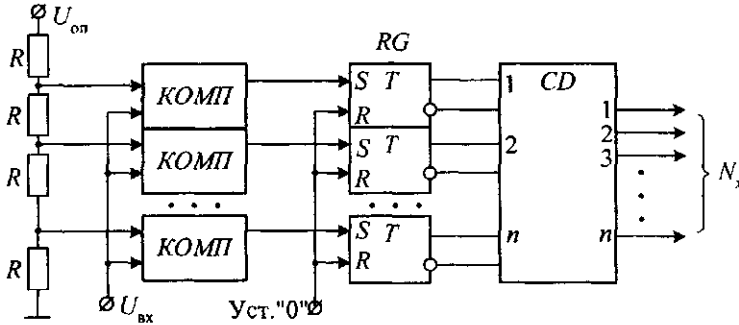


Рис. 5.11. Схема АЦП паралельної дії

У АЦП опорні напруги формуються за допомогою резистивного дільника. Кожна опорна напруга $U_{опі}$ подається разом з вхідною напругою $U_{вх}$ на входи відповідних компараторів КОМП. Спрацьовують лише ті компаратори, в яких $U_{вх} \geq U_{опі}$. Результат порівняння через фіксуючі тригери T подається на шифратор CD , який перетворює його у вихідний n -розрядний двійковий код. Аналого-цифрові перетворювачі паралельної дії є найбільш швидкодіючими, оскільки перетворення здійснюється за одну операцію порівняння. До недоліків цього АЦП відноситься велике число джерел опорної напруги і відповідних їм компараторів.

Паралельно-послідовні АЦП мають підвищену розрядність, високу швидкодію і прийнятну складність (рис. 5.12).

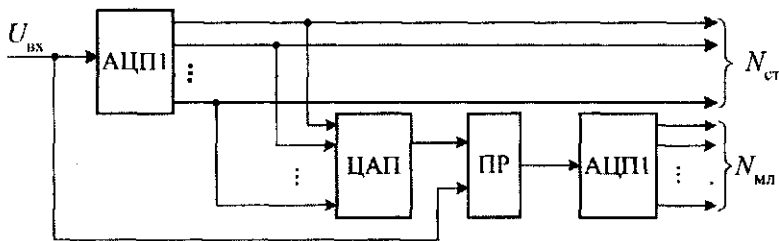


Рис. 5.12. Схема АЦП паралельно-послідовної дії

У таких комбінованих перетворювачах декілька малорозрядних АЦП паралельної дії з'єднуються послідовно між собою.

Вхідний аналоговий сигнал подається на перший АЦП (АЦП1), на виході якого формуються старші розряди $N_{ст}$ вихідного коду. Ці розряди підключаються також на вхід ЦАП. Вихідний сигнал з ЦАП порівнюється в підсилювачі різниці (ПР) з вхідним сигналом. Посилена різниця цих сигналів подається на вхід другого АЦП (АЦП2), який перетворює її у вихідний код $N_{мл}$ молодших розрядів.

5.2.2. Основні параметри і характеристики аналого-цифрових перетворювачів

У АЦП використовують методи порозрядного кодування послідовної лічби з двійковим інтегруванням і паралельного перетворення. Мікросхеми АЦП виконують за гібридною і напівпровідниковою технологією. У останні роки випускають, в основному, напівпровідникові АЦП. Основні параметри і характеристики деяких напівпровідникових АЦП наведені в табл. 5.2.

Таблиця 5.2

Тип мікросхеми	Розрядність, n	Максимальне відхилення, $\delta_{лд}$, %	Час перетворення $T_{п}$, мкс	Технологія	Примітка
KP572PB3	8	$\pm 0,75$	7,5	КМОН	МК, СМ
K1107PB1	6	$\pm 0,5$	0,1	Біполярна	—
M1107PB6	10	$\pm 1,5$	0,06	Біполярна	ФЗ, СМ
K1108PB2	12	± 1	0,9	Біполярна	ФЗ

Примітка. У табл. 5.2 СМ — сумісність з мікропроцесорами, ФЗ — функціональна закінченість, МК — багатоканальність.

Деякі АЦП є функціонально закінченими, але більшість вимагають додаткових зовнішніх елементів: операційних підсилювачів, джерел опорної напруги, генераторів тактових імпульсів, резисторів і конденсаторів.

Мікросхеми АЦП звичайно мають діапазон зміни вхідної напруги 0–10 В, а деякі АЦП допускають використання двополярного вхідного сигналу. Розрядність АЦП становить 6–12, причому ряд АЦП допускають нарощування розрядності.

Швидкодія АЦП визначається, в основному, методом перетворення і елементною базою (ТТЛШ, ЕЗЛ, КМОН). Найбільшу швидкодію мають АЦП паралельної дії на базі ЕЗЛ-елементів ($t_{пр} \leq 20$ нс). Перетворювачі за рівнями вихідних сигналів узгоджуються з ТТЛШ-, ЕЗЛ- і КМОН-мікросхемами. Більшість сучасних АЦП сумісні з мікропроцесорними пристроями. Вихідні ланцюги в таких АЦП мають три стійких стани (лог 0, лог 1 і Z).

Вихідним кодом АЦП найчастіше є двійковий. У АЦП з двійковим інтегруванням застосовується двійково-десятковий код для сполучення з індикаторами і вимрювальними приладами. Тут для представлення кожного десяткового знака використовуються чотири двійкових розряди. Використовуються також обернений і доповняльний коди.

Деякі АЦП — це ВІС аналого-цифрової системи збирання даних, що включає в свій склад, крім перетворювача, багатоканальний мультиплексор, оперативний запам'ятовувачий пристрій, схеми буферів і керування.

Основними напрямками вдосконалення АЦП є:

- підвищення швидкодії основних вузлів, особливо компараторів;
- використання АЦП комбінованої дії;
- підвищення точності перетворення, зокрема, збільшення розрядності до 16 і більше;
- зниження споживаної потужності;
- досягнення зручності та гнучкості застосування, особливо узгодження з мікропроцесорними пристроями.

Контрольні запитання

1. Назвіть області застосування ЦАП.
2. Розкажіть про принцип перетворення цифрового коду в напругу.
3. Назвіть відмінності у структурі резисторних матриць зі сходовими резисторами типу $R - 2R$.
4. Яке призначення струмових ключів?
5. Яку функцію в схемі ЦАП виконує ОП?
6. Охарактеризуйте основні параметри ЦАП.
7. Назвіть серії мікросхем вітчизняних ЦАП.
8. Які основні параметри і характеристики АЦП?
9. Поясніть структуру АЦП послідовної лічби.
10. Нарисуйте часові діаграми АЦП послідовної лічби.
11. Поясніть принцип побудови АЦП з двотактним інтегруванням.
12. Нарисуйте структурну схему АЦП порозрядного кодування.
13. Поясніть часові діаграми роботи АЦП порозрядного кодування.
14. Охарактеризуйте роботу АЦП паралельної дії.
15. Поясніть роботу АЦП паралельно-послідовної дії.
16. Як будується перемножуючий ЦАП?

Розділ 6

Основи комп'ютерної техніки**6.1. Загальна характеристика комп'ютерної техніки і коротка історія її розвитку**

Комп'ютерна техніка — це сукупність засобів для автоматизації процесів обробки інформації, а також галузь техніки, яка займається розробленням, виготовленням та експлуатацією комп'ютерів.

Комплекс електронного устаткування, яке призначене для автоматизації процесів обробки дискретної інформації і має спільне керування, називають **цифровою електронною обчислювальною машиною (ЕОМ)**. В наш час діапазон використання ЕОМ суттєво розширився: називають більше 20 тис. областей їх застосування — від науково-інженерних задач до штучного інтелекту, математичного моделювання, робототехніки. Тому замість терміну ЕОМ використовують його більш сучасний синонім — комп'ютер.

Перші відомості про обчислювальні пристрої відносяться до 1500 р., коли Леонардо да Вінчі розробив ескіз 13-розрядного підсумовуючого пристрою з десятизубцевими колесами. У 1642 р. Б. Паскаль у Франції створив першу діючу модель восьмирозрядного цифрового пристрою "Паскаліна". Кожна цифра в ньому визначалася положенням десятисекторного колеса. Додавання здійснювалося поворотом колеса на відповідне число секторів. Німецький вчений Г. Лейбніц у 1670 р. описав механічну лічильну машину, яка виконувала операції додавання, віднімання, множення і ділення.

У 1833 р. англійський математик Ч. Бебідж спроектував і майже 30 років вдосконалював "аналітичну машину", в яку були закладені фундаментальні для комп'ютерної техніки принципи:

- автоматичне (без зупинки) виконання операцій;
- необхідність пам'яті ("складу") для зберігання чисел, набраних зубчастими колесами;
- робота за програмою, що вводиться "на ходу" з перфокарти в виконавчий пристрій ("фабрику") зі швидкістю виконання операцій в машині.

Обмежені технічні можливості того часу не дозволили реалізувати ці прогресивні ідеї.

Автоматичні цифрові машини на механічних і релейних елементах створювали одночасно і незалежно в багатьох країнах. Так, у 1936 р. А. Тьюрінг в Англії і Е. Пост в США розробили концепцію абстрактної автоматичної цифрової машини, яка могла розв'язувати будь-яку алгоритмізовану задачу. У 1941 р. німецький інженер К. Зюс розробив дві електромеханічні машини з програмним керуванням "Зюс-1" і "Зюс-2".

У 1944 р. в США під керівництвом Г. Айкена була розроблена електромеханічна машина "Марк-1", де вперше були використані ідеї Ч. Бебіджа. У тому ж році в США

під керівництвом Д Еккерта, Д Моучлі та Дж фон Неймана був створений перший у світі комп'ютер ENIAC з використанням електронних ламп і електромеханічних реле. В машині було задіяно 18 тис ламп і 1,5 тис реле, вона виконувала за одну секунду 5 тис операцій додавання, 300 — множення, споживала потужність 150 кВт.

Ці перші машини мали такі недоліки

- програми не зберігалися в пам'яті, а набиралися за допомогою зовнішніх складних і довготривалих комутацій,
- дані подавалися і оброблювалися в десятковій системі числення послідовно (розряд за розрядом)

6.2. Архітектура і структура комп'ютерів

Архітектура — це найбільш загальні принципи побудови комп'ютерів. Вона охоплює коло питань, суттєвих в першу чергу для користувача: функціональний склад технічних та програмних засобів та їхню взаємодію в процесі обробки інформації, систему команд, їхні формати і способи кодування, методи адресації команд і даних.

Основні принципи побудови комп'ютерів виклали в 1946 р американські математики Дж фон Нейман, К Голдстайн і А Беркс. Сукупність цих принципів породила класичну нейманівську архітектуру, яка зберігає актуальність і сьогодні.

Загалом нейманівська архітектура має такі основні ознаки

- наявність одного обчислювача, що має процесор, пам'ять, засоби введення–виведення інформації та керування,
- використання двійкової системи числення як для представлення інформації так і для виконання арифметико-логічних операцій,
- розміщення в єдиній спільній пам'яті команд і чисел фіксованої довжини,
- лінійну структуру адресації комірок пам'яті, що вимагає наявності в процесорі лічильника команд,
- централізоване автоматичне послідовне зчитування команд із пам'яті та інтерпретацію їх процесором, дані обробляються паралельно — одночасно над усіма розрядами машинного слова,
- низький рівень машинної мови

Перший комп'ютер EDSAC із збереженою програмою в пам'яті на 512 ртутних лініях затримки був побудований М Уілксом (Англія) у 1949 р. Машина виконувала 15 тис додавань і 120 множень за одну секунду. У 1950 р під керівництвом Дж фон Неймана був збудований перший повністю електронний комп'ютер класичної архітектури EDVAC, який поклав початок машинам першого покоління.

У 1948–1950 рр в Інституті електротехніки в Києві була створена мала електронна лічильна машина "МЕЛМ". Її розробкою керував видатний вчений і конструктор багатьох комп'ютерів академик С О Лебедев. В машині "МЕЛМ" були використані (незалежно від Неймана) основні принципи класичної архітектури. Комп'ютер містив 3500 ламп, 2500 діодів, займав площу 60 м², споживав потужність 25 кВт і виконував 50 операцій за одну секунду.

Комп'ютер класичної архітектури вміщує (рис 6.1)

- арифметико-логічний пристрій (АЛП),
- оперативну пам'ять (ОП),

- засоби зберігання і введення–виведення інформації: зовнішні запам'ятовуючі пристрої (ЗЗП); пристрої введення інформації (ПВв); пристрої виведення інформації (ПВив); усі ці пристрої називають зовнішніми чи периферійними (ПП);
- пристрій керування (ПК). Разом з АЛП він утворює процесор. При наявності в машині декількох процесорів виділяють центральний (ЦП).

Арифметико-логічний пристрій призначений для виконання арифметичних і логічних операцій, передбачених системою команд даного комп'ютера. До складу АЛП входять реєстри і комбінаційні схеми. Дані для обробки в АЛП поступають з ОП і називаються операндами. Результати операцій пересилаються в ОП чи тимчасово зберігаються в реєстрах АЛП.

Пристрій керування зчитує і дешифрує у відповідній послідовності команди, формує і подає керуючі сигнали для інших пристроїв комп'ютера. До ПК (рис. 6.1) віднесено і пульт оператора.

Оперативна пам'ять призначена для тимчасового зберігання програм і даних, в ній виконуються операції записування і читання інформації. Крім ОП, використовують також постійну пам'ять, в якій здійснюються тільки операції читання. Оперативну (ОЗП) і постійну пам'ять (ПЗП) та реєстри АЛП називають **внутрішньою пам'яттю** (рис. 6.2). Процесор і ОП разом створюють **ядро комп'ютера**.

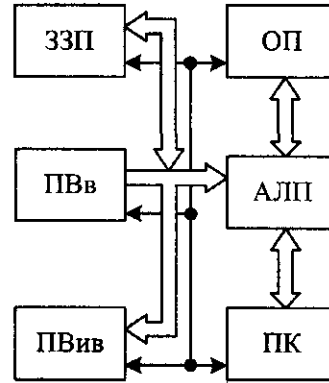


Рис. 6.1. Структура комп'ютера

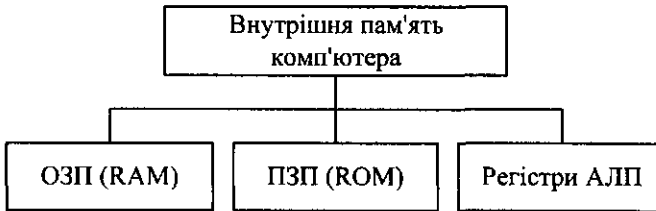


Рис. 6.2. Внутрішня пам'ять комп'ютера

Операції введення–виведення — це обмін інформацією між ядром машини і ПП. Операція введення передає інформацію з ПП в ядро комп'ютера, а операція виведення — навпаки.

Зовнішня пам'ять призначена для тривалого і енергонезалежного зберігання великих об'ємів інформації. Фізично її реалізують у вигляді накопичувачів (рис. 6.3):

- на гнучких магнітних дисках (НГМД);
- на жорстких магнітних дисках (НЖМД); вони називаються вінчестерами;
- на оптичних (лазерних) дисках (НОД);
- на магнітних стрічках (НМС).

Накопичувачі на дисках мають електромеханічний привід (дисковод), який забезпечує обертання диска, блок магнітних головок для читання чи записування, сис-

тему установлення (позиціювання) магнітної головки в потрібне положення і електронний блок керування.

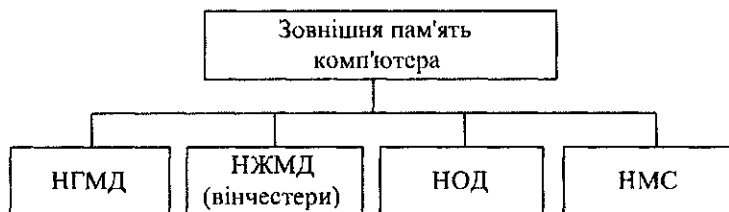


Рис. 6.3. Зовнішня пам'ять комп'ютера

Усі види зовнішньої пам'яті забезпечують обмін інформацією з ядром комп'ютера. Але ЗЗП виділяють в окремий вид ПП за такими ознаками:

- зовнішня пам'ять забезпечує зберігання великих масивів інформації та швидкодійний обмін з ядром комп'ютера (більше 30 тис. байт/с);
- інформація в ЗЗП зберігається у вигляді, недоступному для безпосереднього сприймання людиною.

Пристрої введення і виведення інформації (ПВВ) розглядають як єдину функціональну частину комп'ютера. Різні за своїми функціями, принципами побудови та характеристиками ПВВ і ЗЗП разом створюють групу дуже різноманітних зовнішніх або периферійних пристроїв.

До ПВВ інформації входять (рис. 6.4):

- клавіатура, яка призначена для введення програм і даних та керування роботою машини;
- сканери, які призначені для введення графічної інформації;
- ручні маніпулятори — миша, кульковий маніпулятор і джойстик — призначені для швидкого переміщення курсора в задану точку екрана дисплея та виконання інших дій;
- пристрої мовного введення, які призначені для керування машиною за допомогою мовних команд;
- пристрої введення з перфострічок, перфокарт та ін.

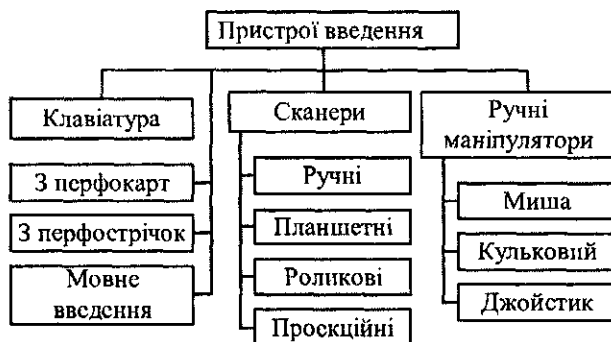


Рис. 6.4. Пристрої введення інформації

До ПВВВ інформації відносять (рис. 6.5):

- принтери (матричні, струминні, лазерні);

- дисплеї (на електронно-променевих трубках (ЕПТ), на рідких кристалах, плазмові, люмінесцентні), що призначені для відображення інформації, яка вводить з клавіатури (для контролю правильності набору даних) або з пам'яті машини;
- плотери (графопобудовники), які призначені для друкування, креслень високої якості;
- синтезатори звуку і мови, призначені для перетворення аналогових сигналів в цифровий код і навпаки;
- пристрої виведення інформації на перфострічки і перфокарти.

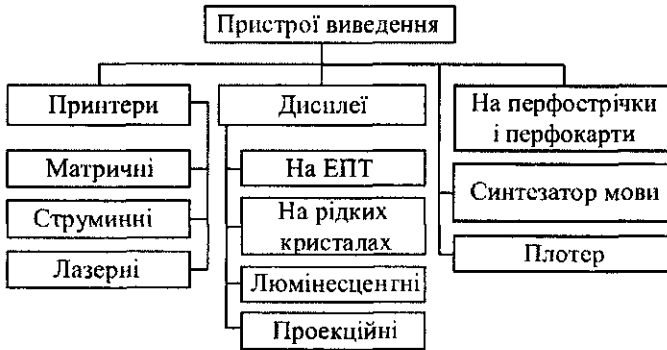


Рис. 6.5. Пристрої виведення інформації

Зв'язок між функціональними частинами машини здійснюють за допомогою **інтерфейсу** — сукупності шин, сигналів, допоміжних мікросхем та алгоритмів, призначених для обміну інформацією між пристроями комп'ютера.

Виділяють три шини (рис. 6.6):

- адреси (ША), призначена для передачі адреси комірок ОП і регістрів ПП;
- даних (ШД), призначена для передачі даних;
- керування (ШК), призначена для передачі керуючих сигналів від процесора до пристроїв і навпаки.

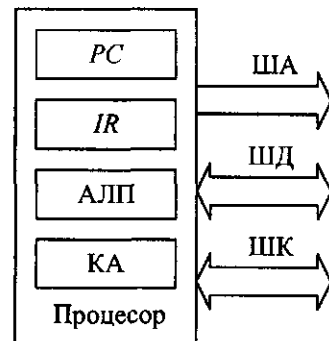


Рис. 6.6. Структура процесора

6.3. Принцип програмного керування

В комп'ютері реалізують принцип програмного керування, суть якого така. Для розв'язання кожної задачі розробляють алгоритм на основі числових методів обчислення. Алгоритм переводиться на мову, властиву даній машині, у вигляді **програми** — мовної конструкції, яка є впорядкованою послідовністю описів і команд, призначених для обробки інформації. Кожна команда визначає дії комп'ютера щодо ви-

конання будь-якої операції, які реалізують апаратні (технічні) і програмні засоби Програма записується в ОП у вигляді машинних слів, які кодуються цифрами 0 і 1 та розрізняються тільки способом використання Код операції поступає в реєстр команд *IR (instruction register)* і потім дешифрується, а дані — в реєстри АЛП (рис 6 6)

Команди програми розміщені в ОП лінійно (одна за одною) і виконуються послідовно Номер команди в ОП визначається програмним лічильником *PC (program counter)* Керуючий автомат (КА) виробляє множину керуючих сигналів, які подаються на всі пристрої машини Реєстр команд, програмний лічильник і керуючий автомат входять до складу ПК Послідовне керування зумовлене наявністю одного процесора Команди умовного і безумовного розгалуження змінюють лінійний порядок зчитування і виконання команд

Множина всіх операцій, що реалізуються в комп'ютері, складає його **операційні ресурси** Комп'ютери, операційні ресурси яких забезпечують виконання будь-якого алгоритму обробки інформації, називають **універсальними** Для цього теоретично достатньо мати в операційних ресурсах тільки чотири операції пересилку слова між будь-якими комірками ОП, додавання (віднімання) одиниці до слова, умовний перехід за збігом слів та безумовну зупинку комп'ютера Проте в комп'ютерах операційні ресурси складаються з десятків і сотень команд, що спрощує програмування

Загалом в комп'ютерах використовують список команд, що забезпечує виконання таких груп операцій

- пересилки даних між реєстрами АЛП, реєстрами і ОП,
- арифметичних операцій над двійковими числами із фіксованою та плаваючою комою додавання, віднімання, знакового і беззнакового множення і ділення,
- логічних операцій заперечення, диз'юнкції, кон'юнкції, додавання за модулем два,
- установлення відношень — більше, менше, нерівно, більше-рівно та ін ,
- зсуву вліво чи вправо — арифметичного, логічного, циклічного,
- керування програмою умовними та безумовними переходами та викликами процедур, безумовними та умовними поверненнями з процедур, перериванням програм, деякі комп'ютери мають спеціальні команди для організації циклів,
- введення–виведення даних між ядром машини і ПП,
- спеціальних операцій для машин із співпроцесорами (математичними розширювачами) обчислення квадратного кореня, синуса, косинуса, логарифмічні та ін ,
- перетворення з одного формату в інший (наприклад, з восьмибітного в 16-бітний),
- системних операцій — завантаження службових реєстрів, захист пам'яті,
- мультимедійних операцій для виконання дій зі звуком, графікою, відображенням

Із зростанням продуктивності процесора збільшується і число команд

Комп'ютери, залежно від їхньої складності та призначення, працюють в таких режимах (рис 6 7)

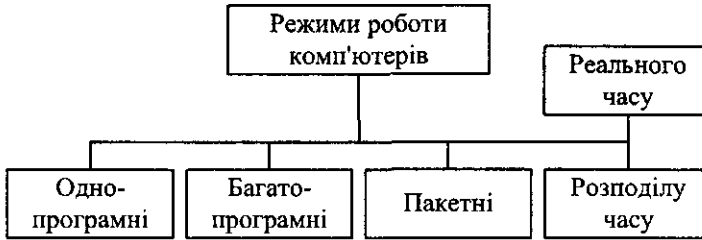


Рис. 6.7. Режими роботи комп'ютерів

- однопрограмному — кожна програма окремо завантажується в ОП і виконується до одержання результату;
- багатопрограмному — в пам'ять завантажується декілька програм; коли виконання однієї з програм зупиняється через потребу звертання до ПП, то машина переключається на виконання іншої програми;
- пакетному — в ЗЗП формується пакет задач, які потім зчитуються в ОП групами і виконуються в багатопрограмному режимі;
- розподілу часу (колективного користування) — доступ до комп'ютера користувачів за допомогою власного терміналу;
- реального часу — забезпечується взаємодія комп'ютера із зовнішніми об'єктами в темпі, який вимагає швидкодія об'єкта.

6.4. Основні характеристики комп'ютерів

Важливою характеристикою комп'ютера є продуктивність — об'єктивна кількісна міра роботи машини. Використовують такі міри продуктивності: пікову, номінальну, системну та експлуатаційну (рис. 6.8).

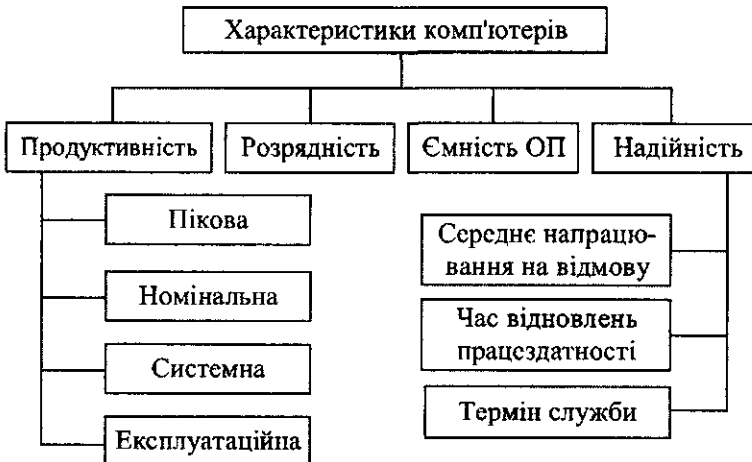


Рис. 6.8. Основні характеристики комп'ютерів

Пікова продуктивність — середнє число коротких операцій типу "регістр—регістр" за секунду (оп./с) без операцій обміну з ОП. За кордоном пікову продуктив-

ність оцінюють для команд типу "Немає операції" в мільйонах операцій за секунду (млн. оп./с) або в MIPS (Million Instruction per Second). Пікову потужність часто називають швидкодією комп'ютера.

Номинальна продуктивність — середнє число суміші команд V_c із врахуванням їхньої статистичної ваги (частоти повторення), які виконує ядро комп'ютера у вибраному класі задач (залежить від швидкості ОП):

$$V_c = 1/(P_1t_1 + P_2t_2 + \dots + P_it_i + \dots + P_nt_n),$$

де P_i та t_i — відповідно ваговий коефіцієнт кожної i -ї команди і час її виконання; n — число команд.

Номинальну продуктивність часто називають "швидкодією комп'ютера на суміші команд". Продуктивність потужних машин часто вимірюють у **мегафлопсах** (MFLOPS) — у мільйонах операцій за секунду над операндами з плаваючою комою.

Системну продуктивність вимірюють за допомогою типових оціночних програм (бенчмарків), реалізованих на мовах високого рівня. Результати оцінки системної продуктивності комп'ютера конкретної архітектури наведені в числових таблицях.

Експлуатаційну продуктивність оцінюють даними про реальне робоче навантаження в основних областях застосування; при цьому враховують необхідну площу розміщення машини, механічні та кліматичні умови експлуатації, споживану потужність та ін.

До характеристик комп'ютерів також відносять:

- розрядність машинного слова, яке зберігається, пересилається і обробляється як єдине ціле; вимірюється в бітах, байтах;
- ємність оперативної пам'яті в бітах, байтах, кілобайтах, мегабайтах, гігабайтах;
- надійність, яка характеризує середнє напрацювання на відмову — не менше 15 тис. год; час відновлення працездатності, термін служби (не менше 10 років).

6.5. Покоління комп'ютерів

Покоління — це категорія в історичній класифікації комп'ютерів, що базується переважно на технології виробництва їхньої елементної бази. Розрізняють п'ять поколінь комп'ютерів (рис. 6.9).

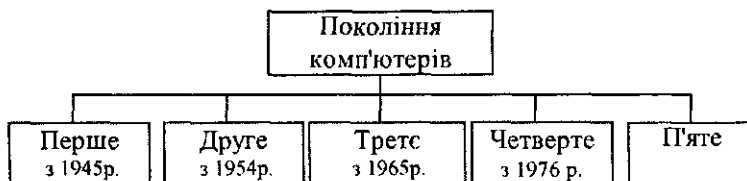


Рис. 6.9. Покоління комп'ютерів

З кожним новим поколінням комп'ютерів збільшуються їхні операційні ресурси, продуктивність і надійність; зменшується споживана потужність, маса і габарити; вдосконалюються мови програмування; спрощується взаємодія користувача з машиною. Наведемо приблизні дати утворення поколінь і основні їхні характеристики.

Перше покоління (з 1945 р.): елементна база — електронні лампи; ємність ОП — 100 байт; швидкодія — 10 тис. оп./с; мова програмування — машинний код. В нашій країні використовувались машини першого покоління "МЕЛМ", "Київ", "Урал-1", "Урал-4", "Мінськ-1, "Мінськ-14" та ін.

Друге покоління (з 1954 р.): елементна база — транзистори та напівпровідникові діоди; ємність ОП — 1000 байт; швидкодія — 1 млн оп./с; мова програмування — асемблер. Моделі комп'ютерів другого покоління — "Мир", "Дніпро" (керівники розробки академіки В. М. Глушков і Б. М. Малиновський), "Урал-11", "Урал-14", "Мінськ-22", "Мінськ-32" та ін.

Третє покоління (з 1965 р.): елементна база — малі і середні інтегральні схеми; ємність ОП — 10 тис. байт; швидкодія — 10 млн оп./с; мова програмування — процедурні мови високого рівня. В нашій країні були створені крупні уніфіковані та програмно сумісні великі й середні універсальні машини єдиної системи від ЄС-1010 до ЄС-1066 (прототип — машина IBM 360); міні- і мікрокомп'ютери — від СМ-1 до СМ-4, "Електроніка-60" та інші (прототип — машина PDP-11 фірми DEC); в комп'ютерах третього покоління порівняно з першим швидкодія і надійність збільшилися приблизно в 1000 разів і настільки ж зменшилась споживана потужність.

Четверте покоління (з 1976 р.): елементна база — великі й надвеликі інтегральні схеми; ємність ОП — 0,1–1 Мбайт; швидкодія — 100–1000 млн оп./с; мова програмування — непроцедурні мови високого рівня; до четвертого покоління відносять персональні комп'ютери на базі мікропроцесорів типу Pentium фірми Intel.

П'яте покоління — контури його архітектури тільки розробляються і можливі такі характеристики: елементна база — оптоелектроніка, криоелектроніка, НВІС; ємність ОП — 1–10 Гбайт; швидкодія — 1000 млрд оп./с; засоби обміну з користувачем — пристрої голосового зв'язку; мова програмування — нові непроцедурні мови високого рівня.

Залежно від продуктивності, розмірів, функціонального призначення і вартості виділяють суперкомп'ютери, великі комп'ютери, міні- та мікрокомп'ютери (рис. 6.10).

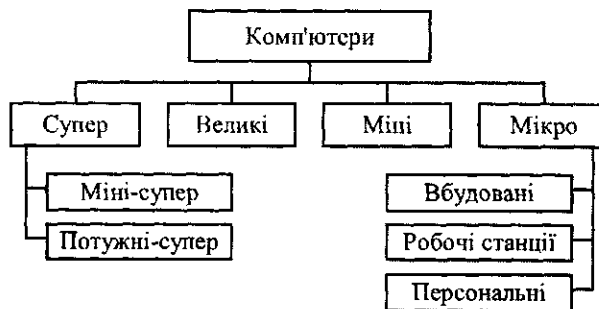


Рис. 6.10. Класифікація комп'ютерів

Суперкомп'ютер визначається тільки відносно — це найпотужніша обчислювальна система, яка діє у відповідний історичний період. У наш час найбільш відомі суперкомп'ютери "Cray" та "IBM SP2" (США). Модель "Cray", що випускається з початку 90-х років 20 ст., містить 16 процесорів і має швидкодію більшу за 10 MFLOPS; в моделі CS 6400 число процесорів доведено до 64. В наш час створюються супермаши-

ни на основі сотень і тисяч процесорів. Суперкомп'ютери вимагають стабільного температурного режиму. Вартість комп'ютера сягає десятків мільйонів доларів. Суперкомп'ютери використовують для вирішення крупних наукових і технічних задач.

Великі комп'ютери (мейнфрейми) — це машини загального призначення з продуктивністю до 400 тис оп/с, ємністю пам'яті — декілька мегабайт, розрядністю — 32 або 64 біт. Їх використовують у вузлах регіональних систем обробки даних, на великих виробництвах. Вартість мейнфреймів — біля одного мільйона доларів. До великих машин відносяться моделі ЕС-1065 (Росія) та IBM-370 (США).

Міні-комп'ютери з'явилися на початку 70-х років 20 ст., їхнє традиційне використання — керування технологічними процесами. Класичною міні-машиною є модель "PDP-11" фірми DEC. Вартість мінікомп'ютерів приблизно 100 тис доларів.

Мікрокомп'ютери будують на мікропроцесорах, які виконують функції процесора. Складові частини мікропроцесора розміщені в одній або декількох інтегральних схемах. Перший мікропроцесор розроблений фірмою Intel в 1971 р. З цього часу розвиток комп'ютерної техніки має новий кількісний і якісний характер. Вартість мікрокомп'ютерів складає декілька тисяч доларів. На основі мікрокомп'ютерів побудовані керуючі системи, автоматизовані робочі місця та персональні комп'ютери.

Виділяють два напрямки побудови комп'ютерів

- з повним набором складних і багатформатних команд CISC (Complex Instruction Set Computer),
- з обмеженим набором команд RISC (Reduced Instruction Set Computer), усі команди працюють з операндами, які розміщені в регістрах АЛП, при цьому команди мають однаковий формат.

Комп'ютери RISC працюють у 2–4 рази швидше, ніж CISC-комп'ютери при однаковій тактовій частоті.

Контрольні запитання

1. Дайте загальну характеристику комп'ютерної техніки.
2. Викладіть коротко історію розвитку комп'ютерної техніки.
3. Що розуміють під архітектурою комп'ютерів?
4. Нарисуйте класичну структуру комп'ютера.
5. Назвіть складові частини внутрішньої пам'яті комп'ютера.
6. Що відноситься до пристроїв введення-виведення інформації комп'ютерів?
7. Охарактеризуйте принципи програмного керування.
8. Поясніть основні характеристики і режими роботи комп'ютерів.
9. Що розуміють під поколіннями комп'ютерів?

Розділ 7

Пам'ять комп'ютерів

7.1. Загальна характеристика пам'яті

7.1.1. Функція пам'яті

Пам'яттю комп'ютера називається сукупність різних пристроїв, призначених для приймання, зберігання і видачі двійкової інформації. Окремий пристрій називається запам'ятовуючим (ЗП) або просто пам'яттю. Термін "запам'ятовуючий пристрій" вживають тоді, коли треба підкреслити принцип його побудови: на магнітному осерді, напівпровідниках та ін. Термін "пам'ять" застосовують, коли вказують на функцію, яку вона виконує: основну, постійну та ін.

Пам'ять комп'ютера функціонує під керуванням операційної системи, яка розміщує масиви інформації в пам'яті, забезпечує їхній захист від несанкціонованого доступу та виконує інші функції. Продуктивність і обчислювальні можливості комп'ютера значною мірою визначаються складом і характеристиками ЗП, які застосовуються.

Пам'ять сучасних комп'ютерів класифікують за функціональним призначенням, видом носія інформації, способом організації доступу до інформації. За функціональним призначенням пам'ять комп'ютерів поділяється на дві основні групи: зовнішню і внутрішню.

Зовнішні ЗП призначені для тривалого зберігання великих масивів інформації з ємністю до гігабайта і більше та малою швидкодією. Зовнішня пам'ять містить в собі накопичувачі на магнітних стрічках, дисках, барабанах та оптичних дисках.

Внутрішні ЗП призначені для зберігання програм і даних, які виконуються в поточний момент часу. До внутрішньої пам'яті відносяться:

- надоперативні (регістрові) ЗП, які використовують реєстри загального призначення процесора; вони мають невелику інформаційну ємність і швидкодію роботи процесора;
- кеш–пам'ять, яка служить для зберігання копій інформації, що використовуються в поточних операціях обміну. Висока швидкодія кеш–пам'яті підвищує продуктивність комп'ютера;
- оперативні, які характеризуються високою швидкодією та інформаційною ємністю до сотень мегабайт; оперативна пам'ять комп'ютерів перших поколінь будувалася на магнітних осердях, тепер же ОП реалізується на напівпровідникових ВІС ЗП. У процесі роботи інформація із зовнішньої пам'яті при необхідності переписується в оперативний ЗП (ОЗП);
- постійні, які будуються на напівпровідникових ВІС. У постійну пам'ять інформація записується заздалегідь і її можна тільки прочитувати. Оперативні й постійні ЗП утворюють основну пам'ять комп'ютера;
- спеціалізовані види пам'яті — багатопортові, асоціативні, відеопам'ять та ін.

За фізичним принципом побудови пам'ять комп'ютера буває:

- магнітна (на осерді та плівках, на циліндричних і плоских магнітних доменах);
- ультразвукова (магнітострикційна, електрострикційна);
- сегнетоелектрична і голографічна (лазерна), на основі надпровідності;
- напівпровідникова на ВІС і НВІС, ультра-ВІС.

Напівпровідникові ВІС ЗП в свою чергу характеризуються:

- технологією виготовлення: на біполярних транзисторах (ТТЛШ, ЕЗЛ, І²Л), на МОН-структурах (p -МОН, n -МОН, КМОН); серед новітніх розробок слід відмітити ЗП, де використані ПТШ на основі арсеніду галію;
- способом зберігання інформації — статичні та динамічні (у статичних ЗП елементом пам'яті є тригер, а у динамічних елемент пам'яті будують на конденсаторі і МОН-транзисторах);
- енергозалежністю: розрізняють енергозалежні ВІС ЗП, в яких при відключенні джерела живлення інформація, яка зберігається, руйнується (що справедливо в цей час для більшості напівпровідникових мікросхем пам'яті), і енергонезалежні (звичайно на сегнетоелектриках), в яких інформація зберігається;
- структурною організацією ВІС ЗП, яка символічно подається у вигляді $N \times m$, де N — кількість адресних одиниць інформації, що зберігаються; m — розрядність (організацію у вигляді $N \times 1$ називають однорозрядною, а $N \times m$ — словниковою, при цьому $m > 1$).

Елементний базис пам'яті сучасних комп'ютерів складають мікросхеми різного ступеня інтеграції. Основою будь-якого ЗП є елемент пам'яті (ЕП) статичного або динамічного типу, призначений для записування, зберігання і зчитування одного біта інформації — цифри 0 або 1. Сукупність ЕП, які утворюють n -розрядне слово, називають **коміркою пам'яті** (КП). Множина КП утворює запам'ятовуючий масив, який називається матрицею M елементів пам'яті.

7.1.2. Основні параметри пам'яті

Основними операціями в пам'яті є записування і зчитування певної одиниці інформації, наприклад, байта. Ці операції називаються також зверненням до пам'яті. Пам'ять характеризується інформаційною ємністю, фізичним об'ємом, питомою ємністю і вартістю, шириною вибірки, споживаною потужністю і швидкодією.

Інформаційна ємність E являє собою максимальний об'єм даних, який може одночасно зберігатися в пам'яті. Ємність виражається в бітах, байтах (8 біт = 1 байт), кілобайтах (2^{10} байт = 1 Кбайт), мегабайтах (2^{10} Кбайт = 1 Мбайт) і гігабайтах (2^{10} Мбайт = 1 Гбайт) (при цьому потрібно врахувати, що $2^{10} = 1024$).

Питома ємність визначається відношенням інформаційної ємності ЗП до його фізичного об'єму. Питома вартість — це відношення вартості ЗП до його інформаційної ємності. Ширина вибірки подається числом розрядів, які записуються в ЗП або зчитуються з нього за одне звернення.

Споживану потужність задають або для усього ЗП, або на зберігання одного біта інформації. Основними вимогами до пам'яті є максимально велика інформаційна ємність, висока швидкодія (малий час звернення: $t_{за} < 10$ нс), мінімальна споживана потужність (менша за 1 мкВт на 1 біт інформації, яка зберігається).

У наш час жоден з видів ЗП не задовольняє цих вимог повною мірою. Тому в пам'яті використовуються різні види ЗП, які розрізняються принципами побудови і своїми характеристиками.

Швидкодія ЗП вимірюється часом записування і зчитування та тривалістю відповідних їм циклів. **Час записування** t_{WR} — це інтервал між моментами появи керуючого сигналу записування і установленням КП в стан, який задають вхідні сигнали. **Час зчитування** — це інтервал між моментами появи керуючого сигналу читання t_{RD} і даних на виході пам'яті. Мінімально допустимий інтервал між послідовними читаннями t_{CYR} і записуваннями t_{CYW} створює відповідний цикл. Тривалість циклів може перевищувати час читання чи записування, оскільки після цих операцій необхідна додаткова затримка для встановлення початкового стану пам'яті. Як тривалість циклу звернення до пам'яті беруть величину $t_{CY} = \max(t_{CYW}, t_{CYR})$.

7.1.3. Вхідні та вихідні сигнали мікросхеми пам'яті

Мікросхеми ОП мають типові виводи, на яких діють визначені адресні, інформаційні та керуючі сигнали (рис. 7.1, а).

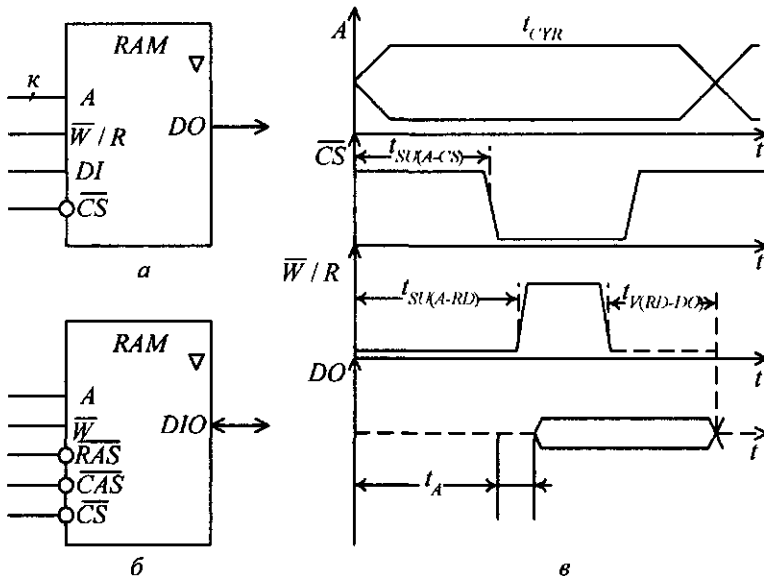


Рис. 7.1. Мікросхеми ОП: а, б — умовні графічні позначення; в — часові діаграми сигналів

Призначення виводів і сигналів на них такі:

- A (Address) — входи адреси, розрядність якої k визначається співвідношенням $k = \log_2 N$, де $N = 2^k$ — максимально можливе число даних (біт, байт, слів), що зберігаються в пам'яті і адресуються як єдине ціле;
- DI (Data Input) — шина вхідних даних;
- DO (Data Out) — шина вихідних даних;
- \overline{W}/R (Write/Read) — сигнал записування даних при $\overline{W}/R = 0$ або зчитування при $\overline{W}/R = 1$;
- \overline{CS} (Chip Select) або \overline{CE} (Chip Enable) — сигнал дозволу при $\overline{CS} (\overline{CE}) = 0$ чи заборони, якщо $\overline{CS} (\overline{CE}) = 1$, роботи даної мікросхеми.

Особливістю роботи динамічних ЗП є мультиплексування ША (рис. 7.1, б). Адреса, наприклад, $A=A_{15}, A_{14}, \dots, A_0$ ділиться на старшу напівадресу $A_x=A_{15}, A_{14}, \dots, A_8$ і молодшу $A_y=A_7, A_6, \dots, A_0$. Напівадреси подаються на одні й ті ж входи адреси мікросхеми пам'яті. Подавання напівадреси A_x супроводжується сигналом \overline{RAS} (Row Address Strobe), а напівадреси A_y — сигналом \overline{CAS} (Column Address Strobe). Такий спосіб адресації зменшує число виводів корпусу ІМС. Часто виводи DI і DO об'єднуються у спільний вивід DIO .

7.1.4. Часові характеристики мікросхем пам'яті

Вимоги до взаємного часового положення двох сигналів ($A-B$) задають такими параметрами:

- часом попереднього установлення $t_{SU(A-B)}$ сигналу A відносно сигналу B , тобто інтервалом між початками обох сигналів;
- часом утримання $t_H(A-B)$ — інтервалом часу між початком сигналу A і закінченням сигналу B ;
- часом зберігання $t_V(A-B)$ — інтервалом між закінченнями сигналів A і B .

Тривалість сигналів позначається як t_W (Width — ширина).

Для ЗП характерна така послідовність сигналів у часі (рис. 7.1, в): спочатку адреса, потім вибірка мікросхеми \overline{CS} , потім строб записування–читання $\overline{W/R}$. Індексами A (Access) позначають інтервали часу від появи керуючого сигналу до появи даних на виході (рис. 7.1, в).

7.1.5. Способи доступу до даних у напівпровідниковій пам'яті

У напівпровідникових ЗП виділяють адресні, послідовні й асоціативні способи доступу до даних (рис. 7.2).

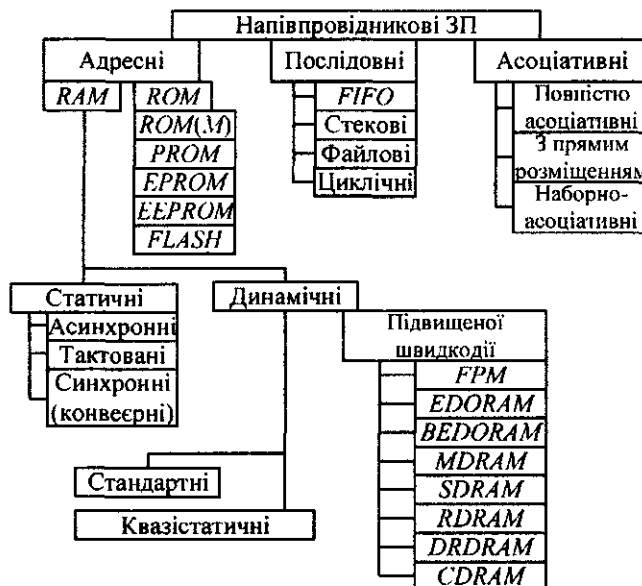


Рис. 7.2. Класифікація напівпровідникових ЗП

При адресному доступі адресний код указує номер комірки пам'яті, з якою має проводитися обмін. Усі комірки в момент звернення рівнодоступні. До адресних ЗП відносяться

- *RAM (Random Access Memory)*, українські синоніми ОЗП (оперативний ЗП) або ЗПДВ (ЗП з довільною вибіркою),
- *ROM (Read Only Memory)*, український термін — ПЗП (постійні ЗП)

Оперативні ЗП зберігають дані, необхідні при виконанні поточної програми, вони можуть бути змінені в будь-який момент часу. Оперативні ЗП в більшості є енергозалежні. У постійних ЗП вміст комірок або взагалі не змінюється або змінюється рідко в спеціальних режимах.

Запам'ятовуючі пристрої *RAM* поділяються на статичні *SRAM (Static RAM)* і динамічні *DRAM (Dynamic RAM)*. У статичних *RAM* елементами пам'яті є тригери. Вони зберігають свій стан, поки схема має напругу живлення і нові дані не записуються. У динамічних *RAM* дані зберігаються у вигляді зарядів конденсаторів, створюваних компонентами МОН-транзисторів. Саморозряд конденсаторів веде до руйнування даних, тому вони періодично (кожні 2–30 мс) мають регенеруватися. Але щільність упакування динамічних ЕП перевищує в 4–5 разів такий же показник для статичних *RAM*. Регенерація даних здійснюється за допомогою спеціальних контролерів. Розроблені також *DRAM* із внутрішніми схемами регенерації, такі ЗП називаються квазі-статичними.

Статичні ОЗП розподіляють на такі типи:

- асинхронні — керуючі сигнали можна задавати як імпульсами, так і рівнями,
- тактовані — в них деякі сигнали мають бути обов'язково імпульсами, наприклад, сигнал дозволу роботи \overline{CS} ,
- синхронні, в яких організований конвеєрний канал передачі даних, що синхронізується від тактової системи процесора.

Динамічні ЗП характеризуються найбільшою інформаційною ємністю і невисокою вартістю, тому вони використовуються як основна пам'ять комп'ютерів. Розроблені варіанти схем *DRAM* високої швидкодії, які подані в класифікації і детально розглянуті в п. 7.7. Статичні ЗП в 4–5 разів дорожчі динамічних і приблизно у стільки ж разів менша їхня інформаційна ємність. Їхнім достоїнством є висока швидкодія, а типовою областю застосування — схеми кеш-пам'яті.

Постійна пам'ять типу *ROM(M)* програмується при виготовленні за допомогою масок, тому її називають ПЗП масочним. В подальших різновидах *ROM* у позначеннях є буква *P* (від *Programmable*). Це — пам'ять, що одноразово програмується користувачем — *PROM* (в українській термінології ППЗП — програмовані ПЗП) та багаторазово програмується — *EPROM*, *EEPROM* (детально див. п. 7.3).

Пам'ять типу *Flash* по ЕП подібна до *EEPROM* (інакше E^2PROM), але має структурні й технологічні особливості, які дозволяють виділити її в окремий тип.

У ЗП з послідовним доступом дані, що записуються, створюють чергу. Зчитування виконується слово за словом в порядку записів або навпаки. Прямий порядок зчитування використовується в буферах *FIFO* з дисципліною "перший прийшов — перший вийшов" (*First In — First Out*), а також у файлових і циклічних ЗП.

Різниця між пам'яттю *FIFO* і файловим ЗП полягає в тому, що у *FIFO* записування у пустий буфер зразу доступне для читання (тобто поступає в кінець ланцюга

моделі ЗП). У файлових ЗП дані поступають у початок ланцюга і з'являються на виході після деякого числа звертань, яке дорівнює числу елементів у ланцюзі.

У циклічних ЗП слова доступні одне за одним з постійним періодом, який визначається ємністю пам'яті. До них відноситься відеопам'ять (*VRAM*).

Зчитування в оберненому порядку властиве стековим ЗП з дисципліною "останній прийшов — першим вийшов". Такі ЗП називаються буферами *LIFO* (*Last In — First Out*).

Час доступу до конкретної одиниці інформації, що зберігається в послідовних ЗП, є випадковою величиною. В найгіршому випадку для такого доступу треба переглянути весь об'єм інформації, що зберігається у цій пам'яті.

Асоціативний доступ реалізує пошук інформації за деякою ознакою, а не за адресою. В найбільш повній версії всі слова, які зберігаються в пам'яті, можуть одночасно перевірятися на відповідність ознаці, наприклад, на збіг визначених полів слів — тегів (від *tag*) за ознакою, яку задає вхідне слово (тегова адреса). На вихід передаються слова, які задовольняють ознаку. Дисципліна видавання слів, якщо тегу задовольняє декілька слів, та дисципліна записування нових даних можуть бути різними. Основна область використання асоціативної пам'яті в комп'ютерах — кешування даних.

7.2. Основні структури напівпровідникової пам'яті

7.2.1. Поняття структури пам'яті

Кожна матриця M в пристрої пам'яті має систему адресних і розрядних ліній (провідників). Адресні (словникові) лінії служать для виділення за адресою будь-якої КП. Сукупність різних адресних кодів утворює адресний простір пам'яті. Розрядні лінії записування (ЛЗП) служать для введення в кожний розряд вибраної КП цифри 0 або 1 відповідно до вхідної інформації. Розрядні лінії зчитування (ЛЗЧ) служать для знімання інформації, яка зберігається, з розряду вибраної КП. Часто використовують спільну лінію записування–зчитування (ЛЗЗ). Адресні та розрядні лінії разом називаються лініями вибірки. Якщо довжина адресного коду дорівнює k , то кількість слів N , які зберігаються в пам'яті як окремі одиниці даних, визначаються із співвідношення $N = 2^k$.

Структуру пам'яті визначає спосіб розподілу КП між адресними та розрядними лініями. За цією ознакою виділяють такі структури пам'яті: $2D$, $3D$, $2,5D$ і модифіковану — $2DM$ (D від *Dimension* — розмірність).

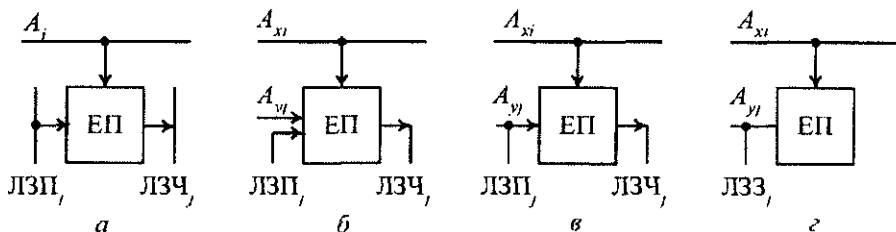


Рис. 7.3. Узагальнене поняття структури пам'яті: а — $2D$; б — $3D$; в — $2,5D$; г — $2DM$

У системі $2D$ кожний ЕП має одну адресну лінію A , (одне D), лінії записування ЛЗП, і зчитування ЛЗЧ, які спільно утворюють друге D (рис. 7.3, а). У структурі $3D$ адресу розділяють на дві частини: старша A_x визначає адреси рядків, а молодша A_y — адреси стовпців; разом вони утворюють $2D$. Лінії записування і зчитування утворюють третє D (рис. 7.3, б). У структурі $2,5D$ одна з ЛЗП або ЛЗЧ суміщена з напівадресою A_x або A_y (рис. 7.3, в). У модифікованій системі $2DM$ використовується спільна лінія ЛЗЗ j , яка поєднується з адресною лінією A_y . Розглянуті структури характерні для статичних ОЗП і пам'яті типу ROM. Структури динамічних ОЗП мають свою специфіку.

7.2.2. Пам'ять із структурою $2D$

Організація мікросхеми ОЗП із структурою $2D$ показана на рис. 7.4.

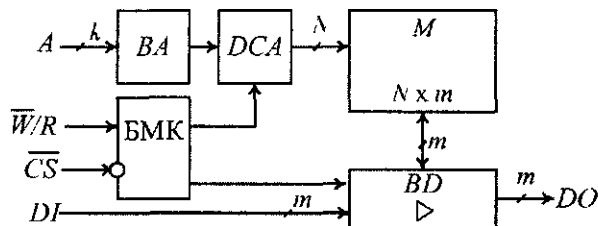


Рис. 7.4. Пам'ять із структурою $2D$

До складу мікросхеми пам'яті входять:

- матриця елементів пам'яті M , яка містить N рядків і m стовпців (за числом розрядів слова);
- буфер BA і дешифратор DCA адреси з числом виходів $N = 2^k$;
- буферні формувачі вхідних DI і вихідних DO інформаційних сигналів у режимах записування і зчитування й BD ;
- блок місцевого керування (БМК).

При звертанні до пам'яті вибираються ЕП, які розміщені на збудженому виході дешифратора адреси DCA . Записування даних здійснюється при значенні сигналу $\overline{W}/R = 0$, а зчитування при $\overline{W}/R = 1$. Ємність пам'яті $2D$ дорівнює $E = N \times m$ біт. Організація матриці M пам'яті із структурою $2D$ (три рядки і три стовпці) показана на рис. 7.5.

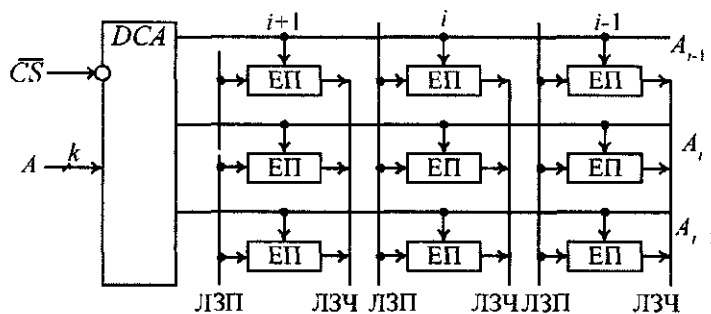


Рис. 7.5. Організація матриці M пам'яті із структурою $2D$

Недоліком структури 2D є складність побудови дешифратора адреси з числом виходів N , яке дорівнює числу слів, що зберігаються у пам'яті. Тому структура типу 2D використовується в ЗП малої інформаційної ємності.

7.2.3. Пам'ять із структурою 3D

У пам'яті із структурою 3D адресний код розділяється на дві рівні частини A_x і A_y (для парного k), кожна з яких декодується окремими дешифраторами відповідно DCX і DCY (рис. 7.6).

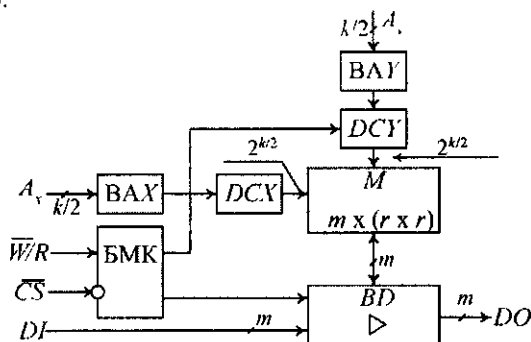


Рис. 7.6. Пам'ять із структурою 3D

Матриця M складається з m підматриць за числом розрядів слова. Кожна матриця зберігає значення свого i -го розряду всіх N слів. Кожна підматриця є квадратною: r рядків і r стовпців, що записується як $r \times r$, при цьому $r = \sqrt{N} = 2^{k/2}$ (рис. 7.7).

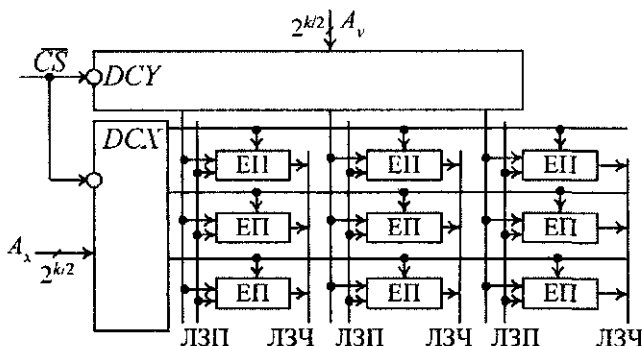


Рис. 7.7. Організація підматриці $r \times r$ пам'яті із структурою 3D

При звертанні до пам'яті вибирають m запам'ятовуваних елементів (по одному з кожної підматриці), які лежать на перетині рядка і стовпця збуджених виходів дешифраторів DCX і DCY . Така структура часто використовується і з однорозрядною організацією $N \times 1$ біт. Для цієї пам'яті ємністю 1 К слів треба мати два дешифратори з числом виходів у кожному $N = 2^5 = 32$. Для пам'яті із структурою 2D з тією самою ємністю дешифратор має 1024 виходи.

Недоліком структури 3D є використання складних ЕП, які допускають двокоординатну вибірку та складнішу структуру матриці M . У пам'яті за модифікованою структурою 2DM поєднуються переваги структур 2D і 3D.

7.2.4. Пам'ять із структурою 2DM

У ЗП типу *SRAM*, *ROM* із структурою 2DM адресний код довжиною k розбивається на дві частини: $A_x = A_k A_{k-1} \dots A_{r+1}$, що поступає на дешифратор рядків *DCX* та $A_y = A_r A_{r-1} \dots A_1$, що подається на входи мультимплексорів *MUX* з організацією "2^k→1" (рис. 7.8).

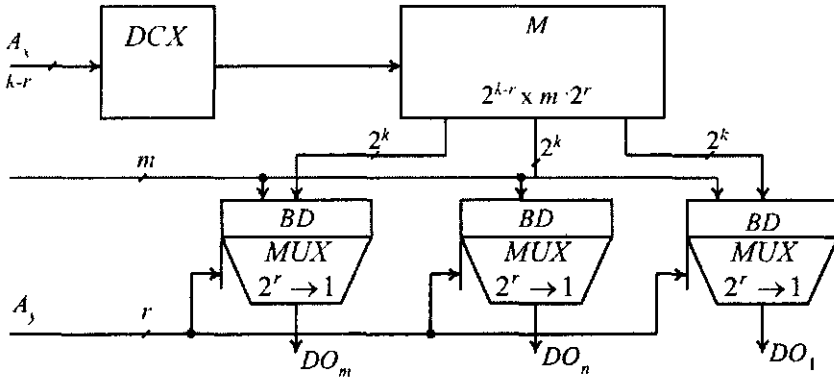


Рис. 7.8. Пам'ять *SRAM* і *ROM* зі структурою 2DM

Дешифратор *DCX* обслуговує 2^{k-r} рядків, кожний з яких зберігає 2^k m -розрядних слів. У кожній групі зберігаються значення однойменних розрядів і обслуговується вона своїм буфером *BD* і мультимплексором *MUX* з організацією "2^k→1" (рис. 7.9).

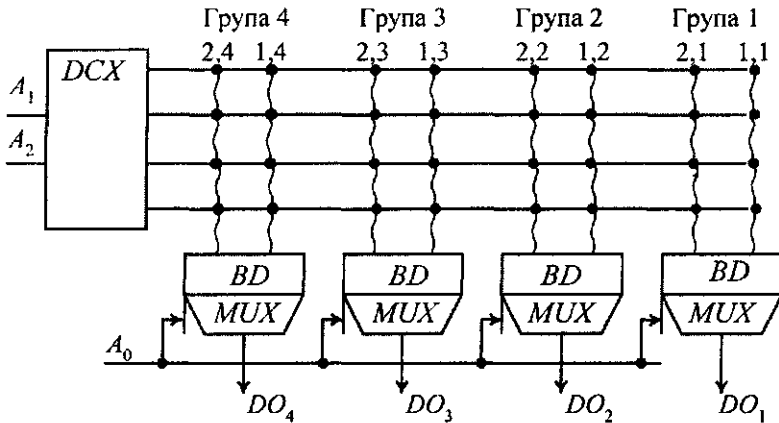


Рис. 7.9. Матриці *M* з організацією 4 x 4·2

Таким чином, активізований вихід дешифратора *DCX* вибирає рядок, а молодші розряди адреси за допомогою мультимплексорів забезпечують формування вихідного слова *DO* (по одному розряду з кожної групи).

Структура 2DM для ЗП типу *RAM* в загальному вигляді показана на рис. 7.10. 3 матриці *M* зчитується "довгий" рядок. Дані у відповідні групи рядка записуються або зчитуються буферами даних *BD*, які керуються сигналами з виходів дешифратора *DCY*. Буфер *BD* також визначає напрямок обміну даними за допомогою сигналу \overline{W}/R .

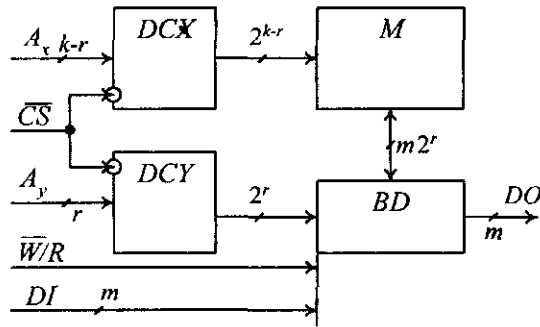


Рис. 7.10. Пам'ять типу RAM із структурою 2DM

Організація пам'яті із структурою 2DM є найбільш розповсюдженою, особливо для мікросхем великої ємності.

7.2.5. Пам'ять з послідовним доступом

Представниками пам'яті з послідовним доступом є буфер FIFO і стек. Буфер FIFO — це пам'ять для зберігання черг даних (списків). Слова з черги вибираються в порядку їхнього надходження. Моменти записування слів у буфер і зчитування з нього задаються зовнішніми керуючими сигналами незалежно один від одного. Дані поступають в темпі джерела інформації, а зчитуються з частотою, необхідною для приймача. Нове слово ставиться в кінець черги, а зчитування здійснюється з початку черги. У схемі (рис. 7.11) перед початком роботи обидва лічильники адрес записування CTR і читання CTW скидаються.

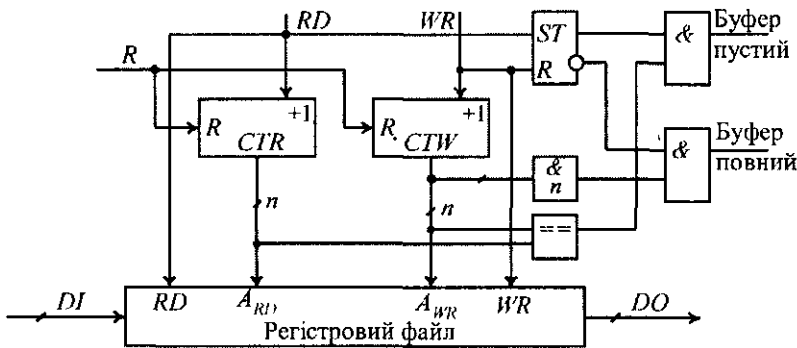


Рис. 7.11. Структура буфера FIFO

При записуванні даних адреси збільшуються на одиницю при кожному звертанні. Те ж відбувається і при читанні даних. Адреси порівнюються при читанні компаратором. За допомогою схеми збігу визначається момент повного завантаження буфера. Ці перевірки автоматично проводяться при поданні сигналів RD або WR . Якщо буфер повний, то треба припинити приймання даних, а якщо пустий — то припинити читання.

7.3. Кеш-пам'ять

7.3.1. Загальна характеристика кеш-пам'яті

Кеш-пам'ять (від *Cache* — тайник) — це засіб копіювання і зберігання блоків даних основної пам'яті типу *DRAM* в процесі виконання програми. Кеш-пам'ять побудована на швидкодіючих тригерних ЕП, але має невелику ємність порівняно з основною динамічною пам'яттю. Кеш зберігає обмежене число даних і тегів. Тег містить інформацію про фізичну адресу і стан даних.

При кожному зверненні до основної пам'яті спеціальний контролер перевіряє за тегом наявність цієї копії в кеші. Якщо вона є, то виробляється сигнал *Hit* (кеш-попадання) і звернення відбувається до кеш-пам'яті (рис. 7.12).

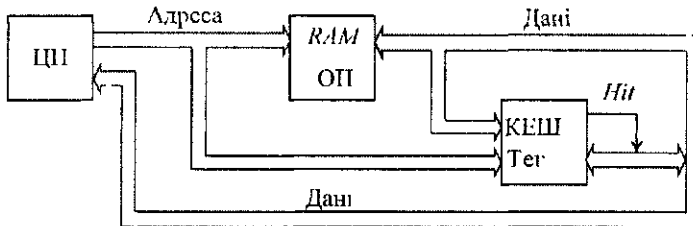


Рис. 7.12. Структура кеш-пам'яті

Якщо копії немає (кеш-промах), то сигнал *Hit* не виробляється і виконується читання з ОП та одночасне розміщення зчитаних даних в кеші.

Обмін з ОП може відбуватися двома способами:

- перший: звернення до ОП поєднується з одночасним пошуком інформації в тегу. Звернення при попаданні до ОП анулюється;
- другий: звертання до ОП проводиться тільки після виявлення кеш-промаху.

У сучасних комп'ютерах кеш будують за дворівневою схемою:

- первинний кеш (*L1 Cache*) має об'єм десятки Кбайт і вбудовується в процесор. Для підвищення продуктивності часто використовуються роздільні кеші для команд і даних (Гарвардська архітектура);
- вторинний кеш (*L2 Cache*), звичайно встановлюють на системній платі, він має об'єм декілька Мбайт.

Більшість прикладних програм має циклічний характер і багаторазово використовує одні й ті самі дані, тому наявність кеша зменшує кількість звернень до відносно повільної ОП.

7.3.2. Повністю асоціативний кеш

Залежно від способу визначення відповідності між даними в кеші й ОП використовують такі структури кеш-пам'яті:

- повністю асоціативний кеш;
- кеш з прямим розміщенням;
- набірно-асоціативний кеш.

У повністю асоціативній кеш-пам'яті кожна комірка зберігає дані, а в полі тег знаходиться повна фізична адреса одиниці інформації, копія якої записана (рис. 7.13).

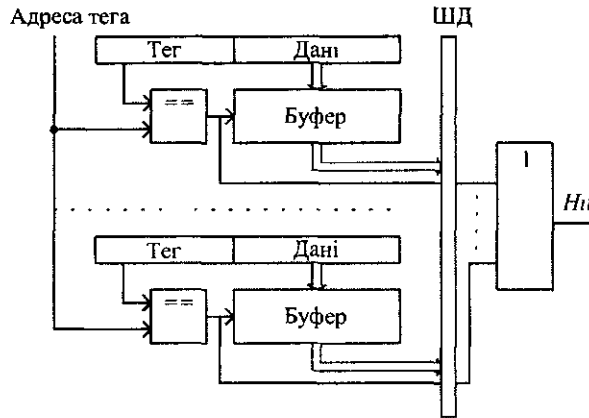


Рис. 7.13. Структура повністю асоціативного кеша

Під час обмінів фізична адреса даних, яка запитується, порівнюється з полем тег всіх комірок. Якщо виявляється збіг з адресою будь-якої комірки, то вона виставляє сигнал *Hit*. При читанні, коли значення сигналу *Hit* = 1, дані видаються на ШД. Якщо збігу немає (*Hit* = 0), то при читанні з ОП дані разом з адресою розміщуються у вільній або давно не використовуваній комірниці кеш-пам'яті.

Рядки кеша завантажуються адресами та даними при операціях читання з ОП. Дані без копії в кеші записуються тільки в ОП. Дані, що мають копії в кеші, записуються в ОП наскрізним чи оберненим способом.

Під час наскрізного записування дані одночасно завантажуються в копію кеша та в ОП. Витрачається час на відносно довгу тривалість записування в *DRAM*, що знижує швидкодію всієї пам'яті.

Під час оберненого записування дані спочатку записуються в свою копію і помічаються ознакою модифікації. Після звільнення системної шини кеш-контролер переписує модифікований рядок в ОП. Обернений спосіб записування складніший від наскрізного, але ефективніший. В обох випадках кеш-контролер забезпечує когерентність, тобто узгодження даних в кеші та ОП.

Повністю асоціативна кеш-пам'ять забезпечує найбільшу функціональну гнучкість, але є дуже складним пристроєм.

7.3.3. Кеш-пам'ять з прямим розміщенням

В кеш-пам'яті з прямим розміщенням ОП умовно розбивається на сторінки, наприклад, 512. Розмір кожної сторінки збігається з розміром кеш-пам'яті. Дані в кеші зберігаються в рядках, які є наборами байтів (часто 32 байт). Якщо кеш-пам'ять має, наприклад, 256 рядків, то її об'єм дорівнює $256 \times 32 = 8$ Кбайт. При цьому об'єм ОП має складати $512 \times 8 \text{ К} = 4$ Мбайт. В рядках кеша можуть зберігатися дані з будь-якої сторінки ОП.

Адреса від процесора ділиться на три частини (рис. 7.14).

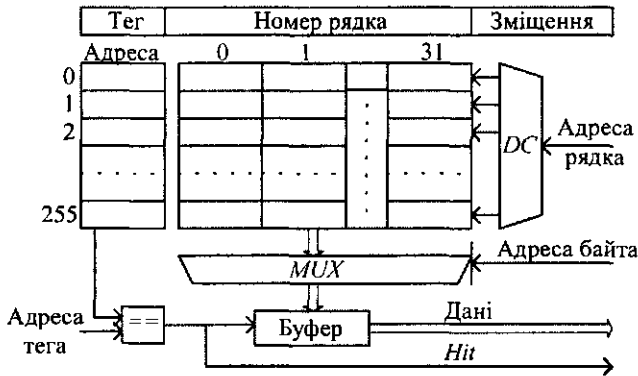


Рис. 7.14. Організація кеш-пам'яті з прямим розміщенням

Молодші розряди (зміщення) визначають положення молодшого байта 32-розрядного слова. Середні розряди адреси задають номер рядка. Разом номери рядка і байта в ньому називаються індексом. Інформація про номер сторінки ОП, дані з якої займають відповідний рядок у кеші, називається тегом.

Кожний тег займає комірку пам'яті, яка зв'язана зі своїм рядком. Сукупність цих комірок створює пам'ять тегів. Розрядність комірок тега має бути достатньою для запису номера сторінки ОП. Наприклад, для 256 сторінок довжина комірки тега має бути 8 біт.

При зверненні до ОП спочатку зчитується рядок кеша із заданим індексом, а потім порівнюються значення тегів даного рядка і заданого в адресі від процесора. При кеш-попаданні сигнал $Hit = 1$ і слово даних мультиплекуються з вибраного рядка і пересилаються в процесор. При кеш-промаху сигнал $Hit = 0$ і дані вибираються з ОП; при цьому за вказаною адресою завантажується весь рядок кеша (пакетний обмін).

Кеш з прямим розміщенням має просту схему, малу розрядність тега, але має й суттєвий недолік — не допускається розміщення в кеші рядків з однаковими індексами, але різними тегами. Це призводить до безперервної черги кеш-промахів. Цей недолік усувається в набірно-асоціативному кеші.

7.3.4. Набірно-асоціативний кеш

У набірно-асоціативній структурі кеш-пам'ять ділиться на набори з невеликим числом рядків, кратних двійці, тобто 2, 4, 8 і т.д. Середні розряди адреси від процесора визначають не один рядок, а весь набір.

Кожний рядок в наборі обслуговується власним блоком кеш-пам'яті, тегом, компаратором та буфером даних. Це подібно паралельній і узгодженій роботі декількох каналів прямого заміщення. Контролер кеш-пам'яті приймає рішення про те, в якому з рядків наборів розміщується черговий пакет даних.

У простому випадку кожний пакет даних з ОП може завантажуватися в один з двох рядків у наборі. Такий кеш містить два блоки кеш-пам'яті: A — для парних рядків і B — для непарних (рис. 7.15). Такі кеші називаються двохходовими.

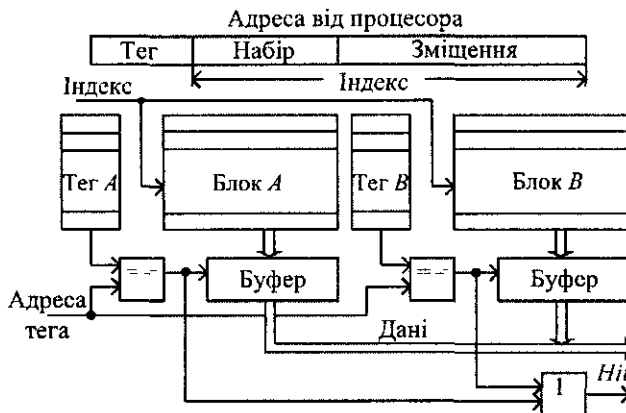


Рис. 7.15. Двовходовий набірно-асоціативний кеш

Сторінку даних з ОП можна помістити тільки в той набір, номер якого дорівнює адресі сторінки за модулем, наприклад, 64 або 128 і так далі. Місце сторінки в наборі довільне.

Одночасно зчитуються і порівнюються парні і непарні рядки (слова з них). Зчитування даних іде від того блока, де є збіг тега і тегової адреси від процесора. При відсутності збігів виконується звертання до ОП і заміщення рядка в одному з блоків кеша.

7.4. Постійна пам'ять

7.4.1. Загальна характеристика постійної пам'яті

Постійна пам'ять призначена для збереження програм, констант, табличних функцій та іншої інформації, яка записується заздалегідь і не змінюється в процесі поточної роботи комп'ютера. Вона застосовується також у перетворювачах кодів, знакогенераторах, у мікропрограмних пристроях керування. Загальним для всіх мікросхем постійної пам'яті є енергонезалежність, словникова організація і використання режиму зчитування як основного.

Мікросхеми постійної пам'яті розділяються на такі групи:

- ПЗП або *ROM (Read Only Memory)* — програмуються одноразово заводом-виготовлювачем, часто називаються масочними;
- ППЗП або *PROM (Programmable ROM)* — програмуються одноразово електричним способом користувачем;
- РПЗП-УФ або *EPROM (Erasable PROM)* — програмуються багаторазово (ре-програмуються) з ультрафіолетовим стиранням і електричним записуванням;
- РПЗП-ЕС або *EEPROM (Electrical EPROM)* — програмуються і стираються багаторазово електричним способом.

7.4.2. Мікросхеми постійних запам'ятовуючих пристроїв

У мікросхемах ПЗП і ППЗП елементами пам'яті є діоди, біполярні й МОН-транзистори, а також КМОН структури. Елементи пам'яті розташовуються у вузлах матриці, утворених адресними лініями рядка X_i і стовпця Y_j . Стан "1" відповідає з'єднанню ЕП із розрядною лінією, а стан "0" означає відсутність зв'язку (для прямих виходів мікросхеми пам'яті). У масочних ПЗП розрив з'єднань ЕП із розрядними лініями при записуванні інформації виконує завод-виготовлювач (рис. 7.16).

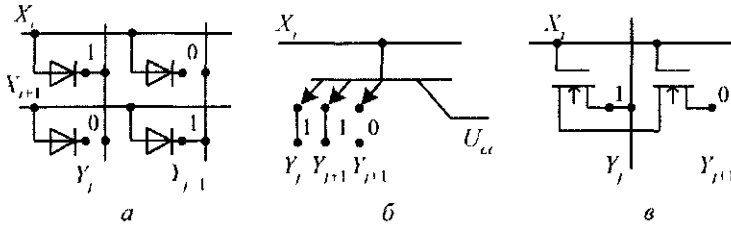


Рис. 7.16. Елемент пам'яті масочних ПЗП:

а — на діодах; б — на багатомітерних транзисторах; в — на МОН-транзисторах

Характеристики ряду серій ВІС ПЗП приведені в таблиці 7.1, а їхні умовні позначення показані на рис. 7.17.

Таблиця 7.1

Тип мікросхеми	Ємність, біт	Технологія	Час вибірки — t_A , нс	P_{CC} , мВт
KP1656PE4	8Kx8	ТТЛШ	40	90
KP568PE3	16Kx8	п-МОН	300	300
KM568PE5	128Kx8	п-МОН	200	300
K536PE2	32Kx8	КМОН	500	20

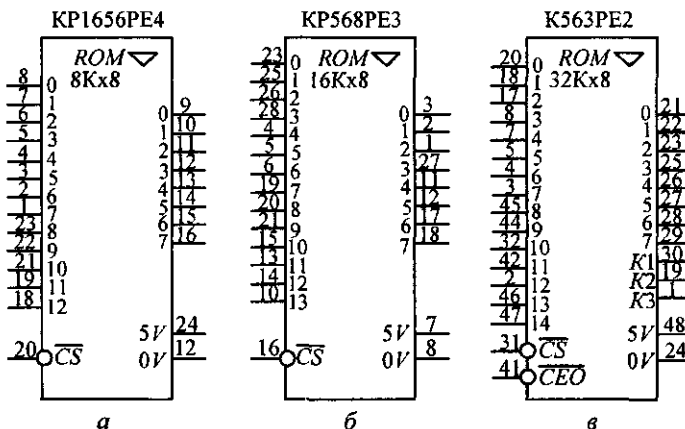


Рис. 7.17. Умовні позначення ВІС ПЗП

Мікросхема K563PE2 має вмонтовану схему самоконтролю і виправлення оди-ночних помилок за допомогою коду Хемінга із встановленням ознаки-помилки на ви-ході $K1$. Коректор можна виключити сигналом $K2 = 0$, при цьому дані зчитують без виправлення помилок.

7.4.3. Мікросхеми програмовних постійних запам'ятовуючих пристроїв

У мікросхемах ППЗП, які поставляються заводом виготовлювачем, усі ЕП, розташовані у вузлах матриці, сполучені зі стовпчиками плавкими перемичками (рис 7.18).

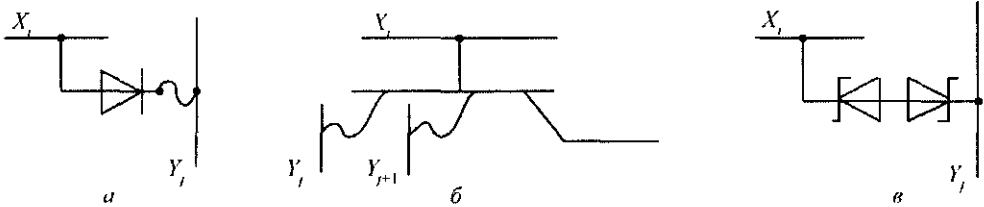


Рис. 7.18. Елемент пам'яті мікросхем ППЗП;

а — на діодах; б — на багатоемітерних транзисторах; в — на діодах Шотки

Записування інформації в ППЗП здійснюється користувачем перепалюванням перемичок (рис.7.18, а, б) або електричним пробоем діодів Шотки (рис. 7.18, в). Для плавких перемичок використовують тонкі плівки з ніхрому або полікристалічного кремнію; струм перепалювання дорівнює 50–100 мА, в результаті чого перемичка необоротно руйнується.

Структура ППЗП з ЕП на багатоемітерних транзисторах показана на рис. 7.19. Пам'ять побудована за системою 2,5D з організацією 4 x 2 біт. Кожний багатоемітерний транзистор в матриці являє собою два ЕП (за числом емітерів) і програмується на запам'ятовування дворозрядного слова.

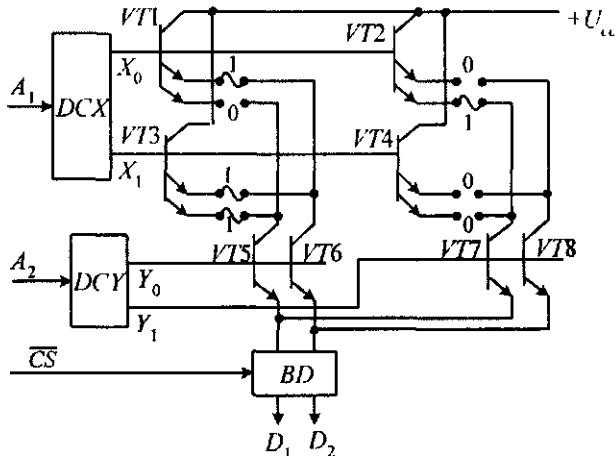


Рис. 7.19. Структура ПЗП з організацією 4 x 2 біт

В ЕП на транзисторах $VT1$ – $VT4$ запрограмовані для зберігання відповідно слова 01,10,11 і 00.

Нехай значення адреси рядка $X_0 = 1$ ($A_1 = 0$) і стовпця $Y_1 = 1$ ($A_2 = 1$); при цьому відкриваються транзистори $VT2$ і $VT7$ і через них протікає в буфер струм, відображаючий лог. 1. Транзистор $VT8$ — закритий, оскільки його колектор не має зв'язку з емітером $VT2$, який відображає лог. 0.

У результаті зчитування на виході буфера BD формується слово $D_2D_1 = 01$. Число транзисторів, які підключаються до кожного виходу адреси стовпчика, визначається розрядністю слова, що зберігається.

Мікросхема ППЗП за структурою аналогічна масочним, але припускає однократне програмування користувачем. Найбільш поширеними є мікросхеми ППЗП серії К556, виготовлені за ТТЛШ-технологією. Функціональний склад цієї серії включає мікросхеми ємністю від 1 К до 64 Кбіт із словниковою (чотирьох і восьмирозрядною) організацією з часом вибірки 45–85 нс і споживаною потужністю 0,6–1 Вт (рис. 7.20).

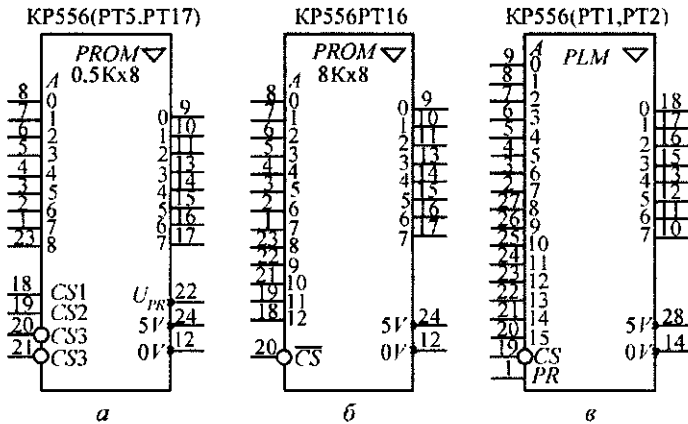


Рис. 7.20. Умовне позначення ВІС ППЗП

Різновидом ППЗП є програмовні логічні матриці (ПЛМ або PLM), до яких відносяться мікросхеми КР556(РТ1, РТ2). Вони мають ідентичні характеристики і конструктивні параметри, але різні за типом виходу: РТ1 мають вихід з відкритим колектором, РТ2 — з трьома станами (рис. 7.20, в). Обидві мікросхеми призначені для реалізації пристроїв, які виконують логічні операції над двійковими змінними.

Аналізовані ПЛМ розраховані на 16 вхідних змінних, інверсій від яких одержують усередині мікросхеми. Функціональні можливості даних ПЛМ: вісім логічних сум, кожна з яких може включати до 48 логічних добутоків із 16 змінних та їх інверсій.

Програмування ПЛМ полягає в перепалюванні плавких перемичок у потрібних вузлах. Це виконують вмонтовані в ПЛМ спеціальні схеми, які керуються сигналом PR . Спочатку програмують матрицю I , потім матрицю $4M$ і вихідні підсилювачі. Для керування доступом до мікросхеми використовують сигнал \overline{CS} , наявність якого дозволяє також нарощувати число вхідних змінних і вихідних функцій способом об'єднання декількох ПЛМ.

Широко застосовуються мікросхеми ПЛМ, які програмуються способом замовленого фотошаблону. Такі мікросхеми включені до комплекту деяких мікропроцесорних серій ВІС як ПЗП мікрокоманд.

Характеристики найбільш поширених мікросхем РПЗП-ЕС на МНОН-транзисторах наведені в табл. 7.2., а умовні графічні позначення деяких із них показані на рис. 7.21.

Здатність до багаторазового програмування забезпечується застосуванням ЕП з властивостями керованих перемичок. Цю функцію виконують транзистори типу ЛІЗМОН або МНОН. Вони являють собою спеціальні МОН-транзистори, у яких об-

ласть під затвором і підкладкою може накопичувати і зберігати заряд, який створюється електричним способом.

Таблиця 7.2

Тип мікросхеми	Тип транзистора	Ємність, біт	t , мкс	$P_{\text{сст}}$ мВт	$U_{\text{сст}}$ В
КР558РР1	Р-МНОН	256х8	5,0	300	5–12
КР1601РР1	Р-МНОН	1Кх4	1,8	625	5–12
КР1601РР3	Р-МНОН	2Кх8	1,6	825	5–12
КР558РР2	п-МНОН	2Кх8	0,35	480	5
КР558РР3	п-МНОН	8Кх0,4	0,4	400	5
КР1611РР1	п-МНОН	8Кх8	0,3	850	5

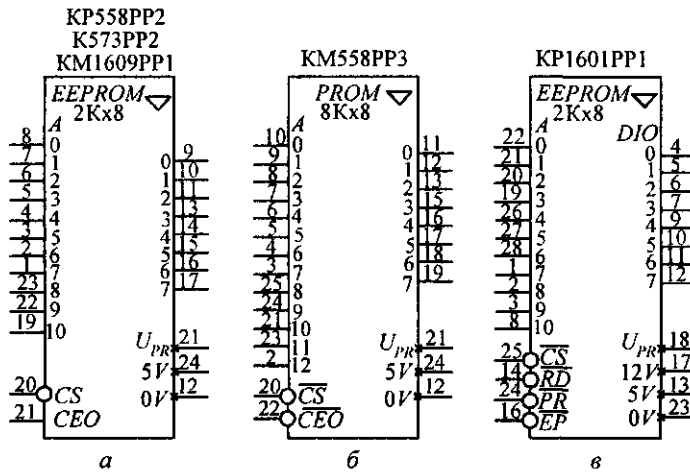


Рис. 7.21. Умовні позначення мікросхем РПЗП-ЕС на МНОН-транзисторах

Для програмування ППЗП і репрограмовних ПЗП використовують спеціальні пристрої — програматори. Для ряду мікросхем пам'яті програмування виконують за допомогою комп'ютера.

7.4.4. Мікросхеми РПЗП-ЕС на МНОН-транзисторах

Для побудови пам'яті типу РПЗП-ЕС широко використовують транзистори МНОН (від слів метал-нітрид-оксид-напівпровідник) з двошаровим підзатворним діелектриком. На поверхні кристала знаходиться тонкий шар діоксид кремнію SiO_2 , далі — більш товстий шар нітриду кремнію Si_3N_4 , а потім уже металевий затвор (рис. 7.22, а).

Елемент пам'яті на МНОН-транзисторі працює в таких режимах: програмування, зберігання, зчитування і стирання інформації.

Програмуванням називається процес занесення заряду під затвор транзистора. При цьому до затвору *n*-канального МНОН-транзистора, в якому створюється заряд, прикладають позитивний імпульс напруги амплітудою біля 20 В. Під дією сильного електричного поля електрони набувають достатню енергію і тунелюють з підкладки через тонкий шар оксиду товщиною біля 5 нм в шар нітриду, де вони захоплюються

"пастками". В нітриді з'являється нерухомий негативний заряд, який виконує функцію носія інформації. Вважають, що наявність заряду відображає лог. 0, а його відсутність — лог. 1. Транзистор, у якому заряд відсутній, відкривається робочим сигналом. В n -канальних МНОН-транзисторах заряд екранує дію позитивної напруги на затворі і відповідно підвищує порогову напругу настільки, що робочий сигнал не може відкрити транзистор (рис. 7.22, б).

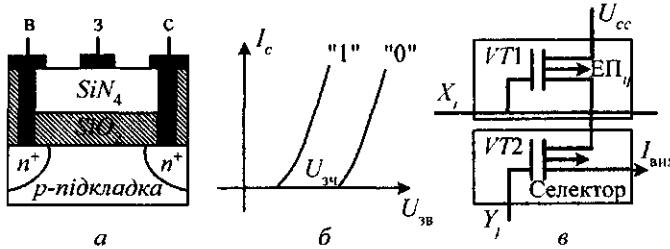


Рис. 7.22. Елементи пам'яті на МНОН-транзисторі:

а — топологія; б — характеристика $I_c=f(U_{зб}, \varphi)$; в — схема ЕП.

У схемі ЕП на МНОН-транзисторі вихідний струм $I_{вих}$, який відображає лог 1, з'являється при таких умовах: $X_i = Y_i = 1$, заряд під затвором транзистора $VT1$ відсутній. Після програмування новий стан МНОН-транзистора зберігається місяцями чи роками. Перед новим записуванням стара інформація стирається записуванням нуля в ЕП. Число перезаписувань інформації знаходиться в межах 10^4 – 10^6 раз.

У режимі зчитування на затвори МНОН-транзисторів подають напругу $U_{зч}$, значення якої лежить між двома граничними рівнями. Якщо в ЕП записана лог. 1, то транзистор відкривається; якщо лог. 0 — транзистор буде закритим.

Отже, залежно від стану транзистора при подачі на його затвор напруги зчитування $U_{зч}$ (по координаті X_i) у розрядній шині Y_j струм протікає або ні (рис. 7.22, в). Підсилювач зчитування трансформує стан розрядної шини у вихідну напругу високого (струм протікає) або низького (струм відсутній) рівнів.

Якщо транзистор має заряд у діелектрику, тобто знаходиться у стані лог 0, то для переходу його у стан "1" накопичений заряд витискують з-під затвора негативним імпульсом напруги 30–40 В, який подається на затвор відносно підкладки. Такий процес називають режимом стирання.

Структуру мікросхеми КР1601РРЗ ємністю 2 К x 8 біт на p -канальних МНОН-транзисторах показано на рис. 7.23.

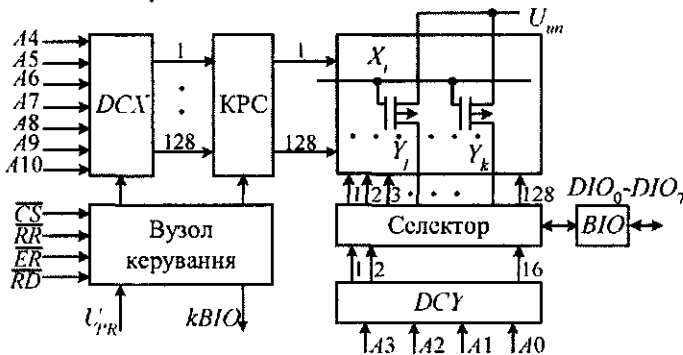


Рис. 7.23. Структура мікросхеми РР3П-ЕС КР1601РР3

Мікросхема КР1601РРЗ вміщує:

- матрицю ЕП, в якій є 128 рядків і 128 стовпців, на перетині яких розташовуються 16384 транзисторів;
- дешифратор адреси рядків DCX , на входи якого надходять сім старших розрядів адреси A_4-A_{10} ; він має 128 виходів, кожен з яких керує вибіркою одного з рядків матриці;
- дешифратор адреси стовпців DCY , на входи якого надходять чотири молодших розряди адресного коду A_0-A_3 . Виходи DCY керують селектором. На входи селектора надходять сигнали, зчитані з ЕП вибраного рядка. Селектор вибирає з 128-розрядного слова (16 байт) одне восьмирозрядне (байт) і передає його через буфер введення-виведення BIO на зовнішню шину;
- вузол керування, на входи якого надходять інверсні керуючі сигнали: \overline{CS} — вибір мікросхеми; \overline{PR} — дозвіл на програмування (запис); \overline{ER} — сигнал дозволу стирання інформації. На вузол подається також напруга програмування U_{PR} . Вузол керування забезпечує режими збереження, зчитування, стирання і записування. Стани керуючих сигналів, які забезпечують зазначені режими роботи, наведені в табл. 7.3.

Таблиця 7.3

Режими роботи	Сигнали стану				$U_{PR}, \text{В}$	$t_A, \text{мкс}$
	\overline{CS}	\overline{RD}	\overline{PR}	\overline{ER}		
Стирання	0	1	1	0	-36	200
Записування	0	1	0	1	-36	20
Зчитування	0	0	1	1	-12	0,4
Зберігання	1	x	x	x	x	

Стирання починається з моменту подачі імпульса \overline{ER} протягом 100–200 мс. Після закінчення стирання всі ЕП матриці переходять в стан лог. 0; допускається також порядкове стирання.

В режимі записування на виводи мікросхеми подають значення розрядів записуваного байта даних, адресний код, керуючі сигнали і потім імпульсний сигнал програмування $\overline{PR} = 0$ на час 20 мс. Для програмування в автоматичному режимі усієї мікросхеми з числом адрес, що дорівнює 2048, потрібно 41 с. У режимі зчитування на вивід U_{PR} комутують напругу живлення мінус 12 В для зниження використовуваної потужності, потім подають код адреси і сигналу керування відповідно до табл. 7.3. Через 0,4 мкс на інформаційних виходах з'являється зчитуване слово.

Режим зберігання визначається сигналом \overline{CS} , який забороняє звертання до мікросхеми незалежно від стану сигналів на других входах.

В режимі зберігання при відімкненні джерела живлення заряд під затвором може зберігатися декілька років. Достоїнством мікросхем РПЗП-ЕС є можливість їхнього перепрограмування без вилучення з пристрою, а також значне число циклів перепрограмування, яке досягає 10 тис.

Ця властивість разом з енергонезалежністю дозволяє широко застосовувати їх у комп'ютерах як вбудованих ПЗП зі стиранням інформації.

7.4.5. Мікросхеми РПЗП на ЛІЗМОН-транзисторах

Запам'ятовуючий елемент типу ЛІЗМОН — це МОН-транзистор з індукованим p - або n -каналом. Металевий затвор цього транзистора розміщений в товщі діелектрика (звичайно діоксиду кремнію) і не має металевого виводу. Цей затвор називається плаваючим (ПЗ). Від кристала ПЗ відокремлюється діелектриком товщиною 0,1 мкм (рис. 7.24, а).

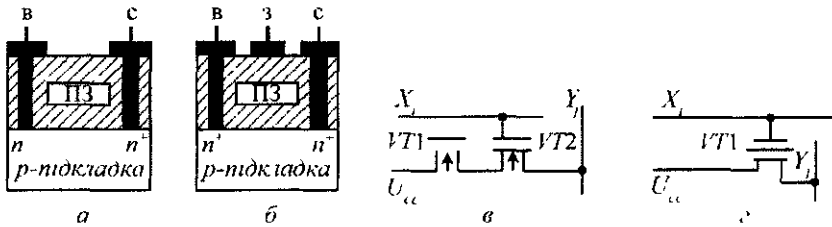


Рис. 7.24. Елемент пам'яті на ЛІЗМОН-транзисторі:
 а, б — топологія транзистора з одним і двома затворами;
 в — схема з одним затвором; г — схема з двома затворами.

При наявності заряду на ПЗ утворюється провідний канал і транзистор відкритий — це відповідає запису лог. 1; при відсутності заряду транзистор закритий (записаний лог. 0).

Структурна схема ЛІЗМОН-транзистора з додатковим затвором вибірки (двозатворний транзистор) показана на рис. 7.24, б. В режимі програмування на витік і стік транзистора відносно підкладки подається позитивний імпульс напруги амплітудою 25 В. На обернено зміщених n - p -переходах витік-підкладка і стік-підкладка виникає процес лавинної інжекції заряду (звідси походить позначення транзистора ЛІЗМОН). Частина електронів попадає на ПЗ; в результаті накладення на ПЗ негативного заряду порогова напруга на передаточній характеристиці зміщується в область більш високого рівня (зсув вправо), що відповідає запису лог. 0. Відсутність зарядів електронів на ПЗ відповідає запису лог. 1.

Стирання записаної інформації полягає в витісненні заряду з ПЗ. Цю операцію в транзисторах ЛІЗМОН здійснюють двома методами. у групі РПЗП-ЕС імпульсом напруги, який подається на затвор; у групі РПЗП-УФ за допомогою ультрафіолетового (УФ) опромінення крізь прозоре скло у кришці корпусу. У першому випадку накопичені на ПЗ електрони витісняються у підкладку електричним полем і відновлюється стан "1". У другому випадку електрони розсмоктовуються з ПЗ у підложку в результаті посилення теплового руху за рахунок одержання від джерела УФ випромінення енергії.

Режим зчитування здійснюється так, як і в мікросхемі РПЗП на МОН-транзисторах. У режимі зберігання забезпечується відсутність напруги на електродах ЕП для того, щоб виключити розсмоктування заряду на ПЗ у діелектричному середовищі. Наприклад, у мікросхемі К573РФ6 гарантований строк збереження інформації без живлення становить п'ять років, а у КМ1609РР2 — до 10 років.

Схема включення в ЕП транзистора $VT1$ з одним затвором показана на рис. 7.24, в, а з двома затворами — на рис. 7.24, г.

В сучасних комп'ютерах використовують

- пам'ять типу РПЗП-УФ з ємністю до 8 Мбіт і часом доступу 45 нс (фірма *Atmel*)
- пам'ять типу РПЗП-УФ з ємністю до 256 Кбіт, часом доступу 90 нс, допустимим числом циклів перезаписування 105 і тривалістю зберігання даних більше 10 років, ці ЗП використовують одне джерело напруги 5 В і споживають струм 5 мА в активному режимі та 0,1 мА при відсутності звернень (фірма *SGS-Thomson*)

В енергонезалежній постійній пам'яті широко використовують режим імпульсного живлення напруга подається тільки на ту мікросхему пам'яті, до якої є звернення. Це в багато разів зменшує енергоспоживання, покращує тепловий режим ІМС і підвищує їхню надійність.

7.5. Флеш-пам'ять

7.5.1. Загальна характеристика флеш-пам'яті

Флеш-пам'ять (*Flash Memory*) використовує ЕП на транзисторах ЛІЗМОН з електричним стиранням і записуванням інформації. Вона відноситься до постійної пам'яті типу *EEPROM*, але ряд архітектурних і функціональних особливостей дозволили виділити флеш-пам'ять в окремий клас. Флеш-пам'ять використовує поряд з традиційними адресними і керуючими сигналами спеціальні команди. Інформація у мікросхемах флеш-пам'яті записується і зберігається в блоках визначеного розміру, іноді — призначення. При цьому стирання інформації здійснюється або для всієї пам'яті разом, або для великих блоків, це спрощує схеми ЕП. Флеш-пам'ять переважає *EEPROM* у тому, що не вимагає спеціальної апаратури для записування чи стирання даних.

Розрізняють такі види флеш-пам'яті:

- файлова флеш-пам'ять (*Flash File*) — масив ЕП розділений на блоки однакового розміру (симетрична архітектура),
- флеш-пам'ять з несиметричною архітектурою (*Boot Block*) — масив ЕП розділений на блоки різного розміру, один з блоків має апаратні засоби для захисту інформації в ньому,
- флеш-пам'ять з можливістю стирання тільки всього масиву ЕП (*Bulk Erase*),
- флеш-пам'ять з можливістю записування інформації за різних напруг програмування (*Start Voltage*),
- пам'ять з використанням нових ЕП з чотирма станами, які зберігають по два біти (*Strata Flash*)

Файлова флеш-пам'ять орієнтована на заміну жорстких магнітних дисків. Такі ЗП в сотні разів зменшують споживану потужність, збільшують механічну міцність та надійність, зменшують їхні розміри і масу та на декілька порядків підвищують швидкість при читанні даних. Мікросхеми, які замінюють магнітні диски, мають ідентичні блоки та розвинені засоби обміну інформацією. Мікросхеми файлової флеш-пам'яті фірми *Intel* мають інформаційну ємність 4–32 Мбіт, час доступу — 70–150 нс.

Мікросхеми *Boot Block* використовують одnobайтову або перемикальну одно- чи

двобайтову організацію і складаються з декількох блоків різного розміру. Один з блоків має додаткові апаратні засоби захисту від зміни даних; він призначений для зберігання дуже важливої інформації, яка не змінюється при модифікації даних в інших блоках. Мікросхеми *Boot Block* призначені для зберігання компонентів системного програмного забезпечення. Привілейований блок містить програму-завантажувач, яка записує з диска необхідні дані для ініціалізації пристроїв комп'ютера.

Мікросхема *Boot Block* типу *28F00BX1N*, яка часто застосовується для зберігання програм базової системи введення-виведення (*BIOS*) в ПЕОМ, має час доступу 75–150 нс, гарантується 10^5 циклів стирання-програмування.

Мікросхема *28F00BX* містить:

- основний блок об'ємом 112 Кбайт;
- два блоки параметрів ємністю по 4 Кбайт кожний;
- блок-завантажувач об'ємом 4 Кбайт, стирання і програмування якого можливе тільки за особливих умов.

Основний блок і блоки параметрів захисту — рівноправні. Виділення невеликих блоків параметрів дозволяє зберігати в них інформацію, яка часто змінюється.

Мікросхеми *Bulk Erase* мають однобайтову організацію ємністю 32–256 Кбайт, час доступу 65–200 нс і являють собою єдиний масив, який стирається відразу.

7.5.2. Мікросхема файлової флеш-пам'яті *28F008SA*

Мікросхема файлової флеш-пам'яті *28F008SA* розроблена фірмою *Intel* в 1989 р. на основі нової технології *ETOX III (EPROM Tunnel Oxide)*. До особливостей технології *ETOX III* відносяться:

- роздільна здатність — 0,8 мкм;
- використання однострижкових ЕП на ЛІЗМОН-структурах з розмірами 2,5 x 2,9 мкм та плаваючим полікремнієвим затвором.

В основі програмування флеш-пам'яті лежить механізм каналної інжекції гарячих електронів. Під час програмування керуючий затвор ЕП підключається до зовнішньої напруги $U_{PR} = 12$ В. При стиранні електрони одночасно витісняються з усіх вибраних ЕП оберненим тунелюванням. Структура мікросхеми *28F008SA* показана на рис. 7.25. Вона є типовою для флеш-пам'яті.

Опишемо основні компоненти мікросхеми та їхнє функціональне призначення. Мікросхема *28F008SA* містить:

- матрицю ЕП ємністю 8 Мбіт з організацією 1 М x 8 біт, яка розбита на 16 блоків по 64 Кбайт в кожному. Блок складається з 1024 рядків і 512 стовпців;
- буфер адреси *BA*, регістр *RGA* та лічильник *CTA* адрес. Буфер *BA* з'єднаний з адресними входами *A19–A0*. При зверненні до пам'яті адреса фіксується в регістрі, з якого вона подається на дешифратори рядків *DCX* і стовпців *DCY*;
- буфери вхідних *BID* та вихідних *BOD* даних. Обмін даними здійснюється за двонаправленими входами-виходами *D7–D0*. Вхідна інформація записується у внутрішній регістр даних *RGD*. Вихідні дані з матриці ЕП через мультиплексор *MUX* подаються на вихід через буфер *BOD*. Обидва буфери мають по три стани;
- регістр-ідентифікатор *RID*; призначений для видачі коду фірми-розробника та коду мікросхеми;

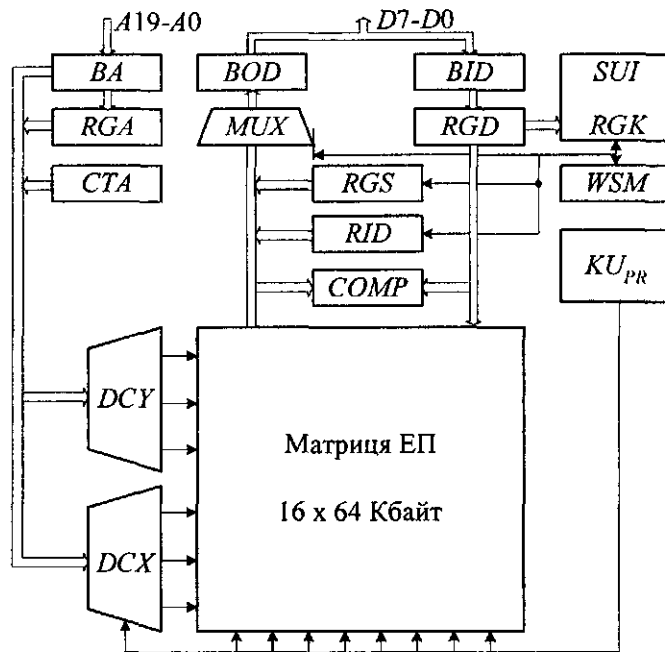


Рис. 7.25. Структура NBIC 28F008SA

- командний інтерфейс (блок команд) користувача *CUI* (*Command User Interface*), у складі якого є регістр команд *RGK*. Блок *CUI* виконує роль інтерфейса між мікропроцесором і внутрішніми операціями в мікросхемі;
- автомат записування байта і стирання блоку *WSM* (*Write State Machine*). Він сприймає команди від блоку *SUI* і автоматично виконує алгоритм записування байта або стирання блоку з наступною верифікацією (перевіркою) результату. Записування даних у пам'ять виконується побайтно з типовим часом 9 мкс. Типовий струм споживання при цьому від джерела програмування $I_{PR} \leq 30$ мА. Напруга записування байта і стирання блоку $U_{PR} = 12$ В;
- регістр стану *RGS* (*Status Register*); зберігає інформацію про стан автомата *WSM* і служить для контролю корекції операцій записування або стирання;
- компаратор *COMP*; використовується при контролі правильності запису байта даних;
- комутатор напруг KU_{PR} .

Для керування роботою мікросхеми служать входи, сигнали на яких активні в стані низького рівня (рис. 7.26).

До вхідних сигналів відносяться:

- \overline{CE} — вибір мікросхеми; \overline{WE} — дозвіл записування, \overline{OE} — дозвіл видачі даних;
- \overline{PWD} — керування енергоспоживанням: $\overline{PWD} = 1$ — нормальна робота; $\overline{PWD} = 0$ — режим мікроспоживання (типове значення струму $I = 0,2$ мкА, потужність споживання — 1 мВт) та блокування операцій записування і стирання;

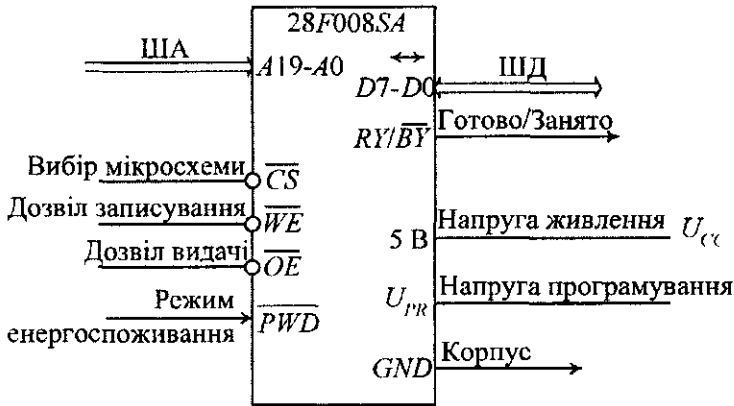


Рис. 7.26. Умовне графічне позначення мікросхеми 28F008SA

- U_{RR} — вхід для подання напруги стирання блоку чи записування байта. Якщо $U_{RR} < U_{PR}$, то змінити дані неможливо;
- U_{CC} — напруга живлення плюс 5 В;
- GND — корпус мікросхеми.

Вихід RY/\overline{BY} (Ready/Busy) показує стан автомата WSM . Якщо $RY/\overline{BY} = 1$, то автомат зайнятий, а якщо $RY/\overline{BY} = 0$, то готовий до приймання нової команди.

Операції в пам'яті відбуваються після записування відповідної команди:

- "Читання масиву" (код FFH);
- "Ідентифікатор" (код $90H$). Спочатку зчитується код фірми-виробника $89H$, потім код мікросхеми $A2H$. Ці коди дозволяють автоматично визначити алгоритм роботи з пам'яттю;
- "Читання регістра стану" (код $70H$);
- "Скидання регістра стану" RGS (код $50H$);
- "Установлення стирання" (код $20H$); "Підтвердження стирання" (код $D0H$). Команди ідуть одна за одною;
- "Призупинити стирання" (код $B0H$); "Відновити стирання" (код $D0H$);
- "Установити записування байта" (код $40H$); "Записування".

Значення розрядів регістра станів:

- $S7 = 1$ — автомат WSM готовий;
- $S6 = 1$ — стирання призупинено;
- $S5 = 1$ — помилка при стиранні блоку;
- $S4 = 1$ — помилка при записуванні байта;
- $S3 = 1$ — низький рівень напруги живлення;
- $S2 - S0$ — резерв.

7.5.3. Основні напрямки розвитку флеш-пам'яті

Для поліпшення техніко-економічних характеристик флеш-пам'яті використовують різні засоби і способи.

1. Переривання тривалого процесу записування при необхідності читання пам'яті.

2. Створення внутрішньої черги команд, яка дозволяє організувати конвеєрний процес роботи.
3. Програмування довжини слів.
4. Введення пониженого енергоспоживання в режимі спокою, коли струм зменшується до 2 мкА.
5. Пристосування для роботи з різними напругами живлення: 5; 3,3 і навіть 2,6 В.
6. Введення в структуру пам'яті буферів, які забезпечують одночасне записування і виведення даних.
7. Вдосконалення засобів захисту від випадкового стирання чи несанкціонованого доступу.

Для прикладу наведемо параметри файлової мікросхеми флеш-пам'яті 28F082SA (вибір фірми *Intel*, 1997 р.):

- організація 2 М x 16 або 4 М x 8 біт (ємність 32 Мбіт) за вибором користувача;
- напруга живлення — 3,3 або 5 В, напруга програмування — 12 В, до 10^6 циклів стирання на блок;
- число незалежних блоків: 64 x 64 Кбайт або 64 x 32 Кслів;
- корпус — типу *TSOP* розмірами 1,2 x 14 x 20 мм з 56 виводами;
- технологія — з роздільною здатністю 0,6 мкм;
- час доступу при читанні — 70 або 150 нс при напругах 5 і 3 В відповідно;
- час записування слова або байта — не більше 9 мкс;
- час записування блоку — не більше 2,1 с для байтового режиму і не більше 1 с — для словникового режиму;
- час стирання блоку — не більше 10 с і стирання кристала — не більше 25,6 с.

7.5.4. Пам'ять типу *Strata Flash*

Флеш-пам'ять типу *Strata Flash* розроблена фірмою *Intel* в 1997 р. У ній вперше в одному ЕП зберігаються два біти. Це забезпечується створенням у плаваючому затворі транзистора ЛІЗМОН чотирьох значень зарядів. Записування двох бітів здійснюється практично на тих же розмірах ЕП, які використовувалися для зберігання одного біта. Таким чином, це дозволило від ємності 32 Мбіт, наприклад, одразу перейти до ємності 64 Мбіт.

Запам'ятовуючі ЕП програмуються введенням у плаваючий затвор однієї з чотирьох кількостей зарядів, кожна з яких відповідає парі двійкових цифр: 11, 10, 01 і 00. Залежно від заряду ЕП має одну з чотирьох порогових напруг. При зчитуванні інформації струм елемента пам'яті має чотири значення, на їхній основі вихідні схеми формують дворозрядний код.

7.6. Статичні запам'ятовуючі пристрої

7.6.1. Загальна характеристика статичних запам'ятовуючих пристроїв

У статичних ЗП функцію запам'ятовування біта інформації виконують тригери. Вони реалізуються за будь-якою схемотехнікою — ТТЛШ, I^2I , ЕЗП, n -МОН, КМОН та іншими. Найбільш інтенсивно розвиваються ОЗП на КМОН-структурах, які при зменшенні роздільної здатності до 0,2 мкм набувають високої швидкодії. При цьому вони зберігають свої традиційні переваги — велику інформаційну ємність та дуже мале енергоспоживання — до долей мікрвольта на один ЕП.

Статичні ОЗП (SRAM) звичайно мають структуру 2DM, а при невеликій інформаційній ємності будуються за структурою 2D. Вони широко використовуються в кеш-пам'яті, яка повинна мати максимально можливу швидкодію.

Для побудови ЕП статичних ЗП широко використовують RS-тригери за схемотехнікою КМОН. Типова схема такого RS-тригера містить:

- власне тригер на транзисторах $VT3$ і $VT5$ (n -тип) та на навантажувальних транзисторах $VT2$ і $VT4$ (p -тип);
- ключі вибірки на транзисторах $VT1$ і $VT6$ (рис. 7.27).

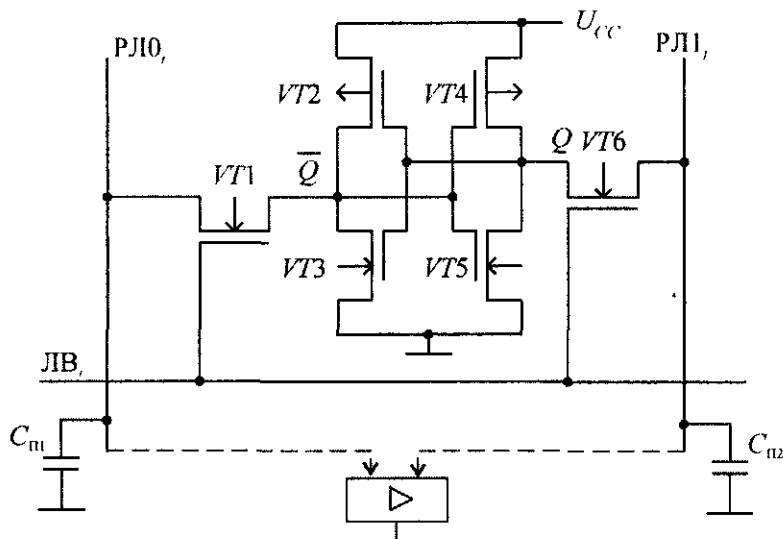


Рис. 7.27. Схема RS-тригера на КМОН структурах

З прямим Q та інверсним \bar{Q} виводами тригера через ключі вибірки пов'язані розрядні лінії записування-зчитування $P\bar{L}0$, та $P\bar{L}1$. В режимі зберігання транзистори $VT1$ і $VT6$ закриті (лог. 1). При виборі даного тригера (лінія вибору $ЛВ$, = 1) для записування одиниці встановлюють $P\bar{L}0$, = 0, $P\bar{L}1$, = 1, а при записуванні нуля — навпаки.

Під час зчитування даних сигнал $ЛВ$, = 1, і двонаправлені ключі $VT1$ і $VT6$ відкриваються. При цьому напруга паразитної ємності $C_{п1}$ або $C_{п2}$ збільшується на значення ΔU в плечі закритого транзистора $VT5$ або $VT3$. Наприклад, якщо тригер зна-

ходиться в стані лог. 1, то заряджається паразитна ємність $C_{п2}$. Якщо тригер знаходиться в стані лог. 0, то заряджається паразитна ємність $C_{п1}$. При цьому стан тригера не змінюється. Значення напруг з паразитних ємностей подаються на диференціальний підсилювач, підключення якого показано на рис. 7.27 штриховою лінією. Підсилювач формує на виході відповідне значення лог. 1 або лог. 0.

Схема вихідного каскада з трьома станами, який широко використовується в статичній пам'яті на КМОН структурах, показана на рис. 7.28, а.

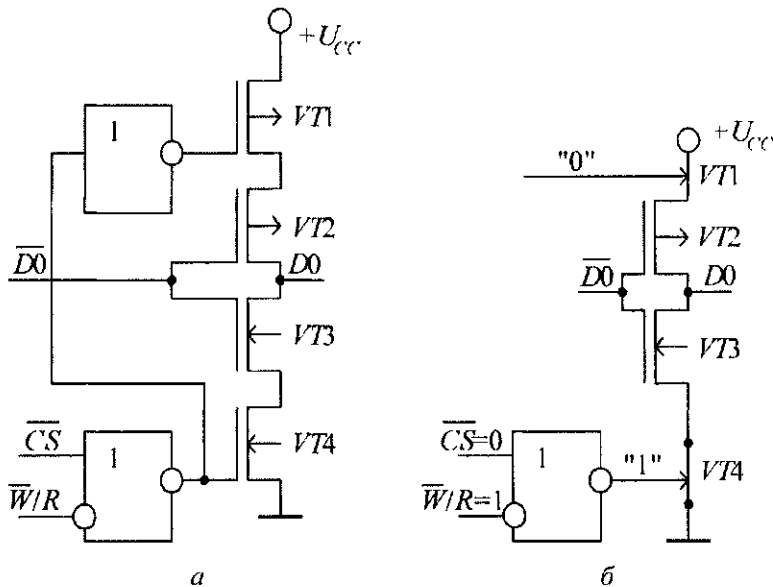


Рис. 7.28. Вихідний каскад з трьома станами на КМОН-структурах:

а — схема; б — еквівалентна схема для відкритого стану

Транзистори $VT1$ і $VT2$ побудовані за p -МОН технологією, а $VT3$, $VT4$ — за n -МОН технологією. Сигнал передається на вихід інверсним кодом тільки за умови $\overline{CS} = 0$, $\overline{W/R} = 1$. У цьому випадку транзистори $VT2$ і $VT3$ створюють інвертор, а транзистори $VT1$ і $VT4$ відкриті й насичені. Для інших комбінацій керуючих сигналів \overline{CS} , $\overline{W/R}$ схема знаходиться в третьому стані.

7.6.2. Модуль статичної пам'яті

Умовне графічне зображення мікросхем SRAM серій K132PY9 та K541PY2 показано на рис. 7.29, а.

Побудова модуля пам'яті з організацією 1 К x 16 біт на основі чотирьох мікросхем K132PY9 показана на рис. 7.30.

Для побудови даного модуля пам'яті необхідно:

- підключити десятирозрядну ША (10) до адресних входів всіх мікросхем;
- на входи всіх мікросхем подавати сумісні керуючі сигнали \overline{CS} , $\overline{W/R}$;
- до двонаправлених входів-виходів підключити відповідні розряди з ШД (16).

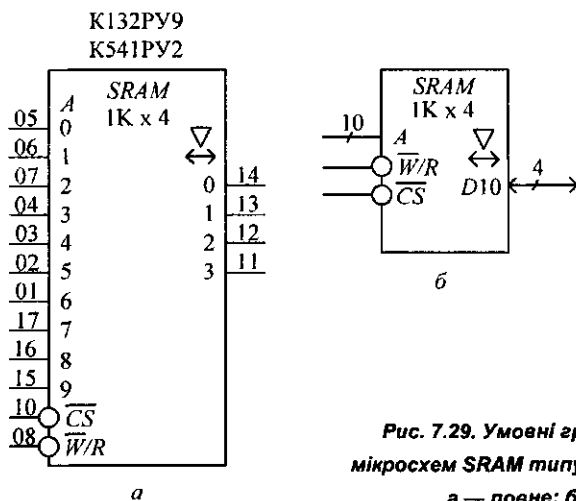


Рис. 7.29. Умовні графічні зображення мікросхем SRAM типу K132PY9 та K541PY2: а — повне; б — спрощене

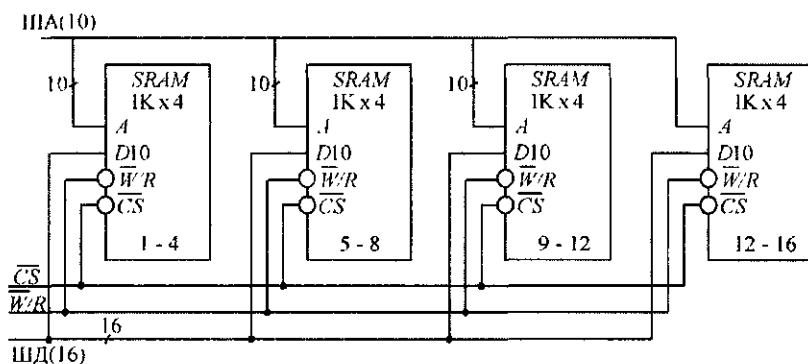


Рис. 7.30. Побудова модуля пам'яті 1К x 16 біт

Розглянутий спосіб побудови блоку пам'яті з більшою розрядністю називають розширенням по горизонталі.

Основні параметри ряду вітчизняних серій мікросхем SRAM наведені в табл. 7.4 (значення параметрів усереднені).

Таблиця 7.4

Серія	Ємність, біт	t_{cy} , нс	U_{cc} , В	P_{cc} , Вт	Схемотехнологія
K500	64 x 4 4 К x 1	30	-5,2	0,8	ЕЗЛ
K1500	16 К x 1 1 К x 4	20	-4,5	0,8	ЕЗЛ
K132	4 К x 4	70	5	0,6	І ² Л-ТТЛ
K541	1 К x 4	130	5	0,4	І ² Л-ТТЛ
K537	64 К x 1	200	5	0,02	КМОП
K1809	1 К x 16	800	5	0,6	n-МОП
K6500	1 К x 41	4	4; -2,4	1,6	GaAs

Американська електронна промисловість випускає мікросхеми типу *SRAM* з організацією 8 К x 8, 16 К x 8, 32 К x 8, 64 К x 8 і 128 К x 8 біт та часом доступу 8–20 нс.

7.7. Динамічна пам'ять

7.7.1. Принцип побудови динамічного запам'ятовуючого елемента

У динамічній пам'яті типу *DRAM* інформація зберігається у вигляді зарядів на дуже малій ємності $C_3=0,01...0,05$ пФ, яка створена між стоком і підкладкою МОН-транзистора (рис. 7.31).

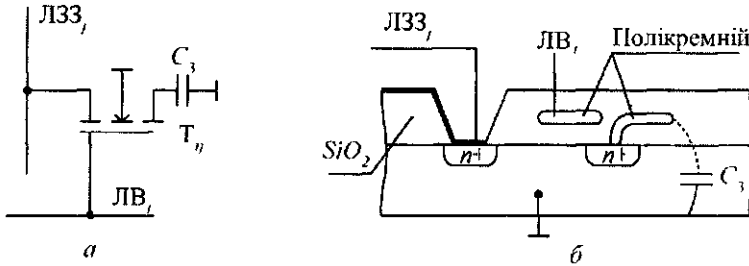


Рис. 7.31. Динамічний однотранзисторний ЕП: а — схема; б — топологія

Стік транзистора не має зовнішнього виводу. Для записування інформації на лінію вибірки ЛВ_{*i*} подається високий рівень напруги, яка відкриває транзистор T_{ij} . Створюється провідний канал, і рівень напруги на розрядній лінії записування-зчитування ЛЗЗ_{*j*} визначає стан конденсатора C_3 : заряджений при високому рівні (стан "1") та розряджений при низькому (стан "0")

Фрагмент ЗП (рис. 7.32) показує два динамічних ЕП в одному стовпці, підсилювач зчитування ПЗЧ, а також ключі K_1 і K_0 відповідно для записування одиниці і нуля.

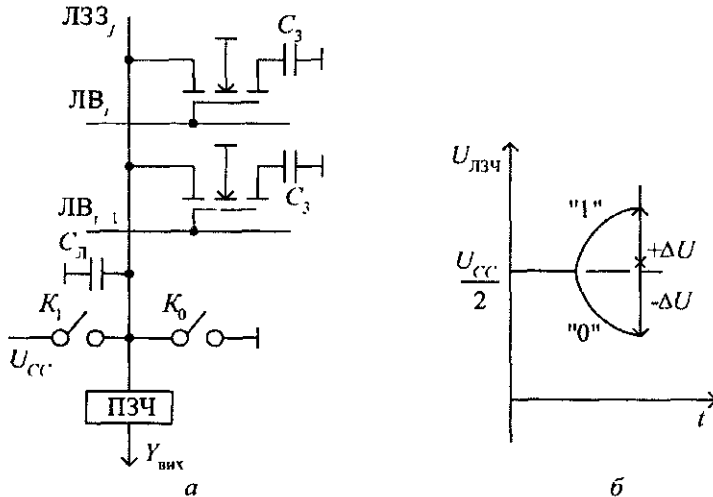


Рис. 7.32. Фрагмент динамічного ЗП: а — схема; б — часові діаграми зчитування

Лінія ЛЗЗ, довга, до неї підключено багато транзисторів (визначаються кількістю рядків), тому вона має велику ємність C_n , яка багаторазово перевищує C_3 запам'ятовуючого елемента. Перед зчитуванням лінія ЛЗЗ, (власна ємність C_n) попередньо заряджається до рівня $U_{CC}/2$ (можливі й інші значення в ряді мікросхем пам'яті). При зчитуванні нуля (сигнал ЛВ₁ = 1) лінія ЛЗЗ, підключається до ємності C_3 з нульовим потенціалом. Тому частина заряду ємності C_n перетікає в ємність C_3 і напруга на них зрівнюється. Потенціал лінії ЛЗЗ, понижується на значення ΔU . Це є сигнал лог. 0, який поступає на ПЗЧ. При зчитуванні одиниці, навпаки, напруга на ємності C_3 спочатку дорівнює значенню U_{CC} . При підключенні ємності C_3 до лінії ЛЗЗ, частина її заряду передається на ємність C_n і напруга на лінії ЛЗЗ, збільшується на ΔU . Це сигнал лог. 1, який поступає на ПЗЧ.

До вибірки ЕП ємність C_n мала заряд $Q = C_n U_{CC}/2$, а після — цей самий заряд має спільна ємність $C_3 + C_n$. Тому справедлива рівність: $Q = (C_n + C_3)(U_{CC}/2 - \Delta U)$. Порівнюючи вирази для одного і того ж заряду Q , запишемо:

$$C_n U_{CC}/2 = (C_n + C_3)(U_{CC}/2 - \Delta U),$$

звідки для ΔU одержуємо

$$\Delta U = U_{CC} C_3 / 2 C_n.$$

Внаслідок того, що $C_n \gg C_3$, сигнал ΔU виявляється слабким. Крім цього, зчитування є руйнівним — ємність C_3 суттєво змінює свій потенціал.

Ці недоліки можна усунути збільшенням ємності C_3 (без зміни площі ЕП) або зменшенням ємності C_n .

Зменшення ємності C_n досягається при розрізанні ЛЗЗ на половини і включенні між ними диференційного підсилювача-регенератора ПРЕГ. Такий спосіб удвічі зменшує ємність C_n і відповідно сигнал ΔU збільшується в два рази (рис. 7.33).

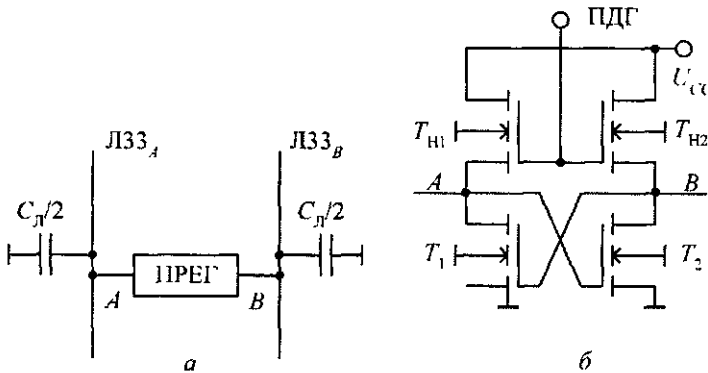


Рис. 7.33. Спосіб зменшення ємності C_n :

а — розрізання розрядної лінії на половини; б — схема ПРЕГ

Схема ПРЕГ будується на основі тригерів з використанням додаткового сигналу підготовки ПДГ для керування навантажувальними транзисторами T_{n1} і T_{n2} (рис. 7.33, б). Спочатку сигнал ПДГ = 0 і транзистори T_{n1} і T_{n2} закриті. Входи-виходи ПРЕГ підключені відповідно до ЛЗЗ_А і ЛЗЗ_В. При зчитуванні сигнал ПДГ = 1 і транзистори T_{n1} і T_{n2} відкриваються. Потенціал ЛЗЗ з вибраним ЕП збільшується або зменшується на ΔU , а друга лінія зберігає постійну напругу $U_{CC}/2$. Стан тригера визначається тим входом-виходом, на якому змінюється напруга.

При збільшенні напруги на ΔU даний вхід-вихід набуває значення лог. 1 (протилежний — лог. 0). При зменшенні напруги на ΔU даний вхід-вихід набуває значення лог. 0 (протилежний — лог. 1).

Після переключення тригер зберігає на виводах *A* і *B* значення записаних даних, за допомогою яких він відновлює на ємності C_3 повне значення зчитаного сигналу. Тим самим автоматично здійснюється регенерація даних в ЕП. Стан тригера вказує також вихідні сигнали зчитаної інформації.

7.7.2. Схема динамічного запам'ятовуючого пристрою

Типова схема динамічного ЗП з однобітною організацією $N \times 1$ містить (рис. 7.34):

- матрицю ЕП (розкритий частково один стовпець);
- буфери адрес рядків і стовпців;
- дешифратори адрес рядків DCX і стовпців DCY ;
- формувачі $\Phi C1$ і $\Phi C2$, які виробляють тактові сигнали $\Phi 1 - \Phi 4$;
- підсилювач-регенератор ПРЕГ;
- інвертори та ключові схеми $K1 - K4$ і $K8$.

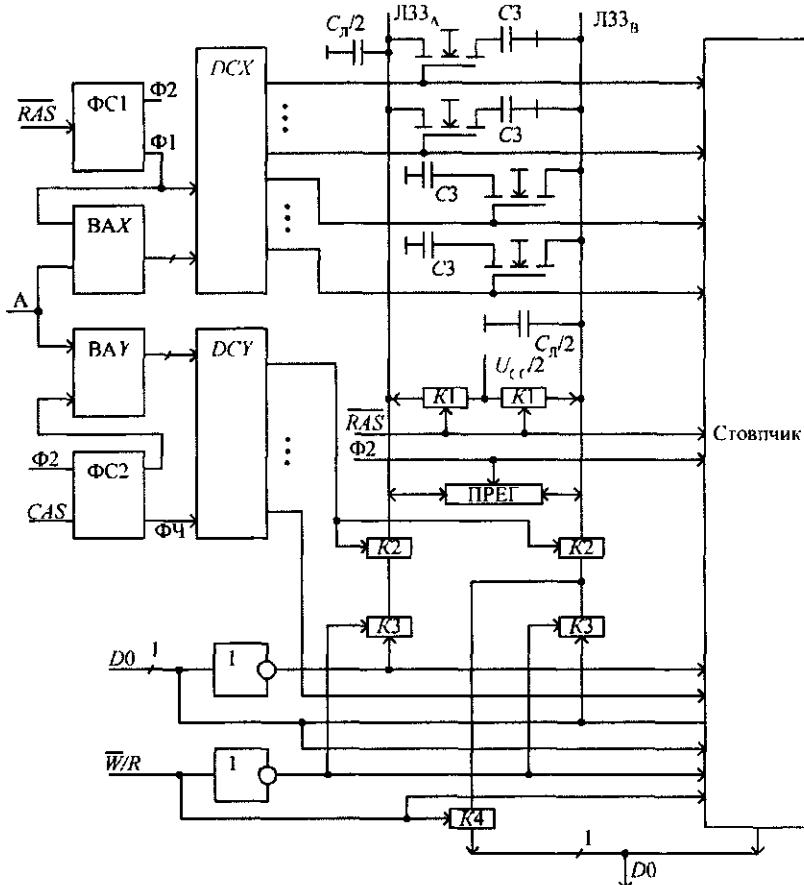


Рис. 7.34. Схема динамічного ЗП

За допомогою схеми ПРЕГ лінії запису-зчитування розрізані на дві рівні частини $L33_A$ та $L33_B$, при цьому ємність розрядних ліній зменшується до значень $C_n/2$

У початковому стані високий рівень сигналу $\overline{RAS} = 1$ замикає ключі $K1$, які подають напругу $U_{cc}/2$ для заряду $L33_A$ та $L33_B$. При звертанні до пам'яті одночасно з сигналом $\overline{RAS} = 0$ подається старша половина адресного коду адреси рядків A_x (підготовчий етап)

Ключі $K1$ розмикаються і $L33$ ізолюються від напруги $U_{cc}/2$. При цьому в усіх ЕП вибраного рядка відбуваються процеси зарядження або розрядження запам'ятовуючого конденсатора C , і створюється дисбаланс напруг на входах схеми ПРЕГ

Другий тактуючий імпульс $\Phi 2$ знімає сигнал ПДГ з підсилювачів-регенераторів і вони переключаються. При цьому на їхніх входах-виходах формуються повні рівні сигнали, які відновлюють стан ЕП вибраного рядка

На виконавчому етапі операції записування або зчитування необхідна наявність сигналу $\overline{CAS} = 0$ — приймання молодшої частини адресного коду A_y (адреси стовпців). Формувач $\Phi C2$ виробляє другу пару тактуючих імпульсів $\Phi 3$ і $\Phi 4$. Сигнал $\Phi 3$ дозволяє завантаження в буфер BAY адресу стовпців, сигнал $\Phi 4$ активізує роботу дешифратора DCY , в результаті чого відкриваються ключі $K2$ вибраних стовпців

Якщо $\overline{W} / R = 1$, то виконується операція зчитування біта інформації з правого плеча схеми ПРЕГ і передача його значення через ключі $K2$ і $K4$ на вихідну лінію даних DO . При цьому дані, зчитані з $L33_A$, передаються інверсним кодом

При $\overline{W} / R = 0$ виконується операція записування біта інформації з вхідної лінії DI через ключі $K3$. При цьому дані, які записуються на $L33_A$, інвертуються

Операція регенерації здійснюється в середині ЗП і для нього достатньо тільки подання сигналу \overline{RAS} (разом з адресами рядків, які регенеруються) і вироблення тактуючих сигналів $\Phi 1$ і $\Phi 2$

7.7.3. Динамічні запам'ятовуючі пристрої підвищеної швидкодії

Динамічні ЗП підвищеної швидкодії функціонують на основі припущення, що адреси поточного і наступного звертання до пам'яті з великою ймовірністю розміщені рядом (сукупність адрес). Це дозволяє зменшити тривалість виконавчого та підготовчого етапів обміну даними або виключити один з них

Розглянемо коротко основні структури в розвитку *DRAM*. Вважається, що пам'ять умовно розбивається на сторінки, адреси яких подаються старшою частиною A_x адресного коду (записується по стробу \overline{RAS}). Слова в сторінці адресуються молодшою частиною A_y адресного коду (записується по стробу \overline{CAS}). Наприклад, для адреси довжиною $k = 20$ маємо 1024 сторінки і 1024 слова в сторінці

FPM

Структура *FPM* (*Fast Page Mode*) забезпечує швидкий пакетний доступ до рядка (сторінки) і слова в ньому. В даній пам'яті адреса рядка A_x подається тільки один раз по стробу \overline{RAS} . В подальшому змінюється тільки адреса стовпця по стробу \overline{CAS} (рис 7.35)

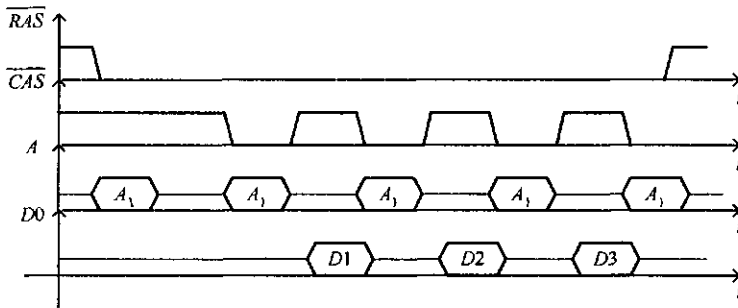


Рис. 7.35. Часові діаграми роботи пам'яті типу FPM

Таким чином, пакетний обмін здійснюється при одному підготовчому етапі і багаторазовому виконавчому. Для FPM характерну пропорційність інтервалів часу першого і наступних звертань записують як 5–3–3...

EDORAM

Структура з розширеним виведенням даних EDORAM (Extended Data Out RAM) відрізняється від FPM модифікацією процесу зчитування даних. У цій пам'яті після закінчення стробу CAS схеми ПРЕГ не обнуляються. Вони створюють ніби статичний регістр, що забезпечує наступне швидке зчитування в межах даного рядка. Як і раніше використовується тільки один сигнал CAS, але його тривалість скорочується. Це збільшує швидкодію ЗП. Для пам'яті EDORAM характерну пропорційність інтервалів часу між першим і наступними звертаннями подають як 5–2–2... Розроблені EDORAM забезпечують роботу до 50 МГц.

BEDORAM

В структурі з пакетним розширеним доступом BEDORAM (Burs EDORAM) є додатковий лічильник адрес стовпців. При звертанні до пакета (групи слів) адреса стовпця формується тільки на початку пакетного циклу. У подальшому адреси стовпців створюються інкрементуванням змісту лічильника (рис. 7.36).

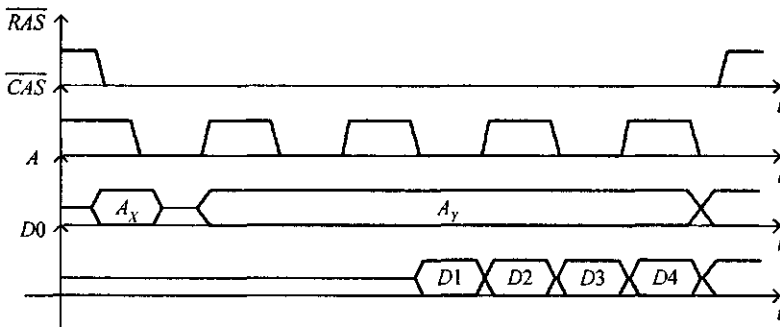


Рис. 7.36. Часові діаграми зчитування пам'яті BEDORAM

Для BEDORAM характерну пропорційність інтервалів часу між першим і наступним звертаннями подають як 5–1–1–1, але і самі інтервали суттєво скорочуються.

MDRAM

У багатобанкових структурах *MDRAM* (*Multibank DRAM*) пам'ять розбита на банки (частини). Звертання до банків здійснюється за чергою. Доки йде обмін з одним банком, в інших закінчуються перехідні процеси. Ефект прискорення роботи такої пам'яті досягається вже при її діленні на два банки — з парними і непарними адресами. Банки типу *MDRAM* можуть будуватися на звичайних мікросхемах динамічної пам'яті без будь-яких змін.

SDRAM

Синхронна пам'ять типу *SDRAM* (*Synchronous DRAM*) — це швидкодіюча динамічна пам'ять, яка працює на частоті системної шини без тактів очікування в середині пакетного циклу. Від звичайної асинхронної динамічної пам'яті, в якій всі внутрішні процеси ініціюються тільки сигналами *RAS*, *CAS*, \overline{W}/R , пам'ять *SDRAM* відрізняється використанням сигналів тактової частоти системної шини. Це дозволяє створювати в середині мікросхеми високопродуктивний конвеєр з використанням звичайних динамічних ЕП з часом доступу 50–70 нс. Синхронний інтерфейс забезпечує трикратний виграш у продуктивності порівняно із звичайними мікросхемами *DRAM* з такою ж швидкодією.

Синхронна пам'ять типу *SDRAM* запропонована в 1994 р. як двобанкова система з триступеневим конвеєром. Вона забезпечувала пропускну здатність 250 Мбайт/с, працювала на частоті 125 МГц.

В цілому мікросхема *SDRAM* — це пристрій з програмовними параметрами, внутрішньою організацією чергування банків та власним набором команд, в тому числі

- *MRS* — програмування параметрів пакетного циклу,
- *ACIT* — активізація верхнього *T* або нижнього *B* банку і введення адреси рядка,
- *DEAC* — попередній заряд шин,
- *WRT, RED* — записування або читання і введення адреси стовпця,
- *REFR* — автоматична регенерація (рефреш) з періодом 15,6 мкс. Можливе задання рефреша з більшим періодом,
- *SLER* — самогенерація без зовнішніх сигналів, при цьому читання і записування заборонені,
- *PDE* — режим зберігання з пониженим енергоспоживанням (живлення зовнішніх буферів відключається).

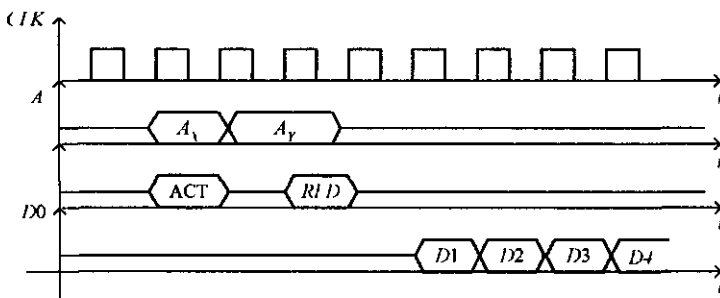


Рис 7.37. Часові діаграми роботи пам'яті SDRAM

Довжина пакетного циклу може програмуватися на 1-, 2-, 4-, 8- або 256 байт. Триступенева конвеєризація дозволяє ініціювати наступний цикл звертання до закінчення попереднього.

Часові діаграми роботи синхронної пам'яті *SDRAM* показані на рис. 7.37. Перше слово після формування адреси з'являється із запізненням на декілька тактів. Адреси наступних слів формуються внутрішнім лічильником і слова з'являються в кожному такті.

RDRAM

Мікросхеми *RDRAM (Rambus DRAM)* — це байт-послідовна пам'ять, в якій синхронізація здійснюється двома фронтами тактових імпульсів і застосовується новий інтерфейс *Rambus Channel* (ім'я фірми-розробника). У першій розробці при частоті тактування 250 МГц темп передачі байта складає 500 МГц, потім частота ще збільшується в 1,5–3 рази. Інтерфейс *Rambus Channel* має всього 13 сигнальних ліній без спеціалізованих ліній адреси. Замість звичайної адресації по інтерфейсу посилаються спочатку пакети команд, потім іде пакет підтвердження, а далі пакет даних.

Перший доступ до даних сильно запізнюється, в перших розробках до 128 нс. Пам'ять типу *RDRAM* ефективна при пакетному обміні: наприклад, для 256 байт в пакеті частота обміну дорівнює 400 МГц, а для 64 байт — 250 МГц і т.д.

Пам'ять *RDRAM* ідеально підходить для графічних і мультимедійних застосувань з типовим для них процесом — швидкою видачею послідовності слів при формуванні зображення на екрані чи для подібних задач.

DRDRAM

Пам'ять типу *DRDRAM (Direct RDRAM)* порівняно з *RDRAM* має менше запізнення при першому доступі до даних. Такі *DRDRAM* мають пропускну здатність в середині пакета 1,6 Гбайт/с. Це перевищує частотні можливості сьгоднішніх системних шин.

CDRAM

В кешованих структурах *CDRAM (Cached DRAM)* на одному кристалі з *DRAM* розміщена також статична кеш-пам'ять рівня L1. При цьому кеш забезпечує швидкий обмін з процесором при наявності інформації в кеші, а також швидке оновлення свого вмісту. Це пояснюється тим, що зв'язки між обома виводами на кристалі є внутрішніми. При цьому розрядність шин може бути великою і обмін може відбуватися великими блоками даних. Наприклад, в *CDRAM* фірми *Ramtron* застосовується 2048-розрядна шина для оновлення вмісту кеша.

7.7.4. Регенерація динамічної пам'яті

Оскільки в процесі роботи пам'яті звертання до її рядків відбувається через різні інтервали часу, то потрібна примусова регенерація. Вона складається з регулярних циклічних звернень до q рядків матриці ЕП за адресами, які формуються зовнішнім або внутрішнім лічильником адреси. Такі цикли називаються холостими, оскільки у них не відбувається обміну інформацією.

Максимальний період звертання до кожного рядка T_{RF} (refresh time) для гарантованого збереження інформації у сучасних мікросхемах пам'яті знаходиться в межах 8–64 мс. Залежно від об'єму і організації матриці пам'яті для однократної регенерації потрібно 512, 1024, 2048 або 4096 циклів звернень (тобто, за числом рядків q).

При розподіленій регенерації одиночні цикли регенерації виконуються рівномірно з періодом $t_{RI} = T_{RF}/q$, який для стандартної пам'яті дорівнює 15,6 мкс (рис. 7.38, а).

Для пам'яті з розширеною регенерацією допустимий період циклів — до 125 мкс. Можливий також варіант пакетної регенерації, коли усі цикли регенерації збираються у пакет (рис. 7.38, б), під час якого звертання до пам'яті за записуванням і зчитуванням блокується. При кількості циклів 1024 ці пакети будуть періодично займати шину пам'яті приблизно на 130 мкс, що в ряді застосувань недопустимо. З цієї причини практично завжди використовують розподілену регенерацію, хоч можливий і проміжний варіант — пакетами по декілька циклів (рис. 7.38, в)

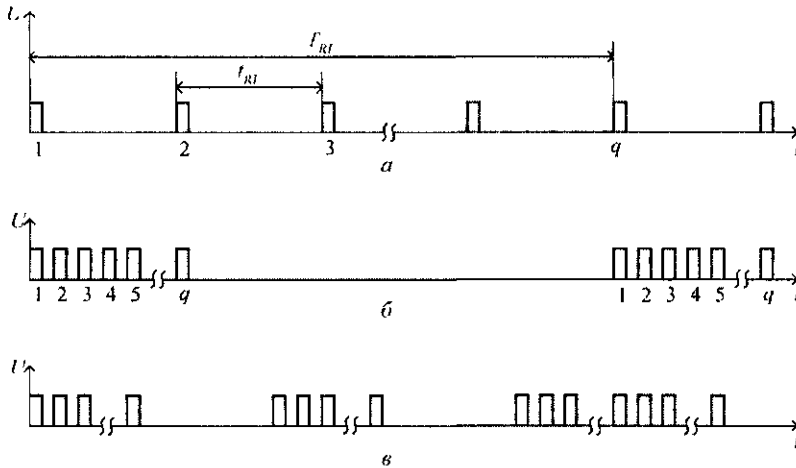


Рис. 7.38. Регенерація динамічної пам'яті: а — розподілена; б — пакетна; в — комбінована

Цикли регенерації можуть організуватися різними способами. Класичним є цикл без імпульсу \overline{CAS} , що скорочено іменується ROR (RAS Only Refresh — регенерація тільки імпульсом RAS). У цьому випадку адреса чергового регенованого рядка на магістралі адреси (MA) виробляється контролером пам'яті до спаду імпульсу RAS чергового циклу регенерації (рис. 7.39, а).

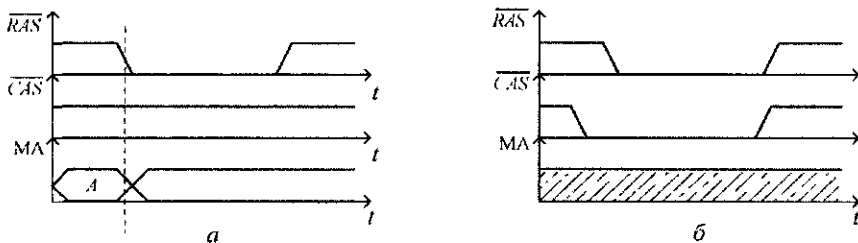


Рис. 7.39. Цикли регенерації динамічної пам'яті: а — ROR; б — CBR

Інший варіант — цикл *CBR* (*CAS Before RAS*), який підтримується практично усіма сучасними мікросхемами пам'яті (рис 7.39, б). Тут спад імпульсу *RAS* здійснюється при низькому рівні сигналу *CAS* (у звичайному циклі звертання така ситуація неможлива). У цьому випадку мікросхема виконує регенерацію рядка, адреса якого знаходиться у внутрішньому лічильнику ВІС ОЗП, і в задачу контролера входить тільки періодичне формування таких циклів. Додатковою перевагою даного циклу є економія споживаної потужності за рахунок неактивності внутрішніх адресних буферів.

Різновидом циклу *CBR* є цикл прихованої регенерації в кінці корисного циклу зчитування або записування сигнал *CAS* утримується на низькому рівні, а сигнал *RAS* підіймається і знову опускається, що є вказівкою мікросхемі пам'яті на виконання циклу регенерації за внутрішнім лічильником. При цьому після циклу зчитування вихідні буфери зберігають щойно зчитані дані (у звичайному циклі *CBR* вихідні буфери знаходяться у третьому стані).

У сучасних комп'ютерах регенерацію пам'яті бере на себе контролер і його задача — по можливості використати для регенерації цикли шини, не зайняті її абонентами (процесорами і активними контролерами).

"Найспритніші" контролери регенерації ставлять запити на регенерацію в чергу, яку обслуговують у вільний для шини час, і тільки коли запитів нагромаджується більше граничної кількості, відкладається поточний цикл обміну по шині й цикл регенерації виконується негайно.

Динамічна пам'ять, яка використовується у відеобуферах графічних адаптерів, спеціальних циклів регенерації не вимагає, оскільки частота зчитування для відображення інформації цілком достатня для збереження інформації.

7.7.5. Модуль динамічної пам'яті

Динамічні ЗП вітчизняного виробництва (1990 р.) в основному представлені мікросхемами К565РУ1 – К565РУ9, які характеризуються такими параметрами:

- ємністю — від 4 Кбіт до 4 Мбіт,
- організацією — 4 К x 1, 16 К x 1, , 256 К x 1,
- часом доступу — 150–500 нс,
- споживаною потужністю при зберіганні інформації 20 – 40 мВт, при обміні 150–400 мВт.

Покоління мікросхем динамічної пам'яті змінюються через п'ять років. Недавно група фірм *IBM*, *Siemens* та інші представили свої ультра-ВІС пам'яті з параметрами:

- технологічні норми — 0,25 мкм,
- кількість транзисторів на кристалі площею 286 мм² — 280 млн ,
- час доступу в пакетному режимі — 26 нс.

Модуль динамічної пам'яті ємністю 64 Мбіт, побудований на чотирьох мікросхемах типу 2100 фірми *Motorola* з організацією 4 М x 4 біт, показаний на рис 7.40.

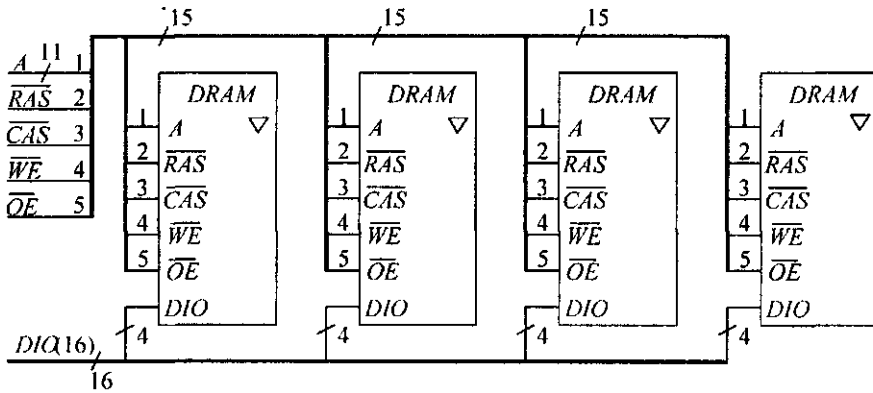


Рис. 7.40. Схема модуля динамічної пам'яті з організацією 4 М x 16 біт

Довжина адресного коду $k = 22$. Код поділений на дві рівні частини, які подаються мультиплексним способом одночасно на адресні входи мікросхем. По стробу \overline{RAS} подається старша частина адреси, по стробу \overline{CAS} — молодша.

Сигнал \overline{WE} дозволяє записувати інформацію, а \overline{OE} дозволяє записування вихідного буфера при читанні: Інформаційні входи DI і виходи DO об'єднані у спільну шину DIO . Мікросхему випускають у корпусі з 26 виводами.

Контрольні запитання

1. Охарактеризуйте поняття "пам'ять комп'ютера".
2. Назвіть основні параметри пам'яті.
3. Чому інформація в ЗП зберігається у двійковому коді?
4. Для чого призначена внутрішня пам'ять?
5. Для чого призначена зовнішня пам'ять?
6. Що таке адресний доступ до даних?
7. Охарактеризуйте принципи побудови постійної пам'яті.
8. Які переваги має динамічна пам'ять порівняно зі статичною?
9. Що таке кеш-пам'ять?
10. Охарактеризуйте флеш-пам'ять.
11. Дайте аналіз структур швидкодіючих динамічних мікросхем пам'яті.

Розділ 8

Арифметико-логічні пристрої та пристрої керування

8.1. Класифікація арифметико-логічних пристроїв

Арифметико-логічний пристрій функціонує на основі мікропрограмного керування. Кожна машинна операція розділяється на послідовність елементарних дій (передача слів, інверсія слів та ін.), які реалізуються в тактах. Елементарне функціональне обчислення, яке виконується в одному машинному такті, називається **мікрооперацією**. Кожна мікрооперація ініціюється відповідним керуючим сигналом. Сукупність мікрооперацій, які виконуються в одному такті, називається **мікрокомандою**. Зокрема, мікрокоманда може вміщувати одну мікрооперацію або ні однієї.

Для вибору порядку проходження мікрооперацій аналізуються логічні умови, які набувають значення одиниці (так) чи нуля (ні) залежно від значень операндів і результатів обчислень. Мікроалгоритм операції, записаний в термінах мікрооперацій і логічних умов, називається мікропрограмою. Кожна машинна операція має свою мікропрограму.

Будь-який цифровий обчислювач, в тому числі й АЛП, може представлятися композицією операційного і керуючого пристроїв. В операційному пристрої виконуються арифметико-логічні операції. Керуючий пристрій забезпечує виконання операцій за допомогою послідовності керуючих сигналів, яку він виробляє залежно від мікропрограми. В математичних моделях АЛП перший пристрій подається операційним автоматом, а другий — керуючим автоматом (рис. 8.1).

Операційний автомат (ОА) приймає по входу A операнди, по входу Y — керуючі сигнали $\{y_i\}$, передає на вихід Z результати операції і формує множину значень логічних умов $\{x_i\}$.

Керуючий автомат (КА) приймає по входу X логічні умови $\{x_i\}$ і залежно від їхніх значень та коду операції по входу F формує послідовність керуючих сигналів $\{y_i\}$.

Арифметико-логічні пристрої класифікують за такими ознаками:

- способом оброблення даних — паралельні, послідовні, паралельно-послідовні,
- системою числення — двійкові, вісімкові, десяткові, шістнадцяткові, а також пристрої на основі спеціальних систем (залишкових класів, зі штучним порядком ваги, чисел Фібоначі) та ін. ,
- формою подання чисел — з плаваючою комою, з фіксованою комою, цілі двійкові та десяткові числа,
- часом виконання операцій — синхронні та асинхронні,
- способом виконання мікрооперацій — із закріпленими мікроопераціями, із спільними операціями,

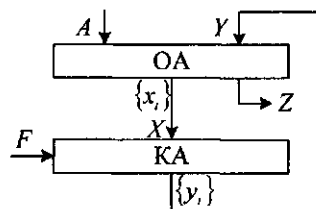


Рис. 8.1 Структура математичної моделі АЛП

- типом керуючого автомата — зі схемною або програмовною логікою;
- методом побудови — багатофункціональні та блочні.

У синхронних АЛП на виконання різних операцій відводиться один і той же інтервал часу, а в асинхронних час виконання залежить від типу операції.

В АЛП із закріпленими мікроопераціями кожен з регістрів за допомогою додаткових комбінаційних схем виконує певний визначений набір мікрооперацій. При цьому комбінаційні схеми часто повторюються, що вимагає значних апаратних витрат.

В АЛП із спільними мікроопераціями виділяють запам'ятовуючу частину — блок регістрів, в яких виконуються однорідні мікрооперації (приймання операндів, їхнє зберігання і видачу), і комбінаційну частину, в якій зосереджені всі схеми для виконання мікрооперацій (формування кодів, зсуви, додавання та ін.). Обидві частини поєднуються між собою за допомогою мультиплексорів та демультимплексорів. Арифметико-логічні пристрої із спільними мікроопераціями часто називають магістральними (до числових магістралей за чергою підключаються регістри).

Багатофункціональні (універсальні) АЛП використовуються для виконання всього списку операцій, що досягається відповідною настройкою і комутацією вузлів. Блочні АЛП складаються з окремих блоків, орієнтованих на виконання окремих типів операцій (наприклад, блок множення чисел з плаваючою комою). Такі структури використовуються у високопродуктивних комп'ютерах.

В АЛП можливі два типи КА:

- зі схемною ("жорсткою") логікою, яка складається з елементів пам'яті (тригерів) і комбінаційних схем. Вони генерують відповідні керуючі сигнали $\{y_i\}$ в машинні такти залежно від коду операції;
- з програмовною (яка зберігається в пам'яті) логікою: для кожної операції в спеціальній пам'яті (найчастіше — різні види ПЗП) записується мікропрограма у вигляді послідовностей керуючих слів — мікрокоманд. Вони містять інформацію про мікрооперації, що мають виконуватися в даному такті, та адресу наступної мікрокоманди.

Узагальнена і найбільш поширена структура АЛП показана на рис. 8.2.

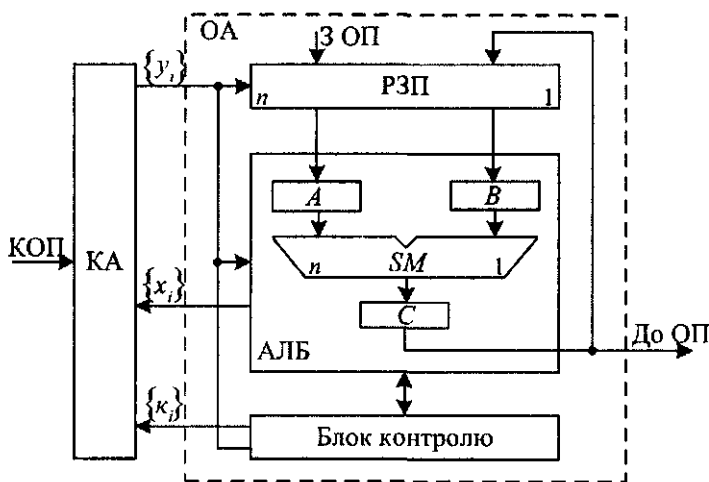


Рис. 8.2. Узагальнена структура АЛП

До складу ОА універсальних комп'ютерів входять:

- арифметико-логічний блок (АЛБ);
- набір регістрів загального призначення (РЗП);
- блок контролю.

В АЛБ виділяють комбінаційний суматор SM , вхідні регістри A і B для приймання операндів та вихідний регістр C для записування результату. В АЛБ є логічні схеми, які виробляють множини $\{x_i\}$ сигналів логічних умов (ознак результату), наприклад, нульовий або від'ємний результат та ін.

Регістри загального призначення використовують для приймання і зберігання операндів, проміжних та кінцевих результатів.

Блок контролю забезпечує перевірку правильності виконання арифметико-логічних операцій одночасною реалізацією тієї ж команди дублюючою апаратурою і порівнянням результатів або виконанням дій над спеціальними кодами, одержаними від операндів при додаванні за модулем два, три та іншими. При виявленні помилок і збоїв в роботі ОА блок контролю посилає в КА код помилок $\{k_i\}$.

На АЛП поступає код операції від центрального пристрою керування. Застосування в АЛП пристроїв керування зі схемною логікою прискорює виконання операцій. Застосування КА з програмовною логікою забезпечує гнучкість мікропрограмування, дозволяє змінювати склад мікропрограм при введенні нових команд. В сучасних АЛП можуть поєднуватись обидва типи КА.

8.2. Мови опису операційних пристроїв

Для описування операційних пристроїв на різних рівнях використовуються відповідні мови, а саме:

- мова електричних рівнянь для струмів і напруг у ланцюгах схем, складених з резисторів, діодів, транзисторів та ін.;
- мова булевих функцій для описування логічних і запам'ятовуючих елементів;
- мова мікрооперацій для описування типових функціональних комбінаційних і послідовнісних вузлів;
- мова мікропрограм для описування роботи операційних пристроїв на рівні мікроалгоритмів машинних операцій;
- алгоритмічні мови для описування обчислювального процесу в комп'ютері на рівні програм (Асемблер, Паскаль, Сі та ін.)

У мові мікрооперацій опис слова містить ідентифікатор A , B , RGA , CT і розрядний покажчик кількості розрядів $A(32)$, $B(16)$, або номери старшого (зліва) і молодшого (справа) розрядів: $A(32:1)$, $B(16:1)$. Розрядний покажчик може пропуститися, якщо слово було описане раніше. Регістр і слово в ньому часто позначають одним і тим же ідентифікатором: RGA , $A(32)$. Слово може подаватися своїми частинами: $A(32:16)$, $A(15:12)$, $A(11:1)$. Значення конкретного i -го розряду слова записують як $A[i]$, наприклад, $A[1]$, $B[n]$.

Мікрооперація описується оператором та ідентифікатором керуючого сигналу у вигляді:

$$y, : A(n) := B(n) * C(n),$$

де y , — керуючий сигнал, який відділяється від оператора двома крапками; $A(n)$ — результат; $:=$ — знак присвоєння; $*$ — вид перетворення в двомісній мікрооперації (в одномісній пропускається); $B(n)$ і $C(n)$ — n -розрядні операнди. Вираз справа від знака присвоєння називається формулою оператора. Двійкове значення, одержане на основі обчислення за формулою оператора, в кінці такту присвоюється слову $A(n)$.

Найбільш поширені такі мікрооперації:

- установка константи: $A(3:1):=111$;
- інвертування слова: $B(n):=\overline{B(n)}$;
- складання (конкатенація) слова з окремих слів чи їхніх полів: $A(16:1):=B(16:8).C(7:1)$;
- передача слів: $A(n):=B(n)$; $RGC := RGD$;
- додавання двох слів: $A := B + C$;
- інкремент та декремент слова на одиницю: $A := A + 1$, $A := A - 1$;
- порозрядні логічні операції: диз'юнкція $A := B \vee C$, кон'юнкція $A := B \wedge C$, додавання за модулем два $A := B \oplus C$;
- лівий і правий логічні зсуви L , R ; арифметичні зсуви L_A , R_A та циклічні зсуви L_C , R_C . Наприклад, логічний зсув слова $A(n)$ вліво на один розряд: $A(n):=L(A)=A(n-1:1).0$; циклічний зсув вправо на один розряд: $A := R_C(A)=A[1].A(n:2)$.

Мова, яка призначена для описування мікропрограм в термінах операторів, називається **мовою мікропрограмування**. Оператори в мікропрограмі виконуються послідовно від початку до кінця; для зміни такого порядку використовують оператори переходу і мітки.

Оператор

Перейти M ,

переносить дію до оператора з міткою M .

Оператор

Перейти, якщо x , то M ,

переносить дію до мітки M , коли $x = 1$, інакше при $x = 0$ оператори виконуються послідовно.

Оператор

Перейти, якщо x , то M , інакше M_1

переносить дію до M , при $x = 1$, інакше виконується оператор з міткою M_1 .

Мікропрограми представляють також направленим графом (рис. 8.3), який складається з вершин: "Початок", "Кінець", операторних (відповідають одній мікрокоманді кожна) та умовних (еквівалентних умовним переходам до двох міток).

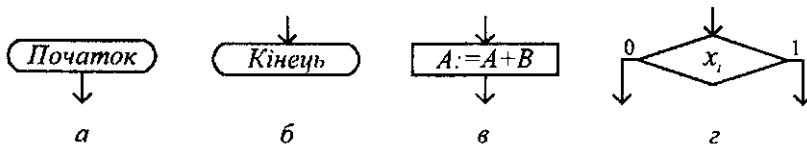


Рис. 8.3. Вершини графа мікропрограми:

a — "Початок"; $б$ — "Кінець"; $в$ — операторна; $г$ — умовна

Всі вершини, крім початкової, мають довільне (не менше одного) число входів.

При всіх можливих значеннях логічних виразів в умовних і перемикальних вершинах має існувати шлях з початкової в кінцеву вершину (рис. 8.4).

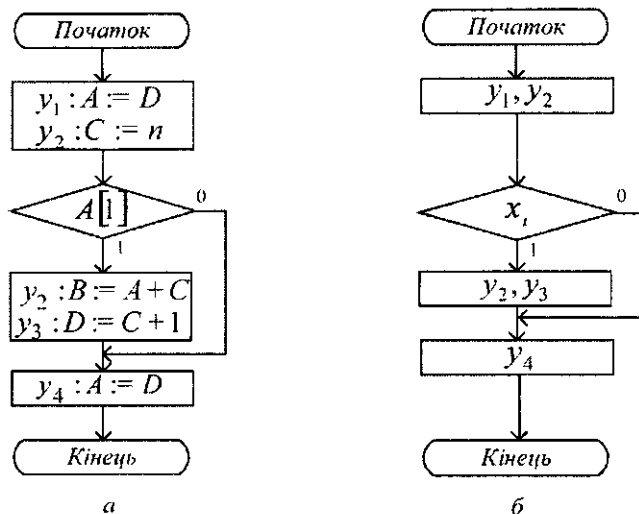


Рис. 8.4. Граф мікропрограми: а — змістовний; б — закодований

Якщо операторні й умовні вершини графа містять повний опис мікрооперацій і булевих виразів (логічних умов), то його називають **змістовним** (рис. 8.4, а). Якщо мікрооперації подані ідентифікаторами керуючих сигналів, а логічні умови — булевими аргументами, то такий граф називається **закодованим** (рис. 8.4, б).

Примітка. Проектування спеціалізованих арифметико-логічних пристроїв розглядається в наступному розділі.

8.3. Пристрої керування

Пристроєм керування називається функціональна частина комп'ютера, яка призначена для автоматичного керування обчислювальним процесом за допомогою послідовності керуючих і синхронізуючих сигналів. Пристрій керування забезпечує координацію роботи всіх функціональних вузлів комп'ютера в процесі виконання програми.

Час, протягом якого виконується одна машинна команда, називається машинним циклом. Протягом машинного циклу ПК забезпечує виконання таких дій:

- зчитування чергової команди з ОП, її дешифрацію та зберігання протягом циклу;
- формування адрес операндів з інформації, яка міститься в коді команди;
- вибірку операндів з ОП або РЗП і пересилку їх в АЛБ;
- вироблення необхідної для виконання даної команди послідовності керуючих сигналів;
- пересилку результату операції в РЗП або ОП;
- формування адреси наступної команди;
- переадресацію за безумовними і умовними ознаками;
- реалізацію пультових операцій керування (запуску, зупинки машини, авто-

матичного або покомандного виконання програми), контроль і забезпечення різних режимів роботи, пов'язаних з пристроями введення – виведення та ін

В ПК виділяють дві основні функціональні частини — програмну і мікропрограмну. Програмна частина реалізується центральним пристроєм керування (ЦПК). Він визначає послідовність виконання команд програми, здійснює розшифровку команд, виробляє виконавчі адреси, пересилає операнди в операційний пристрій і підготує його до виконання заданої операції.

Мікропрограмна частина ПК призначена для виконання мікропрограм в операційних пристроях, які є в АЛП, в каналах введення-виведення інформації, в накопичувачах на магнітних дисках і стрічках та ін

Наприклад, в мікропроцесорах може бути два АЛП цілочислової арифметики і арифметичний співпроцесор для виконання операцій з плаваючою комою. Конструктивно мікропрограмна частина реалізується у вигляді блоків місцевого керування (БМК). Таким чином, у загальному випадку ПК комп'ютера — це ієрархічна структура, яка складається із ЦПК і ряду БМК.

Розв'язання будь-якої задачі в машині зводиться до послідовної вибірки і виконання команд відповідної програми, яка організується в ЦПК. У зв'язку з цим ЦПК може бути представлений як перетворювач первинної командної інформації у вторинну, яка подає виконавчі адреси і керуючі сигнали. Крім команд, до первинної командної інформації відносяться коди і сигнали, які характеризують стан окремих вузлів, блоків і пристроїв.

Пристрої керування класифікують за такими ознаками:

- програмною орієнтацією — універсальні та спеціалізовані,
- принципом вироблення сигналів у часі — синхронні та асинхронні,
- способом побудови КА — із схемною або програмовною логікою
- способом реалізації машинних команд — централізовані або змішані,
- методом зберігання програм — з використанням ОП чи з введенням зовні,
- порядком проходження команд — з природним або довільним порядком,
- числом рівнів керування — одно- та багаторівневі.

Універсальні ПК дозволяють виконувати будь-які програми, що записані у вигляді послідовності команд з урахуванням обмежень, пов'язаних з ємністю пам'яті машини, довжини розрядної сітки і швидкодії машини.

Спеціалізовані ПК працюють за фіксованими програмами, які змінюються відповідними переключеннями в них. Спеціалізовані ПК використовують у комп'ютерах, які виконують один або визначений клас задач.

Залежно від тривалості мікрооперацій ПК поділяють на синхронні та асинхронні. У синхронних ПК частота роботи генератора синхросигналів вибирається двома способами:

- із врахуванням часу, необхідного для найтривалішої мікрооперації,
- із врахуванням часу, необхідного для найбільш розповсюдженої мікрооперації. При цьому для реалізації найтриваліших ("довгих") мікрооперацій відводять декілька тактів, наприклад, два.

В асинхронних ПК тривалість такту змінна і залежить від часу виконання поточної мікрооперації. Сигнал закінчення даної мікрооперації є сигналом початку наступної, ПК сучасних комп'ютерів використовують комбінований спосіб тактування, в якому вигідно узгоджуються переваги обох принципів.

За способом побудови розрізняють КА із схемною та програмовною логікою. Останній містить спеціальний блок пам'яті для зберігання мікропрограм.

Керуючий автомат, функціонування якого задають за допомогою мікропрограм, називається **мікропрограмним автоматом** (МПА).

Історично склалось так, що в багатьох посібниках термін "мікропрограмний автомат" використовують тільки відносно КА з програмовною логікою, що методично необгрунтовано.

За способом реалізації машинних команд ПК поділяють на централізовані та змішані. При централізованому способі керування єдиний ПК виробляє всі керуючі сигнали, які необхідні для виконання будь-якої операції із системи команд машини. Такі ПК часто використовують асинхронний принцип тактування і застосовуються переважно в одноадресних машинах із порівняно невеликим числом мікрооперацій.

При змішаному способі керування ЦПК виробляє основні керуючі сигнали — приймання команди, її дешифрації, модифікації та інші. Керування виконанням арифметико-логічних операцій забезпечується БМК.

За числом рівнів керування ПК відносять до одно- або багаторівневих. Це однаковою мірою відноситься як до ПК в цілому, так і до його частин — програмної та мікропрограмної.

На першому, найнижчому рівні, базовими операторами служать мікрооперації, а логічними умовами — повідомні сигнали з операційного пристрою. На другому рівні базові оператори і логічні умови мають такий же порядок складності, як і команди традиційних машин. На третьому рівні базові оператори аналогічні мікрокомандам однорівневих машин, але реалізуються технічними засобами і є органічною частиною ПК.

За порядком проходження команд програми розрізняють ПК з природною і з довільною послідовністю. У більшості комп'ютерів команди розміщуються в пам'яті в комірках з послідовними номерами і адреса наступної команди визначається автоматично — додаванням константи до лічильника команд. У ПК з довільним порядком адресу наступної команди вказують у поточній команді.

За методом зберігання програм виділяють машини з внутрішнім програмуванням (команди зберігаються в ОП) та із зовнішнім (програма вводиться за допомогою перфокарт, магнітних карток, спеціальних панелей та ін.).

8.4. Керуючі автомати із схемною логікою

Термін "автомат" використовують двояко. У техніці з поняттям "автомат" пов'язують деякий пристрій, який здатний виконувати визначені функції без втручання людини або за її обмеженою участю. У другому, широкому аспекті, автомат — це математична модель, яка відображає фізичні або абстрактні явища найрізноманітнішої природи (обчислювальні машини, системи керування і зв'язку, лінгвістика та ін.). Універсальність теорії автоматів дозволяє розглядати з єдиного погляду різні об'єкти, встановлювати зв'язки і аналогії між ними, переносити результати досліджень з однієї області в іншу. Узагальненим прикладом цифрового автомата є комп'ютер, який здійснює приймання, зберігання і перетворення дискретної інформації за заданими алгоритмами.

Загальна теорія автоматів поділяється на абстрактну і структурну. Абстрактна теорія вивчає поведінку автомата відносно зовнішнього середовища і не розглядає способів його побудови. Структурна теорія автоматів вивчає способи побудови логічних схем автоматів на основі алгоритму, заданого на абстрактному рівні.

Абстрактний автомат як систему задають впорядкованою сукупністю шести об'єктів $\{X, Y, Z, \delta, \lambda, z_1\}$, де:

$X = \{x_1, x_2, \dots, x_m\}$ — множина вхідних сигналів;

$Y = \{y_1, y_2, \dots, y_n\}$ — множина вихідних сигналів;

$Z = \{z_1, z_2, \dots, z_r\}$ — множина внутрішніх станів, які визначаються пам'яттю автомата;

δ — функція переходів, яка задає відображення множин $X^*Z \rightarrow Z$;

λ — функція виходів, яка задає відображення множин $X^*Z \rightarrow Y$ або $Z \rightarrow Y$;

z_1 — початковий стан автомата.

Множини X, Y, Z називаються алфавітами, а їхні елементи — буквами. Послідовності вхідних та вихідних букв створюють відповідно вхідні та вихідні слова.

Поняття "пам'ять" (внутрішні стійкі стани) автомата введено для описування систем, сигнали на виходах яких залежать як від вхідних сигналів в даний момент часу, так і від попередньої історії розвитку процесу. Взагалі множина стійких станів автомата — це сукупність поточних значень фізичних параметрів елементів пам'яті (наприклад, напруга на виходах тригерів), яка зберігається до надходження відповідного вхідного сигналу. Внутрішній стан автомата відповідає деякій пам'яті про минуле і дозволяє усувати час як явну змінну і виражати вихідні сигнали як функцію входів і пам'яті.

Автомат працює в дискретному часі й перехід із стану в стан здійснюється миттєво. За способом введення дискретного часу автомати поділяються на синхронні та асинхронні. У синхронних автоматах дискретний час задають генератором синхросигналів: $t = 0, 1, 2, \dots$, де t — номер машинного такту. В асинхронних автоматах моменти переходу із одного стану в інший попередньо не визначені й залежать від деяких подій. В таких автоматах інтервал дискретності змінний.

У теорії найповніше описані й на практиці широко застосовуються синхронні автомати. Автомат, який має початковий стан, називається **ініціальним**. У початковий момент часу $t = 0$ ініціальний автомат завжди знаходиться в початковому стані $z_1 \in Z$.

За способом формування вихідних сигналів розрізняють автомати Мілі, Мура і С-автомати. Функція переходів усіх автоматів однакова і записується у вигляді:

$$Z(t) = \delta[X(t), Z(t-1)].$$

Функції виходів задають вирази:

$Y(t) = \lambda[X(t), Z(t-1)]$ — для автоматів Мілі;

$Y(t) = \lambda[Z(t)]$ — для автоматів Мура;

С-автомат об'єднує властивості автоматів Мілі та Мура.

Абстрактні автомати Мілі та Мура характеризуються одноканальними входами і виходами (рис. 8.5).

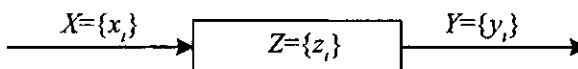


Рис. 8.5. Структура абстрактних автоматів Мілі та Мура

Функція переходів δ показує всі можливі переходи з одного стану пам'яті z_i в інший z_j під дією вхідних сигналів. Функція виходів λ задає всі можливі вихідні сигнали, які виробляються автоматом в дискретні моменти часу залежно від $x_i(t)$ та $z_i(t)$.

В автоматах Мілі вихідні сигнали є функцією вхідних сигналів і стану пам'яті. В автоматах Мура вихідні сигнали визначаються тільки станом пам'яті.

Автомат називається **скінченним**, якщо його множини X , Y і Z скінченні; інакше автомат є нескінченним. У скінченному автоматі перехід з одного стану в будь-який інший завершується за скінченне число тактів.

Ряд процесів, якими керують автомати, не вимагають для своєї роботи попередньої історії, в них вихідний сигнал $y_j(t)$ визначається тільки вхідними сигналами $x_i(t)$. Такі автомати мають тільки один стан, їх задають трійкою X, λ, Y , де функція виходів λ — це відображення виду $y(t) = \lambda [x(t)]$. Автомати з одним внутрішнім станом називаються **автоматами без пам'яті** або **комбінаційними схемами**.

На рівні абстрактної теорії функціонування автомата розглядається як перетворення вхідних букв (слів) у вихідні букви (слова).

Абстрактний автомат можна задавати за допомогою таблиць переходів і виходів, графів, матриць з'єднань або аналітичним способом. Абстрактний автомат називається **повним**, якщо його функції переходів визначені для всіх пар (x_i, z_j) і частковим — в іншому випадку.

У таблиці переходів і виходів повного автомата Мілі рядки і стовпці позначені буквами вхідних сигналів та стану пам'яті. В клітинах таблиці переходів на перетині рядка x_i і стовпця z_j записується новий стан — результат переходу $z_k = \delta(x_i, z_j)$, а в таблиці виходів — вихідний сигнал $y_m = \lambda(x_i, z_j)$. Для повного автомата Мілі з довільними множинами $X = \{x_1, x_2, x_3\}$, $Y = \{y_1, y_2, y_3\}$ і $Z = \{z_1, z_2, z_3, z_4\}$ функція переходів наведена в табл. 8.1, а функція виходів — в табл. 8.2. Часто при поданні автомата Мілі використовують одну сумісну таблицю переходів і виходів (табл. 8.3), яку називають позначеною.

Таблиця 8.1

X	Z			
	z ₁	z ₂	z ₃	z ₄
x ₁	z ₁	z ₃	z ₃	z ₃
x ₂	z ₂	z ₄	z ₁	z ₁
x ₃	z ₃	z ₂	z ₄	z ₂

Таблиця 8.2

X	Z			
	z ₁	z ₂	z ₃	z ₄
x ₁	y ₁	y ₃	y ₂	y ₂
x ₂	y ₁	y ₃	y ₁	y ₃
x ₃	y ₂	y ₃	y ₃	y ₃

Таблиця 8.3

X	Z/Y			
	z ₁	z ₂	z ₃	z ₄
x ₁	z ₁ /y ₁	z ₃ /y ₃	z ₃ /y ₂	z ₃ /y ₂
x ₂	z ₂ /y ₁	z ₄ /y ₃	z ₁ /y ₁	z ₁ /y ₃
x ₃	z ₃ /y ₂	z ₂ /y ₃	z ₄ /y ₃	z ₂ /y ₃

Вид таблиці переходів не залежить від типу автомата (Мілі, Мура чи С-автомата). У частковому автоматі Мілі з довільними множинами X, Y і Z для невідзначеного переходу ставиться прочерк, і будь-яка вхідна буква, що призводить до цього, заборонена. В клітинах таблиці виходів часткового автомата Мілі може стояти прочерк, що означає відсутність вихідного сигналу. При цьому прочерк обов'язково ставиться в тих клітинах таблиці виходів, які відповідають таким же клітинам з прочерком в таблиці переходів.

Таблиця 8.4

X	Z/Y			
	z ₁	z ₂	z ₃	z ₄
x ₁	z ₂ /y ₁	—	z ₃ /y ₃	—
x ₂	—	z ₄ /y ₂	z ₁ /—	z ₂ /y ₃

Позначена таблиця переходів часткового автомата Мілі зведена в табл. 8.4.

У таблиці виходів повного автомата Мура з множинами $X = \{x_1, x_2\}$, $Y = \{y_1, y_2, y_3\}$ і $Z = \{z_1, z_2, z_3, z_4\}$ кожному стану пам'яті приписують свій вихідний сигнал, який не залежить від букв вхідного алфавіту. Прочерки в деяких клітинах таблиці виходів часткового автомата не пов'язані з прочерком в його таблиці переходів. Позначена таблиця довільного часткового автомата Мура зведена в табл. 8.5.

При графічному способі описування абстрактний автомат Мілі подається орієнтованим графом, в якому стани зображаються вершинами графа, а переходи між станами — дугами з позначенням вхідного та вихідного сигналів. Приклад графа автомата Мілі з позначеною табл. 8.3 показаний на рис. 8.6, а. При поданні графом абстрактного автомата Мура вихідні сигнали записуються разом з вершинами станів. Приклад автомата Мура, заданого табл. 8.5, показаний на рис. 8.6, б.

Таблиця 8.5

X	Z/Y			
	z ₁	z ₂	z ₃	z ₄
	y ₁	y ₂	y ₂	y ₃
x ₁	z ₂	—	z ₃	—
x ₂	—	z ₄	z ₁	z ₂

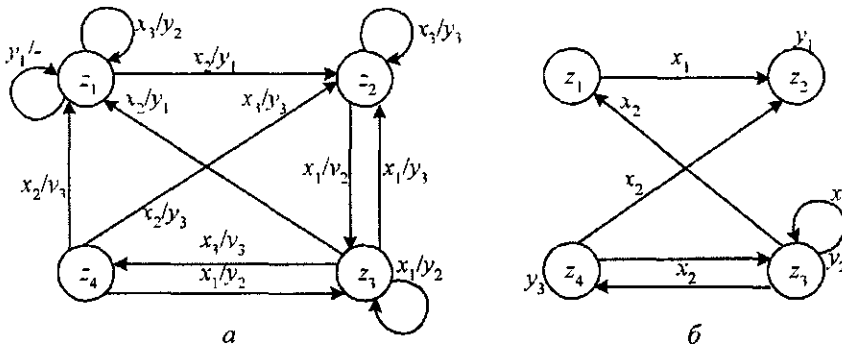


Рис. 8.6. Графи автоматів: а — Мілі; б — Мура

8.5. Структурний синтез керуючого автомата зі схемною логікою

Академік В.М. Глушков розробив канонічний метод структурного синтезу цифрових пристроїв, в якому закон функціонування абстрактного автомата реалізується комбінаційною схемою і набором тригерів. Процес побудови такої схеми називають структурним синтезом.

Набір тригерів і логічних елементів є структурно повним, якщо на їхній основі можна побудувати будь-який автомат. Для цього такий набір повинен містити функціонально повну систему логічних елементів і хоч би один тригер з повною системою переходів і виходів.

При структурному синтезі автоматів використовують *RS*-, *JK*-, *D*-, і *T*-тригери, які є елементарними автоматами Мура з повними системами переходів і виходів. Вони мають два внутрішніх стани, яким відповідають два різних сигнали. Це дає можливість позначати стани, вхідні й вихідні сигнали тригерів двозначним структурним алфавітом (символами 0 і 1), як показано в табл. 8.6 для основних типів тригерів.

Таблиця 8.6

RS-тригер				JK-тригер				D-тригер				T-тригер			
R	S	Q		J	K	Q		D	Q		T	Q			
		0	1			0	1		0	1		0	1		
0	0	0	1	0	0	0	1	0	0	0	0	0	1		
0	1	1	1	0	1	0	0	1	1	1	1	1	0		
1	0	0	0	1	0	1	1								
1	1	-	-	1	1	1	0								

Скінченні автомати Мілі й Мура є основою для побудови керуючих автоматів із схемною логікою.

Для побудови структурного автомата необхідно мати пам'ять і дві комбінаційні схеми: КС1 — для вироблення функцій збудження і КС2 — для формування вихідних керуючих сигналів (рис. 8.7).

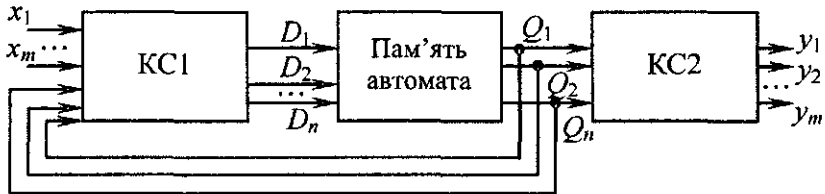


Рис. 8.7. Схема структурного автомата

Структурний синтез МПА із схемною логікою містить такі етапи:

1. Розробка мікропрограми операції та записування її на мові мікрооперацій.
2. Побудова змістовного графа мікропрограми.
3. Побудова закодованого графа мікропрограми (рис. 8.8, а).

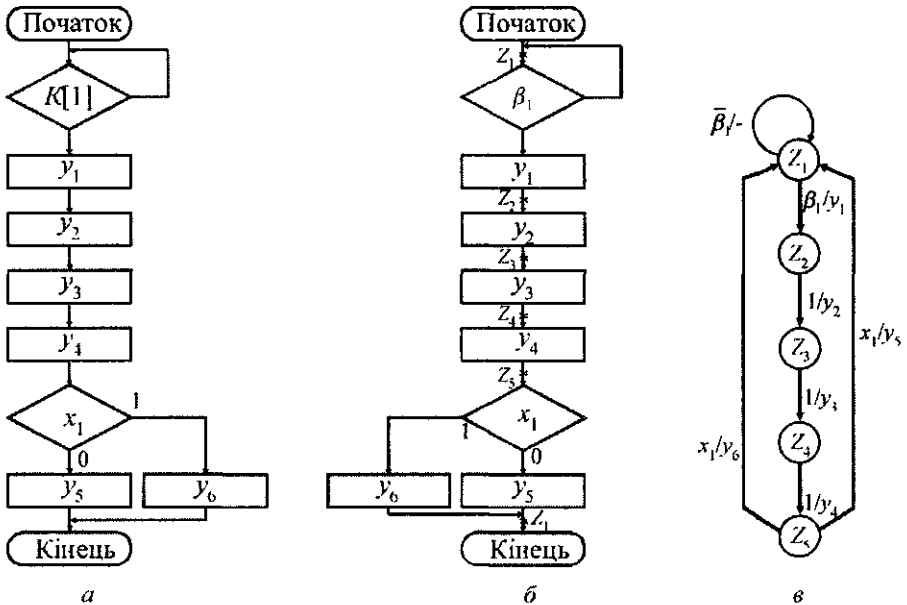


Рис. 8.8. Графи операції додавання:

а — закодований; б — розмічений; в — переходи і виходи автомата Мілі

4. Для одержання числа станів пам'яті N автомата виконують розмітку закодованого графа. Розмітку графа мікропрограми для автомата Мілі виконують за такими правилами (рис. 8.8, б):
- символом стану z_1 позначають вихід вершини "Початок" та вхід вершини "Кінець";
 - виходи операторних вершин позначають символами z_2, z_3, \dots, z_N , де індекс N визначає максимальне число станів пам'яті автомата Мілі (рис. 8.8, б).
5. Будують граф автомата Мілі (рис. 8.8, в):
- зображають N вершин z_1, z_2, \dots, z_N ;
 - шлях між двома вершинами включає одну операторну і довільне число умовних вершин і зображається орієнтованою дугою. Між двома вершинами може бути декілька шляхів і відповідно — орієнтованих дуг;
 - біля дуги записують кон'юнкцію довільного числа сигналів x_i , які записані в умовних вершинах на даному шляху розміченого графа;
 - для виходу з умовної вершини, позначеної одиницею, записують x_i , а позначеної нулем — \bar{x}_i ;
 - біля сигналів умов на дузі записують перелік керуючих сигналів y_i ;
 - при відсутності на шляху умовних вершин замість них записують одиницю;
 - у разі відсутності керуючих сигналів записують прочерк.
6. Розмічаючи граф мікропрограми для автомата Мура, символом z_1 позначають вершини "Початок" і "Кінець". Всі інші операторні вершини позначають символами z_2, z_3, \dots, z_L , де L — максимальне число станів пам'яті. Кожному стану z_i приписують свій набір вихідних сигналів (рис. 8.9, а). На дугах графа переходів автомата Мура записують тільки сигнали логічних умов, а сигнали керування проставляють біля вершин графа (рис. 8.9, б).

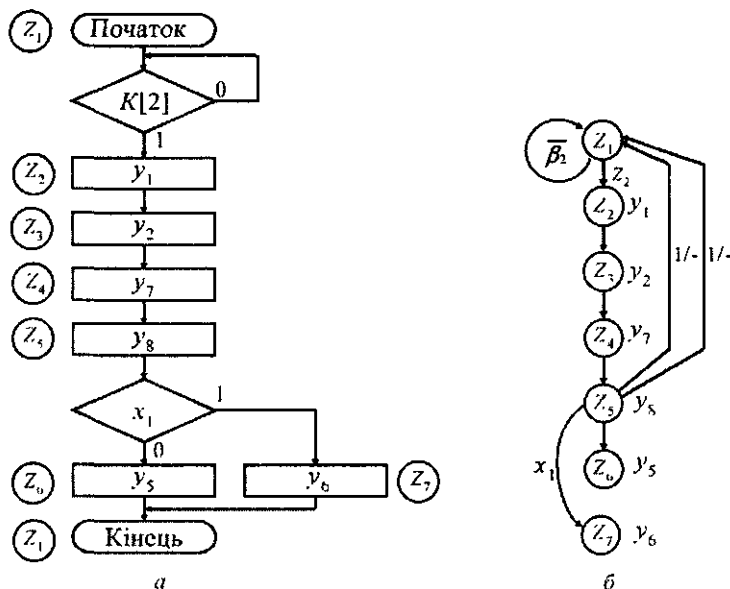


Рис. 8.9. Графи операцій віднімання:

а — закодований і розмічений; б — переходи і виходи автомата Мура

7. Кодування станів пам'яті автомата виконують двійковими наборами сигналів Q_n, Q_{n-1}, \dots, Q_1 з виходів тригерів. При кодуванні станів позиційним кодом кількість тригерів $n = \lceil \log_2 M \rceil$ (для автомата Мілі) і $n = \lceil \log_2 L \rceil$ (для автомата Мура). Звичайно число станів автомата Мура більше числа станів автомата Мілі.
8. При побудові комбінаційних частин автомата для формування вихідних сигналів та функцій збудження входів тригерів пам'яті використовують пряму структурну таблицю МПА, яка містить у своїх стовпцях:
 - послідовність попередніх станів z_i , та їхній код $K[z_i]$;
 - послідовність наступних станів z_j , та їхній код $K[z_j]$;
 - набори вхідних сигналів $\{x_i\}$, що визивають перехід (z_i, z_j) , та вихідних сигналів $y_\alpha, y_\beta, \dots, y_\omega$, які формуються на даному переході;
 - набори інформаційних входів тригерів, що мають переключатися на переході (z_i, z_j) . Якщо використовують *RS*- і *JK*- тригери, то вони мають по два інформаційних входи, а якщо *D*- і *T*- тригери, то вони мають по одному входу. Переключення тригерів на переходах здійснюються згідно з даними табл. 8.6.

8.6. Синтез мікропрограмного автомата з програмовною логікою

Мікропрограмний автомат з програмовною логікою будують на основі операційно-адресної структури з використанням загальних принципів програмного керування. При цьому алгоритм керування представляють впорядкованими наборами керуючих слів-мікрокоманд. Вони визначають порядок функціонування дискретного пристрою протягом машинного циклу. Сукупність із P k -розрядних мікрокоманд створює їхній масив, який зберігається в пам'яті мікрокоманд ПМК[$P:1$] автомата.

Структура МПА з програмовною логікою містить (рис. 8.10):

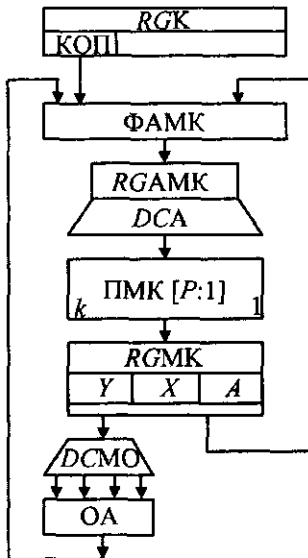


Рис. 8.10. Структура МПА з програмовною логікою

- формувач адреси мікрокоманди (ФАМК);
- реєстр адреси мікрокоманди (*RGAMK*);
- дешифратор адреси мікрокоманд (*DCA*);
- пам'ять k -розрядних мікрокоманд ПМК[$P:1$] (k);
- реєстр мікрокоманд *RGMK*, який розділений на поля мікрооперацій Y (операційна частина), логічних умов X та адреси A ;
- дешифратор мікрооперацій *DCMO*, на виході якого формуються керуючі сигнали для OA ; в деяких типах МПА він відсутній.

В загальному випадку мікрокоманда містить інформацію про мікрооперації, які мають виконуватися в даному такті, адресу наступної мікрокоманди і логічні умови, що аналізуються.

Виконання кожної мікропрограми зводиться до послідовності таких дій

- зчитування мікрокоманд з ПМК за адресою, яку формує вузол ФАМК Адресу першої мікрокоманди задають кодом операції КОП в реєстрі команд *RGK* машини,
- записування зчитаної мікрокоманди в реєстр *RGMK*, дешифрація змісту полів Y та X і видача набору керуючих сигналів в ОА,
- формування адреси наступної мікрокоманди з урахуванням полів X та Y
- вироблення сигналу "Кінець" мікропрограми

Мікропрограмні автомати з програмовною логикою класифікують за такими ознаками

- типом пам'яті мікрокоманд — статичні (динамічні) і постійні (масочні, програмовні та ін),
- способом кодування мікрооперацій — горизонтальні, вертикальні, горизонтально-вертикальні, вертикально-горизонтальні,
- часом виконання мікрокоманд — синхронні та асинхронні й наявністю фаз в такті — одно- і багатофазні
- способом адресації — природні, довільні (примусові) та способом організації умовних і безумовних адресних переходів,
- напрямком удосконалення структури і алгоритмів функціонування

Як ПМК використовують різні типи постійної пам'яті, а також ОП, яка завантажується мікропрограмами з гнучких магнітних дисків при кожному включенні машини

Розрізняють одно- і багатофазні мікрокоманди У першому випадку всі мікрооперації, які вказані в мікрокоманді, виконуються одночасно за один такт У другому — такт розбивається на інтервали, які називаються фазами або мікротактами, при цьому вказані в мікрокоманді мікрооперації виконуються в різних фазах

При горизонтальному кодуванні поле Y містить M розрядів, де M — загальне число мікрооперацій (рис 8 11, а) Якщо в розряді стоїть одиниця, то відповідна мікрооперація виконується незалежно від значення інших розрядів Достоїнством горизонтального кодування є можливість одночасного виконання в одному такті мікрокоманди з будь-яким набором з M мікрооперацій і простота формування керуючих сигналів — вони представляються на виходах розрядів поля Y Недоліком горизонтального кодування є велика довжина мікрокоманди, оскільки число M може досягати більше 100 значень

При вертикальному кодуванні вся множина із M мікрооперацій кодується n -розрядним двійковим позиційним кодом (рис 8 11, б) Наприклад, для $M = 60$ маємо довжину коду $n = \lceil \log_2 M \rceil = 6$ При даному кодуванні в кожному такті виконується мікрокоманда з однією мікрооперацією Достоїнством вертикального кодування є коротка мікрокоманда, а недоліком — необхідність використання складних дешифракторів та зменшення швидкодії автомата

При горизонтально-вертикальному кодуванні операційна частина Y мікрокоманди розбивається на H полів Y_1, Y_2, \dots, Y_H (рис 8 11, в) В кожному полі горизонтально розміщуються множини з M мікрооперацій y_1, y_2, \dots, y_M , яким присвоєні номери $1, 2, \dots, M$ Ці номери кодуються вертикальним способом — n -розрядним двійковим кодом Якщо поле Y_i пуста (містить тільки нулі), то воно не ініціює жодної мікрооперації Таким чином, залежно від коду в полях Y_i мікрокоманда може одночасно ініці-

ювати від нуля до H мікрооперацій. Кожне поле Y_i дешифрується дешифратором на n входів і M виходів.

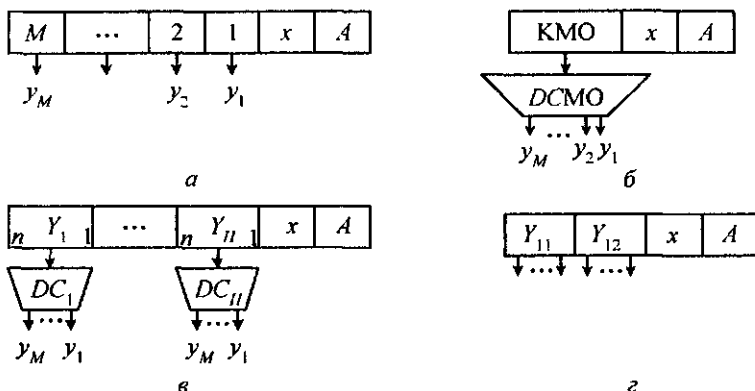


Рис. 8.11. Кодування мікрооперацій: а — горизонтальне; б — вертикальне; в — горизонтально-вертикальне; г — вертикально-горизонтальне

При вертикально-горизонтальному кодуванні (рис. 8.11, г) вся множина з M мікрооперацій розподіляється на k підмножин. В кожній з них об'єднуються мікрооперації, які найчастіше виконуються разом в одному такті. Підмножини по змозі створюють рівнопотужними — з однаковим числом мікрооперацій.

Операційна частина Y мікрокоманди складається з двох полів. У першому полі Y_{11} використовується горизонтальний спосіб кодування, а друге поле Y_{12} показує, до якої підмножини належить мікрооперація у першому полі.

Спосіб адресації задає правила визначення адреси наступної мікрокоманди. Є такі способи адресації:

- природна з послідовним розміщенням мікрокоманд;
- з довільним порядком проходження мікрокоманд;
- відносна — довільна в діапазоні адрес, визначених базовою адресою;
- за фіксаторами — адреса наступної мікрокоманди визначається змістом спеціальних комірок ПМК.

У кожному із способів адресації можна використовувати горизонтальні, вертикальні та змішані способи кодування.

Природна адресація мікрокоманд реалізується таким чином

1. Використовують два формати мікрокоманд (рис. 8.12): операційний та адресний. Формати відрізняються наявністю нуля в старшому k -му розряді операційної мікрокоманди і одиниці — в адресній.

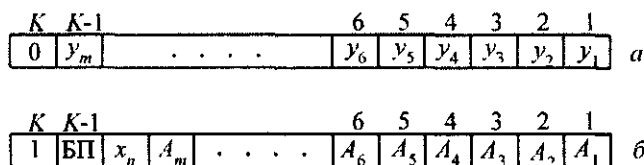


Рис. 8.12. Формати мікрокоманди: а — операційної; б — адресної

2. В операційній мікрокоманді записуються мікрооперації $\{y_i\}$, які мають бути виконані в даному такті. Природна адресація забезпечується лічильником адреси мікрокоманд СТАМК, вміст якого автоматично збільшується на одиницю після виконання поточної мікрокоманди.

В адресній мікрокоманді записують три параметри: однорозрядні значення логічної умови x_i (береться з графа автомата) і безумовного переходу БП та адреси переходу A_m, A_{m-1}, \dots, A_1 . Адресу наступної мікрокоманди задають виразом

$$R = \text{МК}[K]x_i x'_i \vee \text{МК}[K] \text{БП},$$

де $\text{МК}[K]$ — значення старшого розряду мікрокоманди, x'_i — логічна умова з виходу ОА.

Таким чином, якщо $R = 0$, то відбувається природна адресація; при $R = 1$ адреса наступної мікрокоманди визначається її адресним полем A_m, A_{m-1}, \dots, A_1 . Описаний метод дозволяє використовувати ряд логічних умов, але одиничне значення має сприймати тільки одне x_i .

Схема МПА, який використовує природну адресацію та горизонтальне кодування, показана на рис. 8.13.

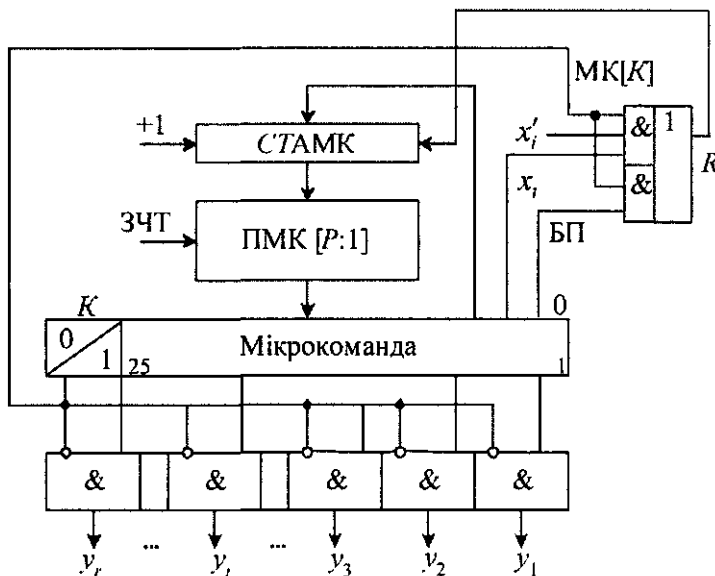


Рис. 8.13. Схема МПА з природною адресацією і горизонтальним кодуванням

У даній схемі при $\text{МК}[K] = 1$ блокується видача керуючих сигналів y_r, \dots, y_1 і здійснюється умовний перехід. При $\text{МК}[K] = 0$ дозволяється видача керуючих сигналів. Інформація з пам'яті мікрокоманд зчитується за сигналом "ЗЧТ".

Схема МПА з природною адресацією і вертикальним кодуванням має вихідний дешифратор мікрооперацій. Поле мікрооперацій, як і поле адреси, кодується двійковим позиційним кодом, що значно скорочує довжину мікрокоманди (рис. 8.14). Адресація в схемі автомата на рис. 8.14 реалізована аналогічно схемі рис. 8.13.

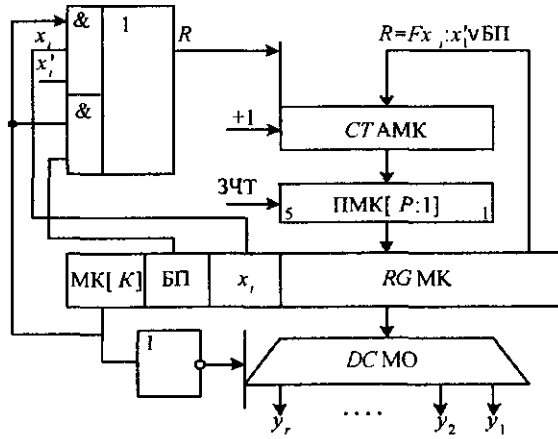


Рис. 8.14. Схема МПА з природною адресацією і вертикальним кодуванням для виконання мікропрограми додавання

Примусова адресація передбачає наявність в єдиному форматі мікрокоманди одного або двох адресних полів, в яких вказують адресу наступної мікрокоманди. Якщо є одне поле адреси, то для виконання умовних і безумовних переходів використовують додаткову комбінаційну схему для інкременту на одиницю змісту адресного поля.

Схема МПА з горизонтально-вертикальним кодуванням мікрооперацій і примусовою адресацією за допомогою двох адресних полів $A0$ і $A1$ показана на рис. 8.15. При такій структурі мікрокоманди в кожному такті може виконуватися не більше двох мікрооперацій.

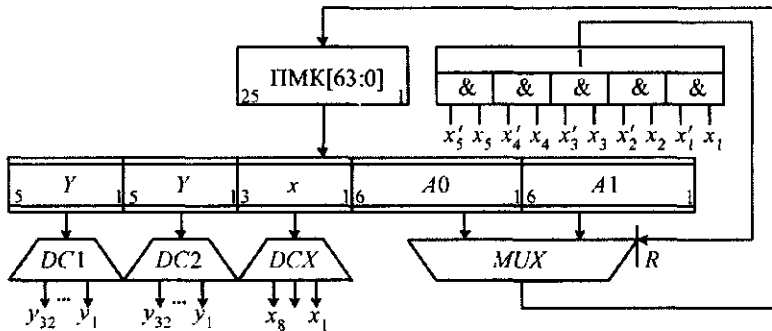


Рис. 8.15. Схема МПА з горизонтально-вертикальним кодуванням і примусовою адресацією

На основі закодованого графа мікропрограми для операції ділення в табл. 8.7 розроблено кодування мікрокоманд для реалізації МПА з горизонтальним кодуванням і примусовою адресацією.

Таблиця 8.7

Адреса ПМК	Y_1	Y_2	X	$A0$	$A1$
A_{28}	y_{16}	y_{17}	0		A_{29}
A_{29}	y_{12}		0		A_{30}
A_{30}	y_{20}	y_{21}	x_4	A_{31}	A_{32}

Таблиця 8.7. Продовження

Адреса ПМК	Y_1	Y_2	X	A_0	A_1
A_{31}	y_{22}		0		Кінець
A_{32}	y_{16}		0		A_{33}
A_{33}			x_4	A_{34}	A_{35}
A_{34}	y_{18}		0		A_{36}
A_{35}		y_{19}	0		A_{36}
A_{36}	y_{20}	y_{21}	0		A_{37}
A_{37}		y_{14}	0	A_{33}	A_{38}
A_{38}		y_{23}	x_4	A_{41}	A_{39}
A_{39}		y_{19}	0		A_{40}
A_{40}	y_{20}		0		A_{41}
A_{41}		y_{24}	0		A_{42}
A_{42}	y_{31}	y_{31}			

У п'ятирозрядних операційних полях Y_1 та Y_2 можна закодувати вертикальним способом до 32 мікрооперацій. У трьохрозрядному полі X кодують до восьми логічних умов. Шестирозрядні адресні поля A_1 і A_0 забезпечують звертання до ПМК [63:0](25). Параметри МПА (рис. 8.15) передбачають виконання мікрооперацій додавання, віднімання, множення та ділення.

Мікрооперації y_j , логічні умови x_j , адреси в полях A_0 та A_1 кодуються двійковим позиційним кодом. Десятковий еквівалент його використовується як індекс відповідного сигналу мікрооперації та адреси.

Значення $y_{32} = 1$ означає кінець операції. Адреси A_0 або A_1 передаються через мультиплексор MUX на вхід ПМК залежно від виконання умови $F = \bigvee x_i x'_i \vee \text{БП}$. В полі логічних умов при необхідності вказують на одиничне значення тільки однієї з умов x_1, x_2, \dots, x_8 , яку порівнюють з відповідними фізичними умовами x'_1, x'_2, \dots, x'_8 від ОА. Таким чином, якщо $F = 0$, то адреса вибирається з поля A_1 , інакше — з поля A_0 . При безумовному переході БП = 1 адреса визначається полем A_0 .

Мікропрограмні автомати з програмовою логікою характеризуються часом, який витрачається на формування однієї мікрокоманди. Цей час складається з трьох компонентів:

- часу формування адреси наступної мікрокоманди;
- часу звернення до ПМК;
- часу дешифрації операційної частини мікрокоманди.

Основна доля часу приходить на зчитування мікрокоманди з ПМК, тому підвищення швидкодії може досягатися або за рахунок зменшення часу звернення до пам'яті, або за рахунок зменшення числа таких звернень.

Збільшення швидкодії МПА з програмовою логікою досягається за таких умов:

- застосування швидкодіючої ПМК;
- паралельної вибірки декількох мікрокоманд;
- випереджувальної вибірки мікрокоманд (тобто до закінчення виконання попередньої).

Використання МПА з програмовною логікою порівняно зі схемною має такі переваги:

- забезпечується більша гнучкість і наглядність, підвищується регулярність структури МПА і процесора в цілому;
- підвищується ефективність математичного забезпечення і продуктивність роботи процесора;
- полегшується модернізація мікропрограм МПА як в процесі автоматизованого проектування, так і при роботі комп'ютера, що збільшує строк морального старіння машини;
- полегшується побудова ефективної мікродіагностики.

Мікропрограмний автомат з програмовною логікою має найширше застосовується в процесорах з малою і середньою продуктивністю. Обидва способи побудови автоматів керування одержали подальший розвиток на основі програмовних ВІС, однорідних середовищ та ін.

8.7. Центральний пристрій керування

Пристрій, який виконує основні функції керування комп'ютером, називається **центральним пристроєм керування**. Під ЦПК розуміють сукупність вузлів і блоків процесора, які забезпечують координацію функціонування всіх пристроїв машини і керування ними для всіх прийнятих режимів роботи.

Центральний пристрій керування реалізує системні й робочі програми, організує всі необхідні дії з оцінки і перетворення початкової інформації для одержання результату обчислень.

Розв'язання будь-якої задачі зводиться до послідовності вибірки і виконання команд програми під керуванням ЦПК. Отже, ЦПК — це перетворювач первинної командної інформації, яка представляється командами програми, у вторинну командну інформацію — виконавчі адреси і керуючі сигнали. До первинної командної інформації відносяться також коди та сигнали, які характеризують стан процесора і окремих блоків. Часто в міні- і мікрокомп'ютерах, основною вимогою яких є мінімум вартості, обладнання суміщують за функціональним призначенням, наприклад, ЦПК і МПА процесора.

В ПК умовно виділяють дві основні частини — мікропрограмну і програмну. Мікропрограмна частина — це МПА, який виробляє сигнали керування мікроопераціями в АЛП. Програмна частина визначає послідовність виконання команд програми, їхню дешифрацію, виробляє виконавчі адреси, за якими зчитуються з пам'яті операнди в АЛП і записується результат операції. Функції програмної частини реалізує ЦПК.

Для виконання своїх функцій ЦПК вміщує (рис. 9.15):

- реєстр команд *РГК* з полем коду операції КОП і полем адреси АДР;
- лічильник адреси команд *СТАК*;
- керуючий автомат МПА;
- дешифратор коду операцій *ДСКОП*;
- операційний блок (ОБ), в який входять суматор адреси, схеми аналізу режимів роботи: готовності пам'яті та периферії до обміну інформацією, запитів переривань і прямого доступу до пам'яті; інтерфейсні схеми та ін.;
- пульт керування "Пульт".

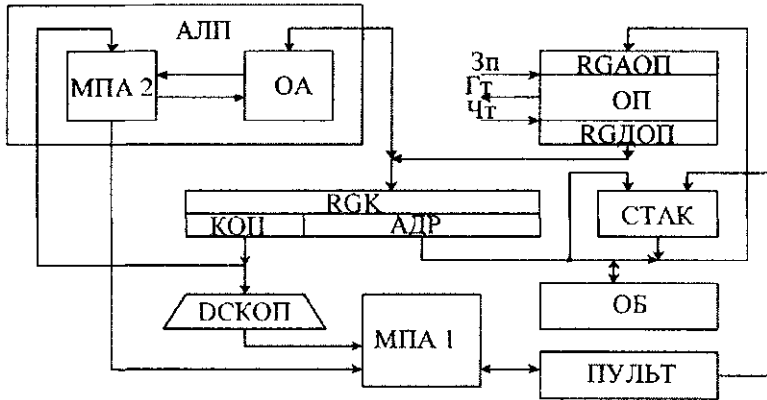


Рис. 8.16. Структура взаємодії ПК, АЛП і пам'яті

Регістр команд *RGK* призначений для приймання команди з ОП і зберігання її протягом робочого циклу. Залежно від типу машини і складності операції команда може мати довжину від одного до 10 і більше байтів. Залежно від довжини шини вибірки даних за одне звернення до ОП може зчитуватися вся команда, її частина або декілька команд.

Лічильник адрес команд *СТАК* призначений для визначення адреси команди. Після зчитування поточної команди зміст *СТАК* автоматично збільшується на константу, яка дорівнює довжині команди в байтах.

Мікропрограмний автомат *МПА1* розшифровує команди і забезпечує керуючими сигналами виконання програмної частини, а *МПА2* — виконання власне мікропрограм операцій в АЛП. Кожний з автоматів може будуватися на основі схемної або програмовної логіки. При централізованому керуванні обидва автомати об'єднуються в єдиний МПА.

Суматор адреси служить для формування виконавчих адрес операндів та результату операції за інформацією, яка міститься в коді команди. Загалом виконавчі адреси одержують додаванням трьох компонент: базової адреси та індексу, які розміщені в блоці РЗП, і коду зміщення в команді.

Пульт керування "Пульт" призначений для керування роботою комп'ютера користувачем. Він містить клавіатуру, перемикальні кнопки та засоби індикації для візуального контролю стану окремих пристроїв і проведення профілактики. На клавіатурі "Пульту" набирається команда введення та адреса першої команди програми, яка має виконуватися після натиснення на кнопку ЗАПУСК (або будь-якої клавіші). Ця інформація надходить в ЦПК, який керує введенням з магнітних дисків програми та початкових даних в ОП. Після закінчення введення програми ЦПК пересилає в лічильник *СТАК* адресу першої команди, яка має виконуватися.

Функціонування комп'ютера складається з робочих циклів, кожний з яких відповідає виконанню однієї команди програми.

У кожному робочому циклі загалом виконуються такі типові дії:

- вибірка з комірки ОП команди, яка має виконуватися, і формування адреси наступної команди. При цьому зміст лічильника адреси *СТАК* пересилається в регістр адреси пам'яті *RGАОП*. Зчитаний за даною адресою код надхо-

дить до регістра даних пам'яті, а звідти пересилається до регістра команд *RGK* після чого зміст лічильника адреси *CTAK* збільшується на константу — довжину команди в байтах,

- формування виконавчих адрес і зчитування за ними операндів з ОП. Зміст адресної частини команди пересилається в ОБ, де виробляються виконавчі адреси. Зчитані операнди надходять до ОА і зберігаються в блоці РЗП,
- розшифрування коду операції в МПА2 виконується послідовність мікрооперацій, які визначені мікропрограмою даної операції та записування результату операції в пам'ять чи в РЗП,
- вироблення в МПА2 сигналу кінця операції. Перехід до п. 1

При кожному зверненні до пам'яті перевіряється сигнал готовності "Гт" пам'яті до обміну інформацією. Обмін (зчитування чи записування) можливі при "Гт" = 1, інакше — процесор переходить в режим чекання. Після кожного звернення до ОП перевіряється наявність сигналу запиту прямого доступу до пам'яті. Якщо такий сигнал є, то процесор перемикає свої шини в третій стан і забезпечує периферійним пристроям режим прямого доступу до пам'яті. Після закінчення кожної команди перевіряється наявність сигналу запиту на переривання програми від периферії.

При запиті на керування виконується процедура реалізації підпрограми обслуговування того зовнішнього пристрою, який встановив запит на переривання. Після обслуговування зовнішнього пристрою комп'ютер повертається до виконання перерваної програми.

Контрольні запитання

1. Охарактеризуйте призначення АЛП
2. Поясніть структуру АЛП у вигляді композиції ОА і КА
3. Перелічіть класифікаційні ознаки АЛП
4. Охарактеризуйте мову мікрооперацій
5. Охарактеризуйте функції ПК
6. Що таке програмна та мікропрограмна частини ПК?
7. Поясніть функцію МПА
8. Укажіть на відмінності побудови КА зі схемною та програмовною логікою
9. Дайте порівняльну характеристику горизонтального та вертикального кодування в мікрокомандах
10. Поясніть структуру ЦПК
11. Що таке структурний синтез автомата?
12. Запишіть функцію переходів автомата Мілі
13. Запишіть функцію переходів автомата Мура
14. Поясніть функцію виходів автомата Мілі
15. Поясніть функцію виходів автомата Мура
16. Нарисуйте схему структурного автомата
17. Що таке ініціальний автомат?
18. Як проводиться розмітка мікропрограми для графа Мілі?
19. Як проводиться розмітка мікропрограми для графа Мура?
20. Поясніть структурну таблицю автомата Мілі
21. Поясніть структурну таблицю автомата Мура

Розділ 9

Проектування спеціалізованих арифметико-логічних пристроїв

9.1. Реалізація операції додавання

9.1.1. Початкові дані

- Тип арифметичної операції — додавання двійкових чисел;
- початковий код подання операндів — доповняльний;
- розрядність операндів — 8 біт;
- код виконання операції у суматорі — доповняльний модифікований;
- структура операційного блока — із закріпленими мікроопераціями;
- тип керуючого блока — автомат Мура з пам'яттю на JK-тригерах;
- схема логічної ознаки переповнення розрядної сітки;
- схема логічного порозрядного додавання кодів вхідних операндів A і B .

9.1.2. Елементна база — інтегральні схеми ТТЛШ серій К1531, КР1533

Перелік обов'язкового графічного матеріалу:

- граф-схеми мікропрограми додавання;
- схема електрична функціональна;
- схема електрична принципіальна;
- перелік елементів (специфікація).

Потрібно виконати розрахунки:

- кількості мікросхем у пристрої АЛП;
- споживаної потужності;
- швидкодії АЛП на прикладі операції додавання типу "регістр–регістр".

9.1.3. Алгоритм додавання двійкових чисел

Додавання і віднімання двійкових чисел можна виконувати в обернених або доповняльних кодах та їх модифікаціях. У сучасних комп'ютерах часто операнди зберігаються в пам'яті в доповняльних кодах. Використання доповняльних кодів в операціях додавання і віднімання та для зберігання операндів у пам'яті має такі переваги:

- однозначне подання знака результату як додатного, так і від'ємного;
- під час записування в пам'ять від'ємного результату не витрачається час для його перетворення в прямий код;
- менше дій для аналізу знака результату, зокрема переповнення розрядної сітки.

Алгоритм додавання двійкових чисел:

- у першому і другому машинних тактах із вхідної шини паралельним кодом записуються операнди A і B у відповідні регістри RGA і RGB . Зчитування операндів здійснюється ЦПК;
- протягом одного машинного такту виконується мікрооперація додавання;
- за відсутності переповнення розрядної сітки результат записується у регістр RGC ;
- за наявності переповнення результат не фіксується і в ЦПК подається сигнал переповнення ПП.

9.1.4. Функціональна схема АЛП для виконання операції додавання

Функціональну схему восьмирозрядного АЛП для виконання операції додавання показано на рис. 9.1.

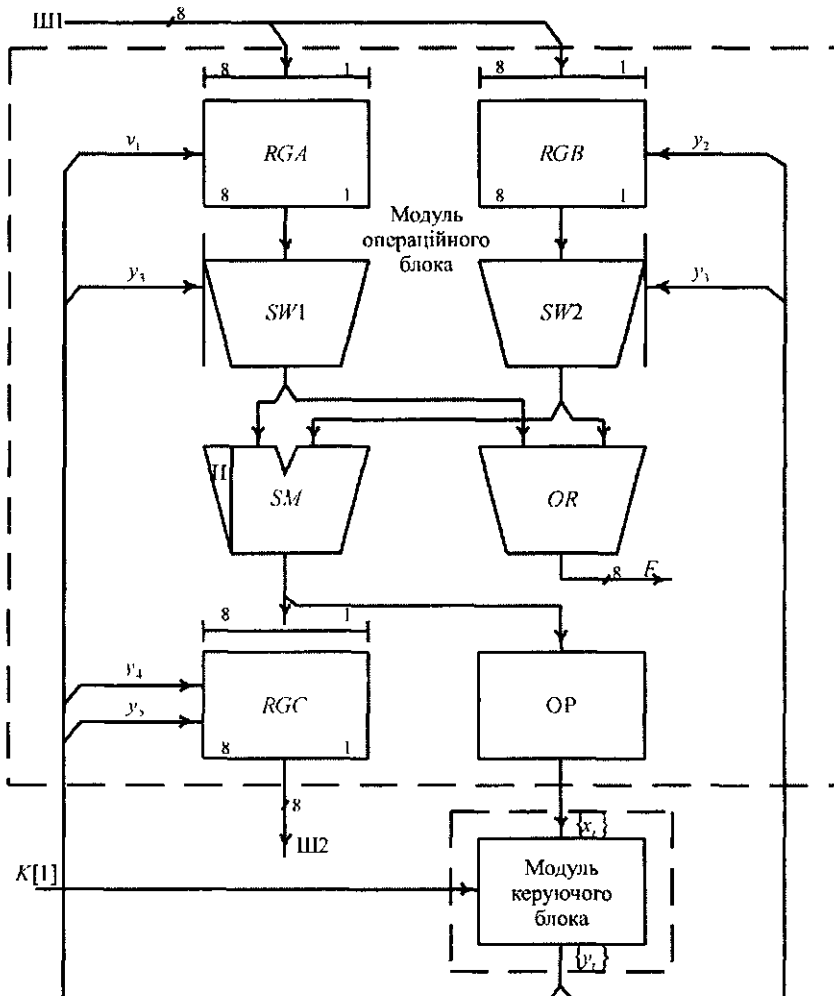


Рис. 9.1. Функціональна схема АЛП для додавання чисел

Схема АЛП містить:

- регістри RGA і RGB для приймання і подальшого зберігання із вхідної шини Ш1 першого і другого операндів;
- паралельний комбінаційний суматор з додатковим старшим розрядом знака П для створення модифікованого доповняльного коду;
- регістр результату RGC , дані з якого пересилаються по вихідній шині Ш2 в оперативну пам'ять;
- схеми електронних ключів $SW1$ і $SW2$;
- схему вироблення ознак переповнення OP ;
- схему диз'юнкторів OR для виконання операцій порозрядного логічного додавання кодів операндів A і B .

Ознаки результату обчислюються за допомогою булевих виразів:

$$\varphi_1 = \overline{\Pi} \cdot \overline{SM}[n] \text{ — додатний результат (знаки 00);}$$

$$\varphi_2 = \Pi \cdot SM[n] \text{ — від'ємний результат (знаки 11);}$$

$$\varphi_3 = \overline{\Pi} \cdot SM[n] \vee \Pi \cdot \overline{SM}[n] \text{ — переповнення розрядної сітки ПП (знаки дорівнюють 01 чи 10);}$$

$$\varphi_4 = \bigwedge_{i=1}^{n+1} \overline{SM}[i] \text{ — нульовий результат.}$$

Після закінчення операції КА аналізує ознаки результату і встановлює значення відповідних тригерів ознак. Ознака переповнення перевіряється до закінчення операції і за її наявності виконання програми переривається.

Ознака OR реалізується за допомогою восьми логічних двохходових елементів ЧИ за співвідношенням

$$F_i = (A_i \vee B_i), \quad i = 1, 2, \dots, 8,$$

де F_i — i -й вихід вузла логічного додавання. Ця операція виконується автоматично незалежно від коду команди.

9.1.5. Мікропрограма додавання

Мікропрограма додавання двійкових чисел у доповняльних кодах має такий вигляд.

Початок. Якщо $K[1]$, то M_1 , інакше — чекати

$M_1 \quad y_1 : RGA := A$ <приймання першого операнда>

$y_2 : RGB := B$ <приймання другого операнда>

$y_3 : SM := A + B$ <додавання>

Якщо φ_3 , то M_2 , інакше

$y_4 : RGC := SM$ <присвоєння результату>

$y_5 : Ш2 := RGC$ <пересилання в пам'ять>

Перейти до M_3

$M_2 \quad y_6 : Tп := ПП$ <тригеру переповнення $Tп$ присвоюється ознака $ПП$ >

M_3 Кінець.

Примітка. $K[1]$ — однорозрядний код команди додавання.

Змістовний і закодований графи мікропрограми додавання показано на рис. 9.2.

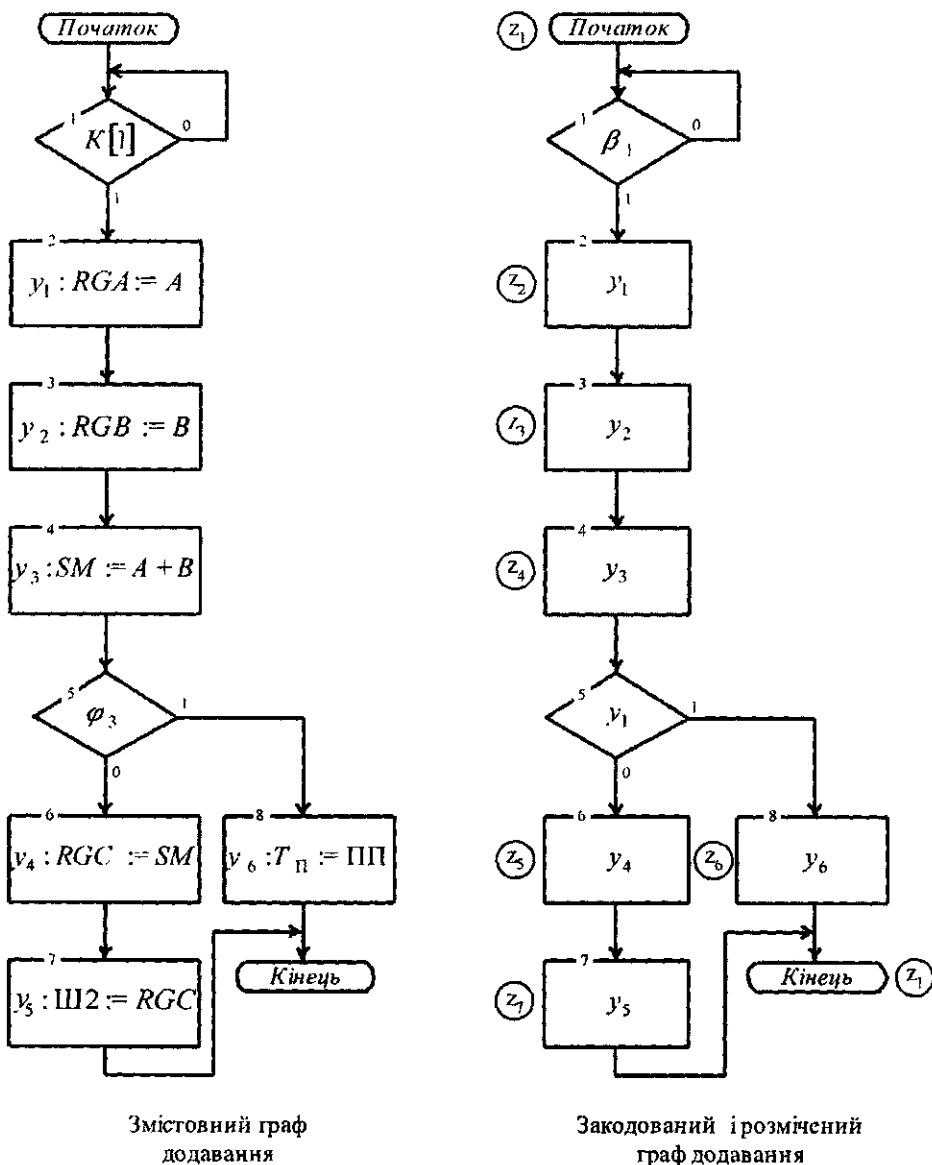


Рис. 9.2. Графи мікропрограми додавання

9.1.6. Принципіальна схема модуля операційного блока

Модуль операційного блока (МОБ) будується на мікросхемах ТТЛШ серії КР1533 за винятком суматора, взятого із серії К1531.

Для побудови принципіальної схеми МОБ використано такі мікросхеми (рис. 9.3):

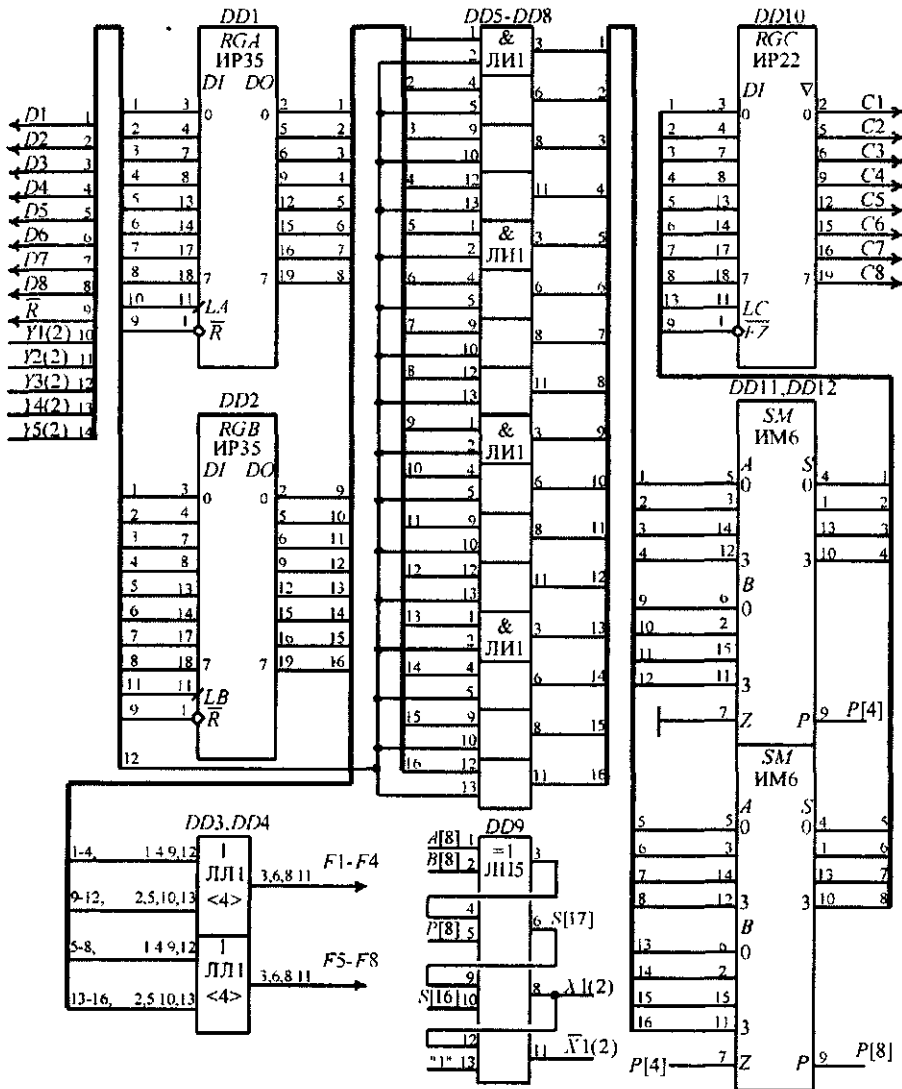


Рис. 9.3. Принципіальна схема МОБ для операції додавання

- два вхідні восьмирозрядні регістри *RGA* і *RGB* типу IP35, позиційне позначення *DD1*, *DD2*;
- дві мікросхеми типу ЛЛ1, кожна з яких містить по чотири двохходових логічних елементи ЧИ, позиційне позначення *DD3*, *DD4*. Їх використовують для реалізації порозрядної диз'юнкції над кодами операндів *A* і *B*. Зображення подано пакетним способом;
- чотири мікросхеми типу ЛЛ1, кожна з яких містить по чотири двохходових кон'юнктори, позиційне позначення *DD5–DD8*. Використовують для підключення виходів регістрів *RGA* і *RGB* до входів суматора *SM*;
- мікросхему типу ЛП5, яка містить чотири логічні елементи "виключальне ЧИ", позиційне позначення *DD9*. Використовують для створення старшого знакового розряду суматора і логічної ознаки $\varphi_3 = x_1$ та \bar{x}_1 ;

- вихідний восьмирозрядний регістр *RGC* типу *IP22* з трьома станами, позиційне позначення *DD10*. Використовують для приймання результату додавання і передавання його на вихідну шину,
- два чотирирозрядні комбінаційні суматори *SM* типу *IM6*, позиційне позначення *DD11*, *DD12*.

9.1.7. Проектування модуля керуючого блока

Проектування модуля керуючого блока (МКБ) на основі автомата Мура з пам'яттю на *JK*-тригерах виконується в такій послідовності.

1. Розмічається закодований граф мікропрограми додавання (див рис 9 2) Визначається максимальна кількість станів автомата Мура, що дорівнює $L = 7$. Для реалізації такого числа станів необхідно використати $n = \lceil \log_2 7 \rceil = 3$ тригери.
2. На основі розміченого графу мікропрограми будується граф автомата Мура (рис. 9 4), який інтерпретує мікропрограму додавання.

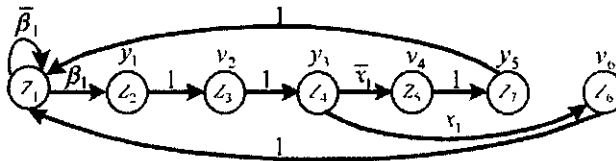


Рис. 9.4. Граф автомата Мура для мікропрограми додавання

3. Стани автомата Мура кодуються значеннями виходів *JK*-тригерів $z_1 = \overline{Q_3}\overline{Q_2}\overline{Q_1}$, $z_2 = \overline{Q_3}\overline{Q_2}Q_1$, $z_3 = \overline{Q_3}Q_2\overline{Q_1}$, ..., $z_7 = Q_3Q_2\overline{Q_1}$
4. На основі графу автомата Мура записується його структурна таблиця переходів (табл. 9 1)

Таблиця 9.1

z_i	$k(z_i)$	z_j	$k(z_j)$	$\{x_i\}$	$\{y_i\}$	<i>JK</i>	
						<i>K</i>	<i>J</i>
z_1	000	z_1	000	$\overline{\beta}_1$	-	-	-
		z_2	001	β_1	y_1	-	J_1
z_2	001	z_3	010	1	y_2	K_1	J_2
z_3	010	z_4	011	1	y_3	-	J_1
z_4	011	z_5	100	\overline{x}_1	y_4	K_2, K_1	J_3
		z_6	101	x_1	y_6	K_2	J_3
z_5	100	z_7	110	1	y_5	-	J_2
z_6	101	z_1	000	1	-	K_3, K_1	-
z_7	110	z_1	000	1	-	K_3, K_2	-

5. На підставі даних табл. 9.1 записуються системи логічних рівнянь для функцій збудження входів *JK*-тригерів і виходів:
 - для функцій збудження входів:

$$\begin{aligned}
 J_1 &= z_1\beta_1 \vee z_3; & J_2 &= z_2 \vee z_5; & J_3 &= z_4; \\
 K_1 &= z_2 \vee z_4\overline{x}_1 \vee z_6; & K_2 &= z_4 \vee z_7; & K_3 &= z_6 \vee z_7.
 \end{aligned}$$

- для вихідних керуючих сигналів:

$$Y_1 = z_2; Y_2 = z_3; Y_3 = z_4; Y_4 = z_5; Y_5 = z_7; Y_6 = z_6.$$

- Будується принципіальна схема модуля керування МКБ (рис. 9.5).

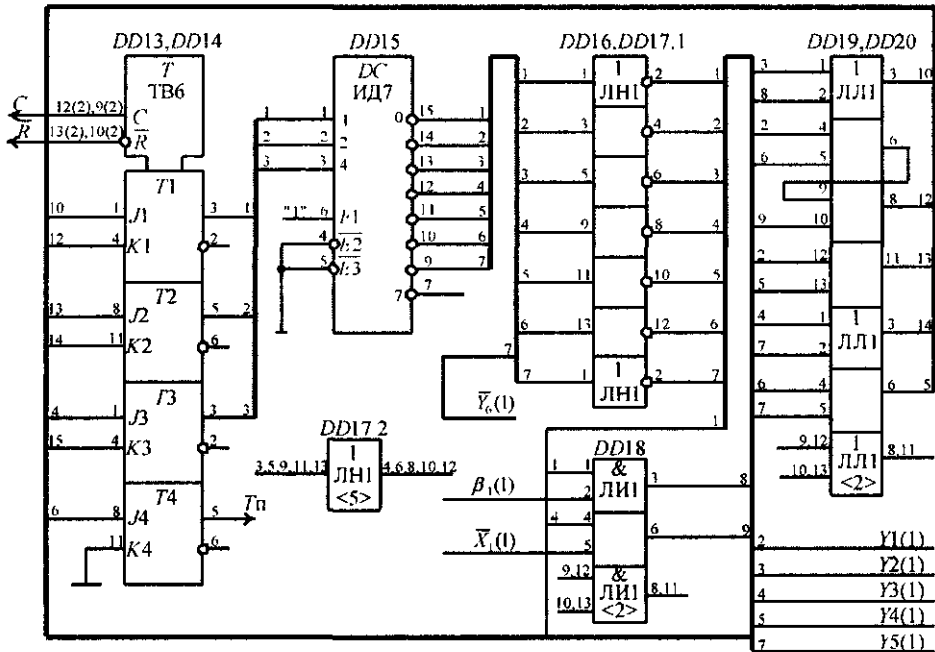


Рис. 9.5. Принципіальна схема МКБ для операції додавання

Відповідність між входами керування мікросхем МОБ і сигналами мікрооперацій наведено в табл. 9.2.

Таблиця 9.2

Входи	LA	LB	ЛИ1	LC	$T_n(J_4)$	\overline{EZ}
Сигнали мікрооперацій	y_1	y_2	y_3	y_4	y_6	$\overline{y_5}$

Для побудови принципіальної схеми модуля МКБ (рис. 9.5) використані такі мікросхеми:

- DD13, DD14 — дві мікросхеми типу ТВ6, кожна з яких містить по два JK-тригери (пам'ять автомата Мура);
- DD15 — дешифратор типу ИД7. Використовується для виділення станів автомата;
- DD16, DD17.1 — дві мікросхеми типу ЛН1, кожна з яких містить шість інверторів;
- DD18 — мікросхему типу ЛН1, яка містить чотири кон'юнктори;
- DD19, DD20 — дві мікросхеми типу ЛЛ1.

Перелік мікросхем, використаних для побудови АЛП, що реалізує операцію додавання, наведено в табл. 9.3.

Позиції	Найменування мікросхеми	Кількість
<i>DD1, DD2</i>	КР1533ИР35	2
<i>DD3, DD4, DD19, DD20</i>	КР1533ЛЛ1	4
<i>DD5 – DD8, DD18</i>	КР1533ЛИ1	5
<i>DD19</i>	КР1533ЛП5	1
<i>DD10</i>	КР1533ИР22	1
<i>DD11, DD12</i>	КР1533ИМ6	2
<i>DD13, DD14</i>	КР1533ТВ6	2
<i>DD15</i>	КР1533ИД7	1
<i>DD16, DD17</i>	КР1533ЛН1	2

9.1.8. Характеристики АЛП

Мікросхеми ТТЛШ серії КР1533, які використано для побудови АЛП, що реалізує операцію додавання, споживають таку потужність:

$$P_{ИР35} = 122 \text{ мВт}; \quad P_{ИР12} = 100 \text{ мВт}; \quad P_{ИМ6} (\text{тип F, серія КР1531}) = 180 \text{ мВт};$$

$$P_{ИД7} = 50 \text{ мВт}; \quad P_{ТВ6} = 22,5 \text{ мВт}; \quad P_{ЛП5} = 30 \text{ мВт};$$

$$P_{ЛН1} = 13 \text{ мВт}; \quad P_{ЛЛ1} = 22,5 \text{ мВт}; \quad P_{ЛИ1} = 16 \text{ мВт}.$$

Потужність, яку споживає даний АЛП, розраховується додаванням потужностей всіх мікросхем:

$$P_{АЛП} = 2P_{ИР35} + P_{ИР12} + 2P_{ИМ6} + P_{ИД7} + 2P_{ТВ6} + P_{ЛП5} + 2P_{ЛН1} + 4P_{ЛЛ1} + 5P_{ЛИ1} =$$

$$= (2 \cdot 122 + 100 + 2 \cdot 180 + 50 + 2 \cdot 22,5 + 30 + 2 \cdot 13 + 4 \cdot 22,5 + 5 \cdot 16) 10^{-3} = 1,125 \text{ Вт}.$$

Мікросхема КР1531ИМ6 реалізує додавання двох чотирирозрядних операндів за $t_{SM1} = 15 \text{ нс}$. При додаванні восьмирозрядних операндів тривалість мікрооперацій подвоюється, тобто $t_{SM2} = 2 \cdot t_{SM1} = 30 \text{ нс}$. Враховуючи затримки сигналів, що подаються на входи суматора, рекомендується обирати тривалість машинного такту $T_C = 2 \cdot t_{SM2} = 60 \text{ нс}$. При цьому швидкодія АЛП, виражена кількістю операцій додавання за секунду типу "регістр-регістр", становитиме $F = 1/T_C \cong 16 \text{ млн оп/с}$.

9.2. Реалізація операції віднімання

9.2.1. Початкові дані

- Тип арифметичної операції — віднімання двійкових чисел;
- початковий код подання операндів — доповняльний;
- розрядність операндів — 8 біт;
- код виконання операції у суматорі — доповняльний модифікований;
- структура операційного блока — із закріпленими мікроопераціями;
- тип керуючого блока — автомат Мілі з пам'яттю на *D*-тригерах;
- схема логічної ознаки переповнення розрядної сітки;
- схема логічного порозрядного множення кодів вхідних операндів *A* і *B*.

Елементна база, перелік обов'язкового графічного матеріалу та необхідні розрахунки — згідно з пунктом 9.1.2.

9.2.2. Алгоритми віднімання двійкових чисел

Алгоритм віднімання двійкових чисел реалізується у такій послідовності:

- у реєстри RGA і RGB із вхідної шини один за одним паралельним кодом записуються відповідні їм операнди A і B ;
- операнд B безумовно інвертується;
- мікрооперація віднімання виконується в доповняльних модифікованих кодах протягом одного машинного такту;
- аналізується результат віднімання. За відсутності переповнення розрядної сітки результат записується у реєстр RGC і потім пересилається в оперативну пам'ять, за наявності переповнення результат не фіксується і в ЦПК передається сигнал переповнення T_n .

9.2.3. Функціональна схема АЛП для виконання операції віднімання

Функціональну схему восьмирозрядного АЛП для виконання мікропрограми віднімання показано на рис. 9.6:

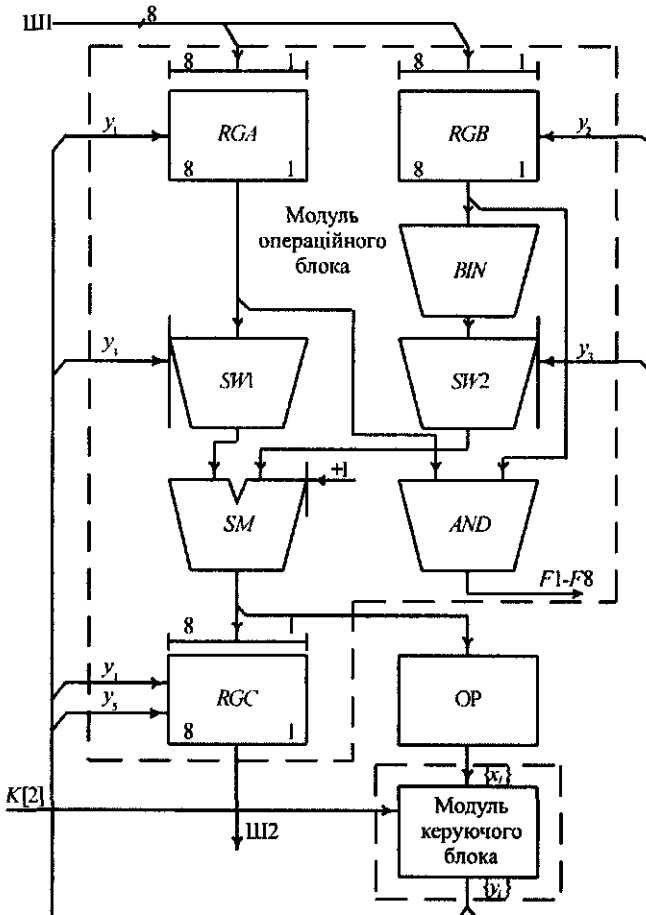


Рис. 9.6. Функціональна схема АЛП для віднімання чисел

Схема АЛП містить:

- регістри RGA і RGB для приймання із вхідної шини III1 операндів A і B та їх зберігання протягом часу виконання мікропрограми;
- схему інвертування BIN змісту регістра RGB ;
- електронні ключі $SW1$ і $SW2$ для комутації операндів;
- схему порозрядного логічного множення AND ;
- комбінаційний суматор SM з додатковим знаковим розрядом Π для створення модифікованого коду. На вхід перенесення першого розряду суматора подається "лог. 1";
- регістр результату RGC ;
- схему ознаки переповнення OP ;
- модуль керуючого блока (МКБ) на основі автомата Мілі з пам'яттю на D -тригерах.

9.2.4. Мікропрограма віднімання

Мікропрограма віднімання двійкових чисел у модифікованих доповняльних кодах має такий вигляд:

Початок. Якщо $K[2]$, то M_1 , інакше — чекати

M_1 $y_1 : RGA := A$ <приймання першого операнда>

$y_2 : RGB := B$ <приймання другого операнда>

$y_3 : SM := A + \bar{B} + 1$ <мікрооперація віднімання в доповняльних кодах>

Якщо φ_3 , то M_2 , інакше

$y_4 : RGC := SM$ <присвоєння результату>

$y_5 : III2 := RGC$ <пересилання в пам'ять>

Перейти до M_3

M_2 $y_6 : T\Pi := III$ <тригеру переповнення $T\Pi$ присвоюється ознака III >

M_3 Кінець.

Примітка. $K[2]$ — однорозрядний код команди віднімання.

Змістовний і закодований графи мікропрограми віднімання показано на рис. 9.7.

9.2.5. Принципіальна схема модуля операційного блока

Модуль операційного блока (МОБ) будується на мікросхемах ТТЛШ серії КР1533 за винятком суматора, взятого із серії КР1531.

Принципіальна схема МОБ містить (рис. 9.8):

- два вхідних восьмирозрядних регістри RGA і RGB типу ІР35, позиційне позначення $DD1, DD2$;
- дві мікросхеми типу ЛН1, позиційне позначення $DD3, DD4.1$. Призначені для інвертування кода операнда B ;
- чотири мікросхеми типу ЛН1, позиційне позначення $DD5-DD8$. Призначені для побудови двох схем електронних ключів $SW1, SW2$;
- мікросхему типу ЛП5, позиційне позначення $DD9$. Використовується для створення старшого знакового розряду суматора і вироблення логічної ознаки переповнення \bar{x}_1 ;

- дві мікросхеми суматорів ІМ6, позиційне позначення $DD10, DD11$;
- вихідний 8-розрядний регістр RGC типу ІР22 з трьома станами, позиційне позначення $DD12$. Використовується для приймання результату віднімання і його передавання на вихідну шину ШІ2.

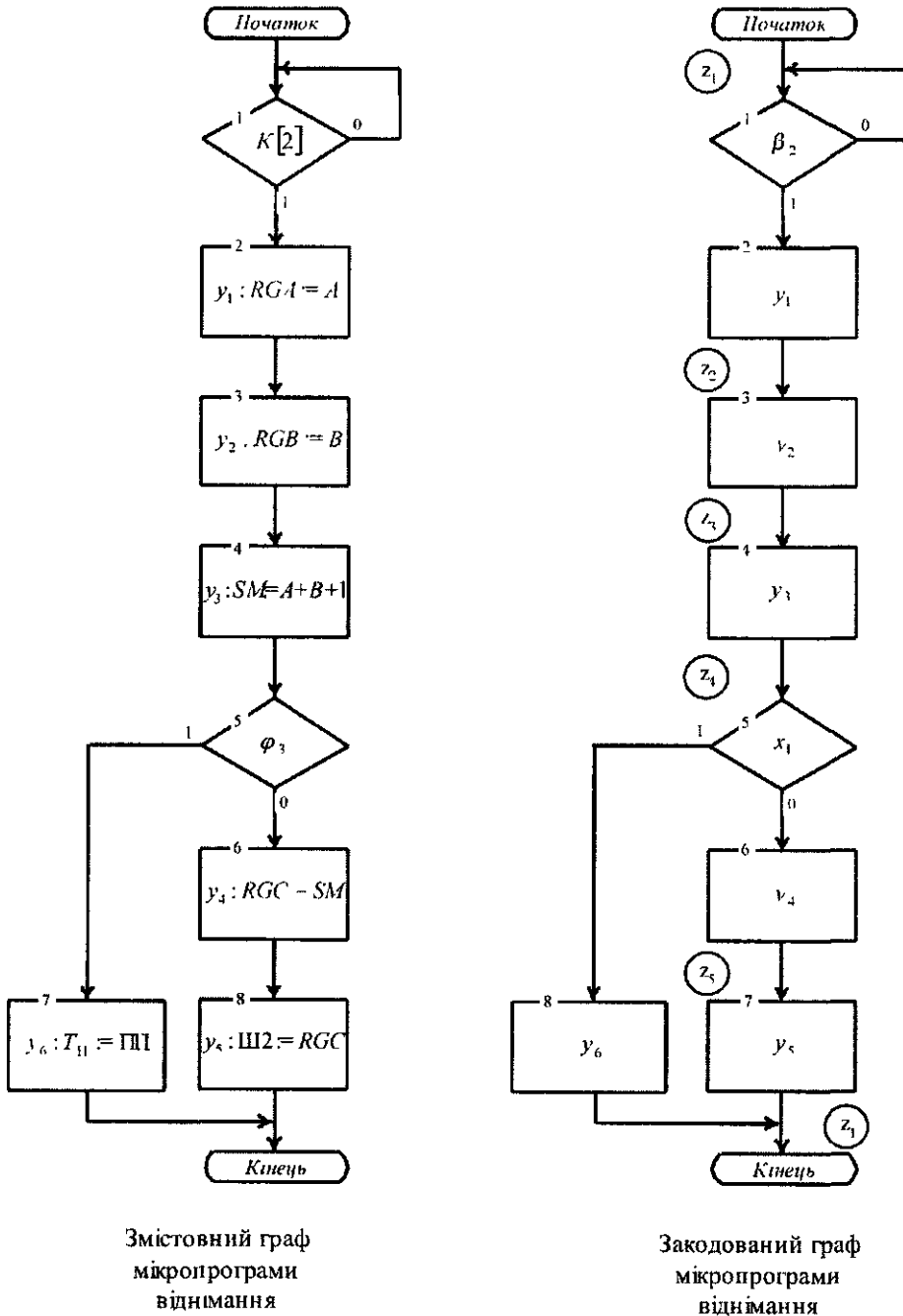


Рис. 9.7. Графи мікропрограми віднімання

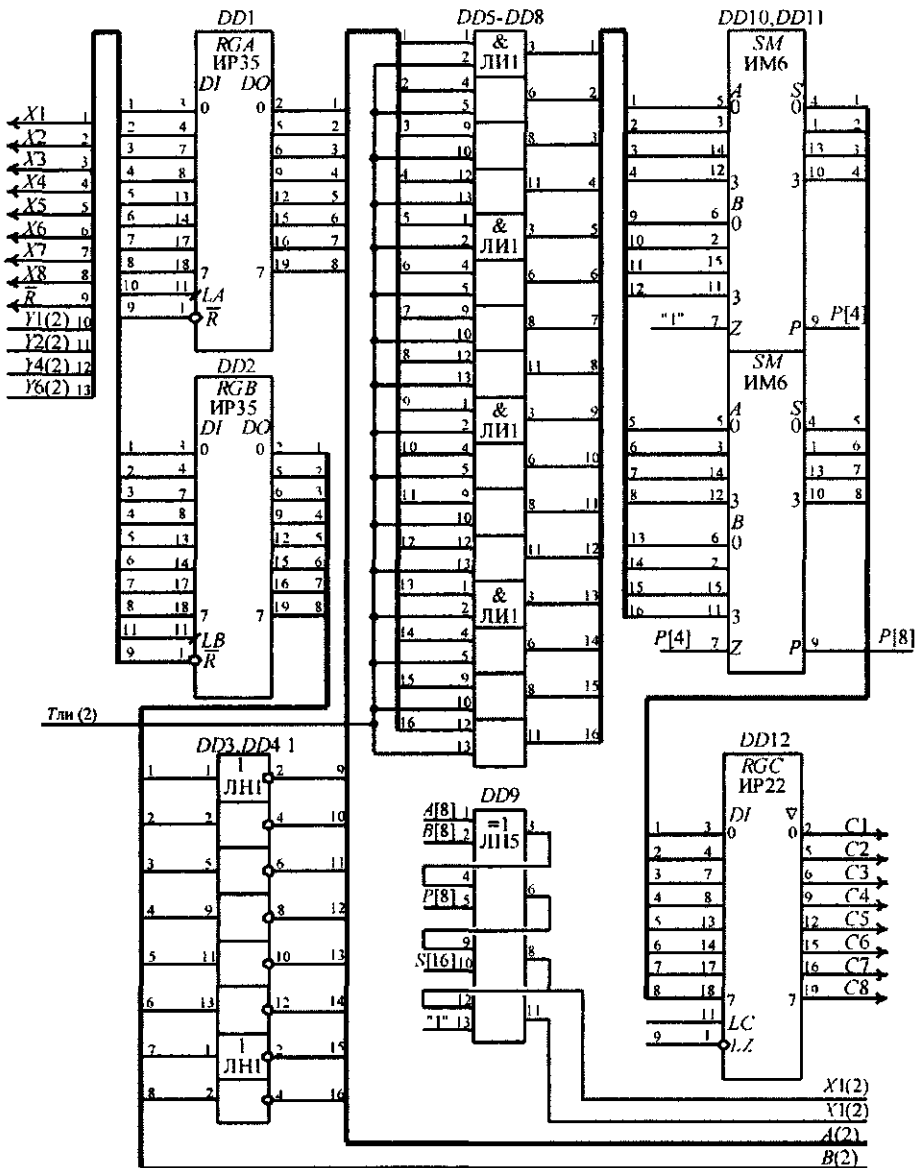


Рис. 9.8. Принципіальна схема МОБ для операції віднімання

9.2.6. Проектування модуля керуючого блока

Проектування МКБ на основі автомата Мілі з пам'яттю на *D*-тригерах виконується в такій послідовності:

1. Розмічається закодований граф мікропрограми віднімання (див. рис. 9.7). Визначається максимальна кількість станів автомата Мілі, яка дорівнює $L = 5$. Для реалізації такої кількості станів необхідно використати $n = \lceil \log_2 5 \rceil = 3$ тригери.

2. На основі розміщеного графу мікропрограми будеться граф автомата Мілі (рис. 9.9), який інтерпретує мікропрограму віднімання.

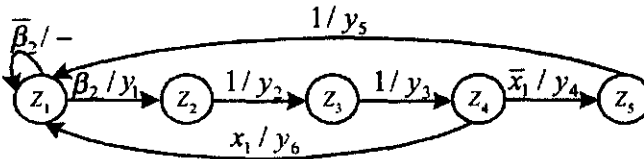


Рис. 9.9. Граф автомата Мілі для мікропрограми віднімання

3. Стани автомата Мілі кодуються значеннями виходів D -тригерів: $z_1 = \overline{Q_3}\overline{Q_2}\overline{Q_1}$, $z_2 = \overline{Q_3}\overline{Q_2}Q_1$, ..., $z_5 = Q_3\overline{Q_2}\overline{Q_1}$.
4. На основі графу автомата Мілі записується його структурна таблиця переходів (табл. 9.4).

Таблиця 9.4

z_i	$k(z_i)$	z_j	$k(z_j)$	$\{x_i\}$	$\{y_i\}$	D -тригери
						D_3, D_2, D_1
z_1	000	z_1	000	$\overline{\beta_1}$	-	-
	000	z_2	001	β_2	y_1	D_1
z_2	001	z_3	010	1	y_2	D_2
z_3	010	z_4	011	1	y_3	D_2, D_1
z_4	011	z_5	100	$\overline{x_1}$	y_4	D_3
		z_1	000	x_1	y_6	-
z_5	100	z_1	000	$\overline{x_1}$	y_5	-

5. На підставі даних табл. 9.4 записуються системи логічних рівнянь:

- для функцій збудження входів:

$$D_1 = z_1\beta_2 \vee z_3 = y_1 \vee z_3;$$

$$D_2 = z_2 \vee z_3;$$

$$D_3 = z_4\overline{x_1} = y_4;$$

- для вихідних сигналів:

$$y_1 = z_1\beta_2; \quad y_2 = z_2; \quad y_3 = z_3; \quad y_4 = z_4\overline{x_1}; \quad y_5 = z_5; \quad y_6 = z_4x_1.$$

6. Будеться принципальна схема модуля керування МКБ (рис. 9.10).

Відповідність між входами керування мікросхем модуля МОБ і сигналами мікрооперацій наведена в табл. 9.5.

Таблиця 9.5

Входи	LA	LB	$T_{ли}$	LC	\overline{EZ}	$T_{п}$
Сигнали мікрооперацій	y_1	y_2	y_3	y_4	$\overline{y_5}$	y_6

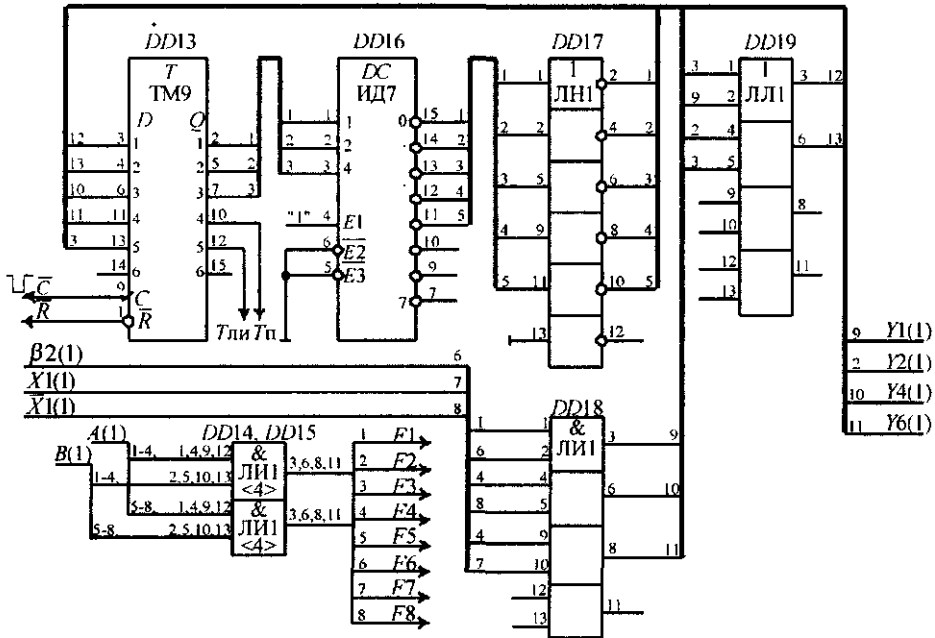


Рис. 9.10. Принципіальна схема МКБ для операції віднімання

Принципіальна схема на рис. 9.10 містить:

- *DD13* — мікросхему типу *TM9*, яка містить шість *D*-тригерів. Три тригери створюють пам'ять автомата Мілі, тригер на виході *Tn* фіксує сигнал переповнення. На виході *Tn* п'ятого тригера формується сигнал керування тривалістю в машинний такт для керування мікросхемами *DD5–DD8*;
- *DD14, DD15* — дві мікросхеми типу *ЛИ1*, які реалізують порозрядну кон'юнкцію кодів операндів *A* і *B* на основі співвідношення $F_i = (A_i \wedge B_i)$, де $i = 1, 2, \dots, 8$;
- *DD16* — мікросхему дешифратора типу *ИД7*;
- *DD17–DD19* — мікросхеми логічних елементів типу *ЛН1, ЛИ1, ЛЛ1*, які реалізують функції комбінаційних схем автомата Мілі.

Розрахунки схеми АЛП виконують аналогічно розділу 9.1.

9.3. Реалізація операцій додавання і віднімання

9.3.1. Початкові дані

- Типи арифметичних операцій — додавання та віднімання двійкових чисел;
- початковий код подання операндів — доповняльний;
- розрядність операндів — 8 біт;
- код виконання операцій у суматорі — доповняльний модифікований;
- структура операційного блока — із закріпленими мікроопераціями;
- тип керуючого блока — автомат Мілі з пам'яттю на *D*-тригерах;

- схема логічної ознаки переповнення розрядної сітки;
- схема логічної порозрядної операції "виключальне ЧИ" кодів початкових операндів A і B .

Елементна база, перелік обов'язкового графічного матеріалу та необхідні розрахунки — згідно з пунктом 9.1.2.

9.3.2. Алгоритми додавання і віднімання двійкових чисел

Алгоритм додавання та віднімання двійкових чисел можна виконувати в обернених або доповняльних кодах. У сучасних комп'ютерах часто операнди зберігаються у пам'яті і оброблюються в доповняльних кодах, перевагу яких описано в розділі 9.1.

Алгоритм додавання (код команди $K[1]$) або віднімання (код команди $K[2]$) виконується у такій послідовності:

- у регістри RGA і RGB із вхідної шини один за одним паралельним кодом записуються відповідні їм операнди A і B ;
- під час операції віднімання (код команди $K[2] = 1$) операнд B безумовно інвертується;
- мікрооперації додавання або віднімання виконуються в доповняльних кодах протягом одного машинного такту;
- аналізується результат операції. За відсутності переповнення розрядної сітки (знаки операндів A і B та результату збігаються) результат записується у регістр RGD і потім пересилається в оперативну пам'ять, за наявності переповнення (знаки операндів A і B та результату на виході суматора не збігаються) результат не фіксується і в ЦПК подається сигнал переповнення ПП.

9.3.3. Функціональна схема АЛП для виконання операцій додавання і віднімання

Функціональну схему восьмирозрядного АЛП для виконання мікропрограми додавання та віднімання подано композицією модуля операційного блока МОБ і модуля керуючого блока МКБ (рис. 9. 11).

Схема містить:

- регістри RGA і RGB для приймання із вхідної шини Ш1 операндів A і B та їх зберігання протягом часу виконання мікропрограми;
- схему інвертування BIN змісту регістра RGB ;
- схему XOR для порозрядного логічного "виключальне ЧИ" кодів операндів A і B ;
- схеми електронних ключів $SW1$ і $SW2$ для комутації операндів;
- комбінаційний суматор SM . На вхід перенесення суматора подається лог. 1 при операції віднімання;
- регістр результату RGD ;
- схему ознаки переповнення OP ;
- модуль керуючого блока МКБ на основі автомата Мілі з пам'яттю на D -тригерах.

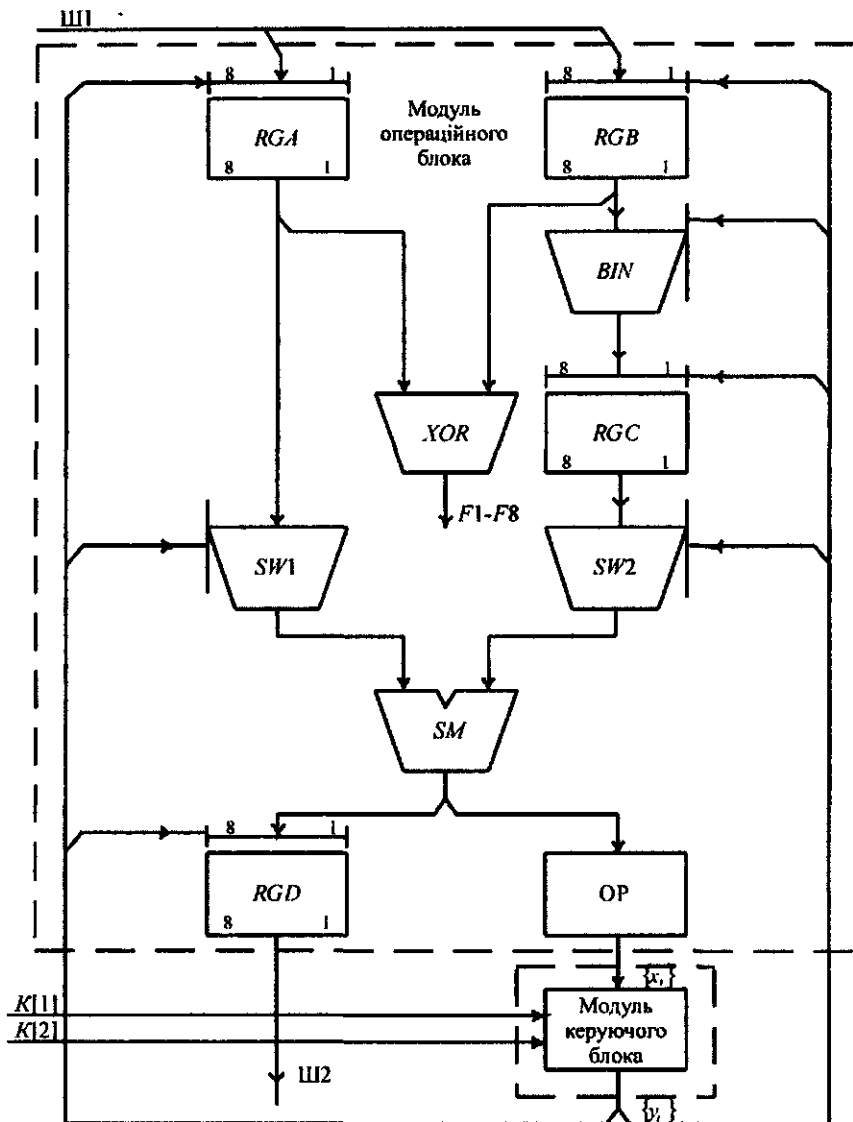


Рис. 9.11. Функціональна схема АЛП для додавання і віднімання чисел

9.3.4. Мікропрограма додавання та віднімання двійкових чисел

Суміщена мікропрограма додавання та віднімання двійкових чисел має вигляд:
Початок. Якщо $K[1]$ або $K[2]$, то M_1 , інакше — чекати

$M_1 \quad y_1 : RGA := A$ <приймання першого операнда>

$y_2 : RGB := B$ <приймання другого операнда>

Якщо $K[1]$, то

$y_3 : RGC := B$, інакше

$y_4 : RGC := \bar{B}$ <пересилання з інвертуванням другого операнда>

Якщо $K[1]$, то

$y_5 : SM := B + C$, інакше $y_6 : SM := B + C + 1$

Якщо φ_3 , то перейти до M_2 , інакше

$y_7 : RGD := SM$ <присвоєння результату>

$y_8 : Ш2 := D$ <пересилання результату в оперативну пам'ять>

Перейти до M_3

M_2 $y_9 : Tп := ПП$ <фіксація переповнення>

M_3 Кінець.

Змістовний і закодований графи мікропрограми додавання і віднімання показані на рис. 9.12.

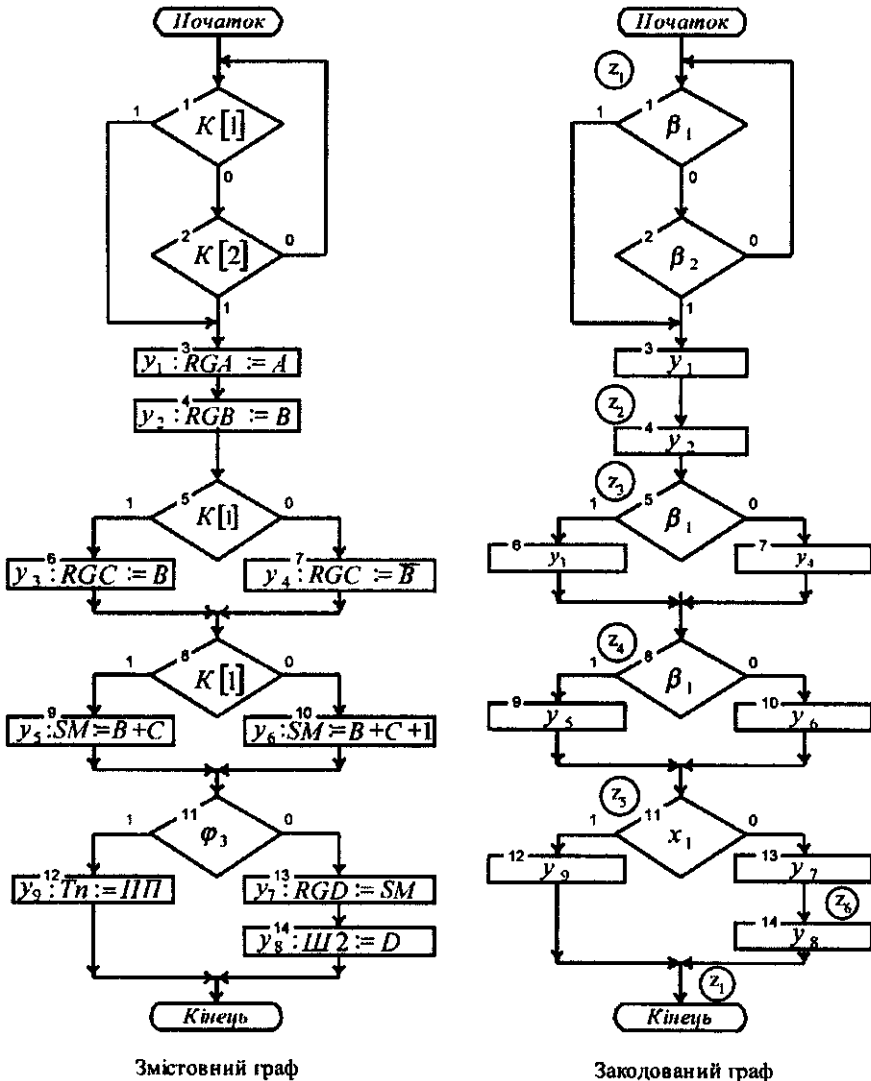


Рис. 9.12. Графи мікропрограми додавання та віднімання

9.3.5. Принципіальна схема модуля операційного блока

Модуль операційного блока (МОБ) будується на мікросхемах ТТЛШ серії КР1533 за винятком суматора, взятого із серії К1531.

Принципіальна схема МОБ містить (рис. 9 13):

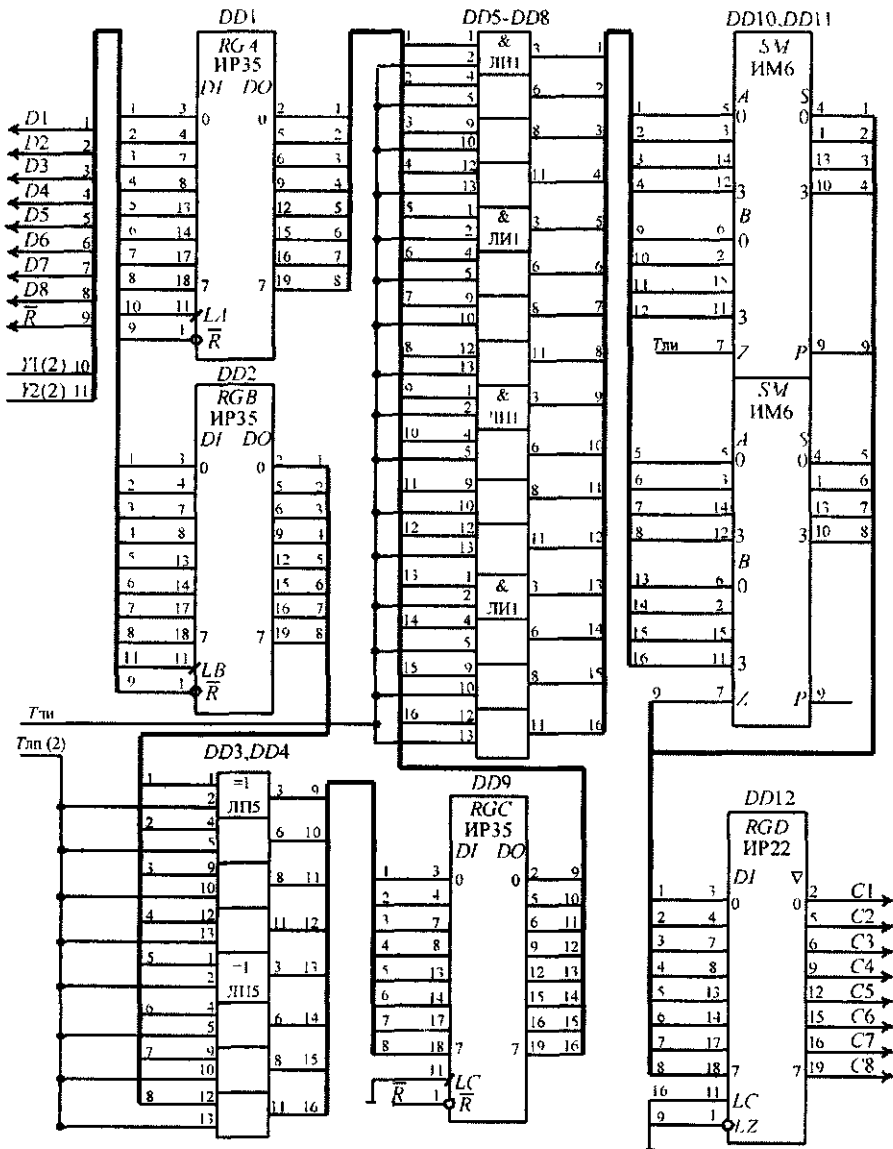


Рис. 9.13. Принципіальна схема МОБ для операцій додавання і віднімання

- DD1, DD2 — два вхідні восьмирозрядні регістри RGA і RGB типу IP35;
- DD3, DD4 — дві мікросхеми типу ЛП5. Призначені для інвертування змісту регістра RGB;
- DD5–DD8 — чотири мікросхеми типу ЛИ1. Реалізують схеми електронних ключів;

- $DD9$ — мікросхему допоміжного регістра RGC типу $IP35$ позиційне позначення
- $DD10$ $DD11$ — дві мікросхеми чотирирозрядних суматорів типу $IM6$
- $DD12$ — мікросхему регістра результату з трьома станами типу $IP22$

9.3.6. Проектування модуля керуючого блока

Проектування модуля МКБ на основі автомата Мілі з пам'яттю на D -тригерах виконується в такій послідовності

1. Розмічається закодований граф мікропрограми додавання і віднімання (рис 9 12) Визначається максимальна кількість станів автомата Мілі яка дорівнює $L = 6$ Для реалізації такої кількості станів необхідно використати $n = \lceil \log_2 6 \rceil = 3$ тригери
2. На основі розміченого графу мікропрограми будується граф автомата Мілі (рис 9 14) який інтерпретує мікропрограму додавання і віднімання

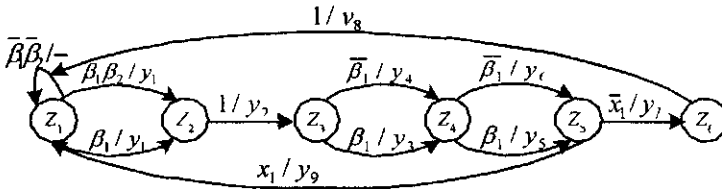


Рис 9 14 Граф автомата Мілі для інтерпретації мікропрограми додавання і віднімання

3. Стани автомата Мілі кодуються значеннями виходів трьох D -тригерів
 $z_1 = \overline{Q_3} \overline{Q_2} \overline{Q_1}$, $z_2 = \overline{Q_3} \overline{Q_2} Q_1$, $z_6 = Q_3 \overline{Q_2} Q_1$
4. На основі графу автомата Мілі записується його структурна таблиця переходів (табл 9 6)

Таблиця 9 6

z_i	$k(z_i)$	z_j	$k(z_j)$	$\{x_i\}$	$\{y_i\}$	D -тригери
						D_3, D_2, D_1
z_1	000	z_1	000	$\overline{\beta_1} \overline{\beta_2}$	-	-
		z_2	001	β_1	y_1	D_1
		z_2	001	$\overline{\beta_1} \beta_2$	y_1	D_1
z_2	001	z_3	010	1	y_2	D_2
z_3	010	z_4	011	β_1	y_3	D_2, D_1
		z_4	011	$\overline{\beta_1}$	y_4	D_2, D_1
z_4	011	z_5	100	β_1	y_5	D_3
		z_5	100	$\overline{\beta_1}$	y_6	D_3
z_5	100	z_6	101	$\overline{x_1}$	y_7	D_3, D_1
		z_1	000	x_1	y_9	-
z_6	101	z_1	000	1	y_8	-

5. На підставі даних табл 9 6 записуються системи логічних рівнянь
 - для вихідних сигналів

$$y_1 = z_1\beta_1 \vee z_1\bar{\beta}_1\beta_2 = z_1\beta_1 \vee z_1\beta_2 \text{ (використовується тотожність)}$$

$$\beta_1 \vee \bar{\beta}_1\beta_2 = \beta_1 \vee \beta_2);$$

$$y_2 = z_2; \quad y_3 = z_3\beta_1; \quad y_4 = z_3\bar{\beta}_1;$$

$$y_5 = z_4\beta_1; \quad y_6 = z_4\bar{\beta}_1; \quad y_7 = z_5\bar{x}_1;$$

$$y_8 = z_6; \quad y_9 = z_5x_1;$$

- для функцій збудження входів *D*-тригерів:

$$D_1 = z_1\beta_1 \vee z_1\bar{\beta}_1\beta_2 \vee z_3 \vee z_5\bar{x}_1 = z_1\beta_1 \vee z_1\beta_2 \vee z_3 \vee z_5\bar{x}_1 = y_1 \vee z_3 \vee y_7 \text{ (враховуються вирази для вихідних сигналів мікрооперацій);}$$

$$D_2 = z_2 \vee z_3;$$

$$D_3 = z_4 \vee z_5\bar{x}_1 = z_4 \vee y_7;$$

- ознака переповнення $\varphi_3 = x_1$ визначається за формулою:

$$\varphi_3 = x_1 = \bar{A}[8]\bar{C}[8]SM[8] \vee A[8]C[8]\bar{SM}[8].$$

6. Будується принципіальна схема МКБ (рис. 9.15).

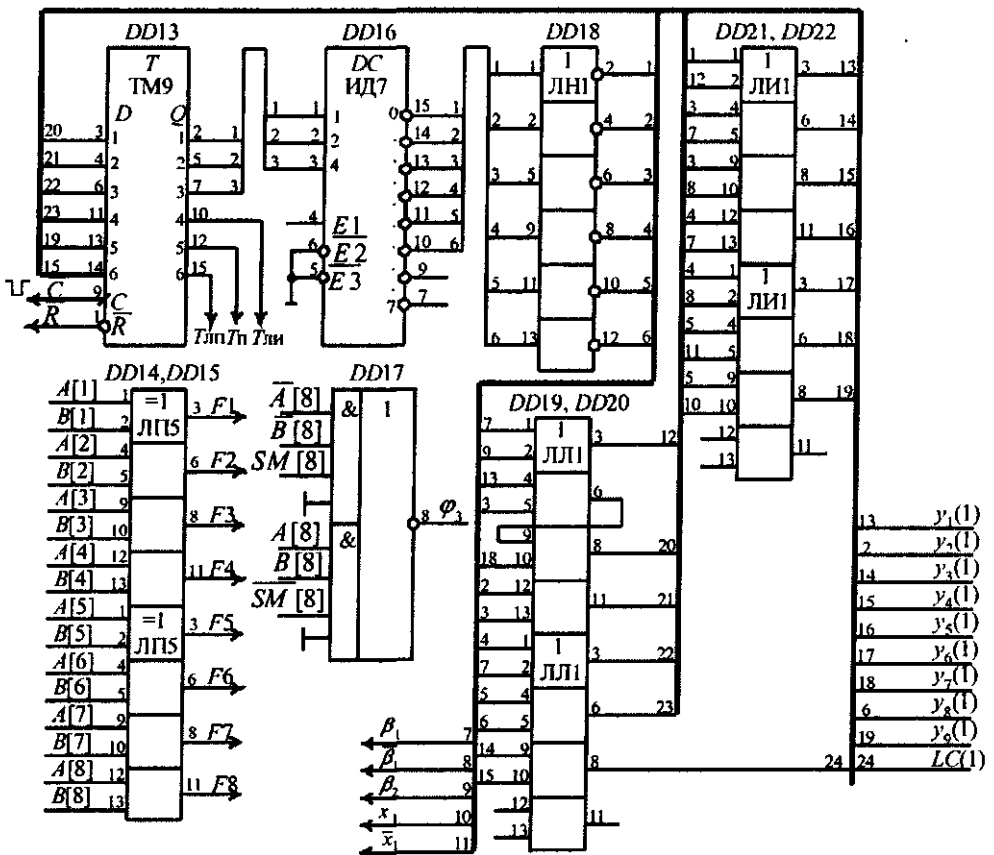


Рис. 9.15. Принципіальна схема МКБ для операцій додавання і віднімання

Відповідність між входами керування мікросхем і сигналами мікрооперацій на-

Таблиця 9.7

Входи	LA	LB	LC	$T_{ли}$	LD	\overline{EZ}	$D4 = T_{ли}$	$T_{п}$
Сигнали мікрооперацій	y_1	y_2	$y_3 \vee y_4$	y_4	y_7	$\overline{y_8}$	$T_{ли} = y_5 \vee y_6$	y_9

Пояснення до табл. 9.7:

- $LA = y_1$; $LB = y_2$ — записування операндів у регістри RGA , RGB ;
- $LC = y_3 \vee y_4$ — записування у регістр RGC , при цьому одночасно формується сигнал $LC = y_3 \vee y_4$ для керування мікросхемами "виключальне ЧИ";
- $T_{ли}$ — сигнал на виході четвертого тригера мікросхеми ТМ9 для керування електронними ключами і входом перенесення Z суматора;
- $LD = y_7$ — записування результату у регістр RGD ;
- $\overline{EZ} = \overline{y_8}$ — пересилання результату в пам'ять;
- $T_{ли} = y_5 \vee y_6 = D4$ — вихід четвертого тригера мікросхеми ТМ9 для керування електронними ключами;
- $T_{п} = y_9$ — вихід п'ятого тригера мікросхеми ТМ9 для фіксації переповнення.

9.4. Реалізація операції множення

9.4.1. Початкові дані

- Тип арифметичної операції — множення двійкових чисел;
- початковий код подання операндів — прямиий;
- розрядність — 8 біт;
- код виконання мікрооперації у суматорі — доповняльний;
- структура операційного блока — із закріпленими мікроопераціями;
- тип керуючого блока — автомат Мілі з пам'яттю на RS -тригерах.

Елементна база, перелік обов'язкового графічного матеріалу та необхідні розрахунки — згідно з пунктом 9.1.2.

9.4.2. Алгоритм множення двійкових чисел із зсувом суми часткових добутоків вправо

Множення двійкових чисел A і B зводиться до обчислення добутку їх модулів та присвоєння йому знака. Добуток двох n -розрядних операндів містить $2n-1$ цифрових розрядів і один знаковий. Якщо перемножуються цілі числа, кома розміщується після молодшого розряду, а якщо дробі — перед старшим розрядом.

1. Множене і множник у прямих кодах послідовно записуються відповідно в регістри A і B . Регістри C і D обнуляються. У лічильник CT записується кількість циклів.
2. Для розрядів множника 1, 2, ..., $n-1$ виконуються такі дії:
 - якщо молодша цифра множника $B[1]=1$, то до суми часткових добутоків додається модуль множеного, інакше — не додається;
 - далі для обох випадків зміст регістрів C і B зсувається вправо на один розряд, причому молодший розряд регістра C передається в старший розряд

регістра B . Після кожного зсуву в молодший розряд регістра B поступає наступний розряд множеного, за яким визначається черговий частковий добуток (нуль або множене);

- після $n-1$ циклів виконується додатковий зсув вправо для передачі в тригер T знака множника і визначається знак добутку додаванням за модулем два знаків множеного і множника.
3. Результат подається конкатенацією чисел C і B ; при цьому в регістрі C розміщуються старші розряди добутку, а в регістрі B — молодші розряди.

9.4.3. Функціональна схема АЛП для операції множення

Функціональна схема АЛП для множення цілих двійкових чисел із зсувом вправо часткових добутків містить (рис. 9.16):

- регістри A і B для приймання з вхідної шини ШІ відповідно множеного та множника;
- паралельний комбінаційний суматор SM ;
- регістр C для приймання часткової суми із SM при одиничному значенні синхросигналу;
- регістр D для приймання і тимчасового зберігання часткової суми з регістра C при спаданні синхросигналу;
- лічильник циклів CT ;
- тригер $T1$ для керування ключами $SW1, SW2$;
- тригер $T2$ для записування знака множника.

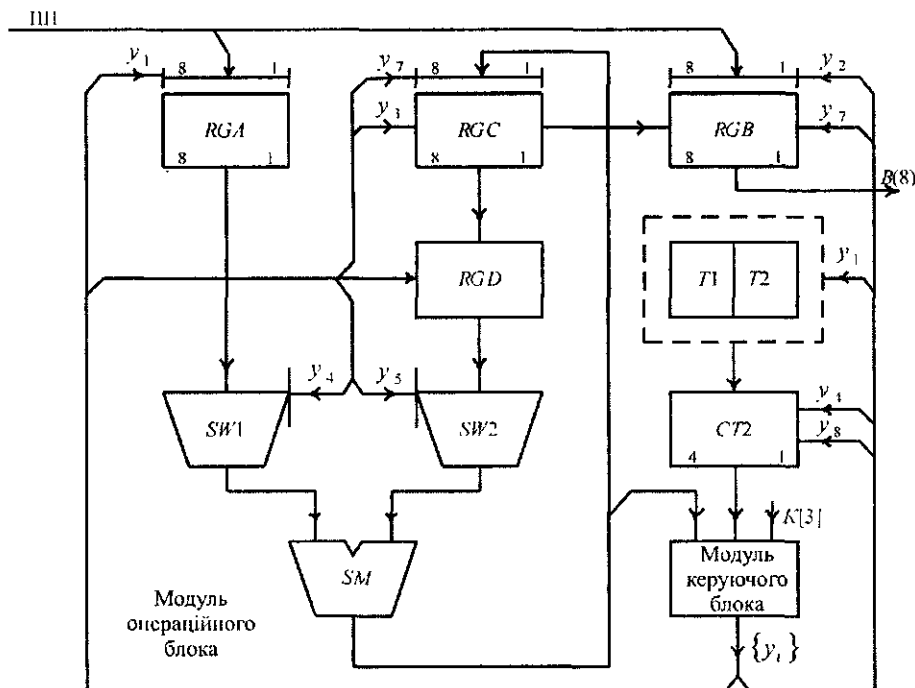


Рис. 9.16. Функціональна схема АЛП для множення

Регістри C і B забезпечують зсув вправо чисел, при цьому значення молодшого розряду регістра $C[1]$ пересилається в старший розряд регістра $B[n]$.

9.4.4. Мікропрограма множення цілих чисел

Початок. Якщо $K[3]$, то M_1 , інакше — чекати

M_1 $y_1 : RGA := A; T2.T1 := 0.0$ <приймання множеного та обнулення тригерів $T1, T2$ >

$y_2 : RGB := B; T2 := B[n]$ <приймання множника та дублювання його знака в тригері $T2$ >

$y_3 : RGC.RGD := 0.0$ <обнулення регістрів>

$y_4 : CT := n - 1$ <запис кількості циклів>

M_2 Якщо $\bar{B}[1]$, то M_3

$y_5 : SM := |A| + D$ <додавання>

$y_6 : RGC := SM$ <пересилання часткової суми>

M_3 $y_7 : RGC.RGB := R(C.B)$ <однорозрядний зсув вправо>

$y_8 : CT := CT - 1$ <декремент лічильника>

Якщо $CT \neq 0$, то M_2 , інакше

$y_7 : RGC.RGB := R(C.B)$

$y_9 : RGC := L(C)$ <однорозрядний зсув вліво>

$y_{10} : RGC := T2.R(C)$ <запис знака результату в регістр RGC >

Кінець.

Змістовний і закодований графі мікропрограми множення показано на рис. 9.17.

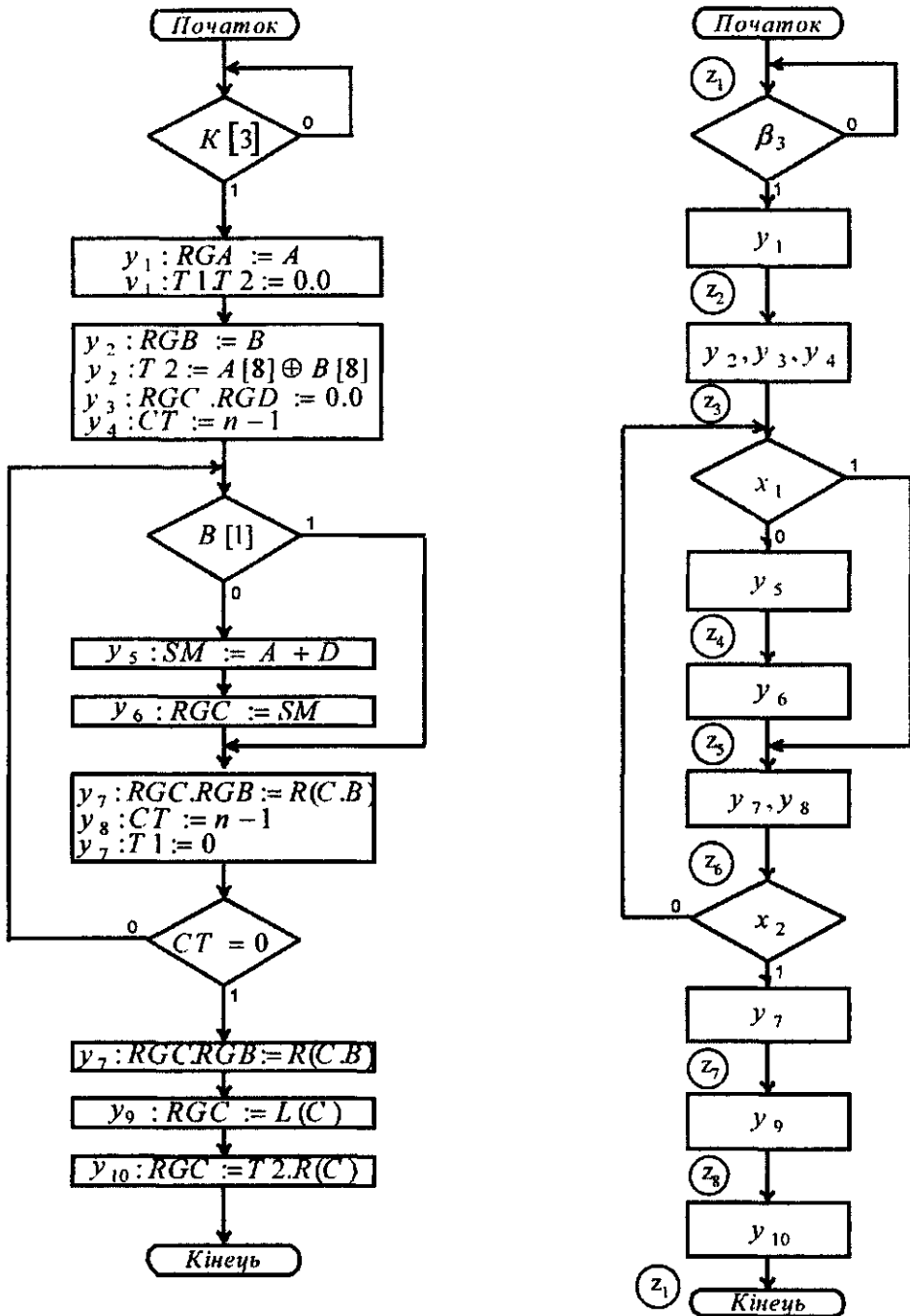


Рис. 9.17. Змістовний і закодований графи мікропрограми множення

9.4.5. Модуль операційного блока для множення цілих чисел

Модуль МОБ для множення цілих чисел містить (рис. 9.18, 9.19):

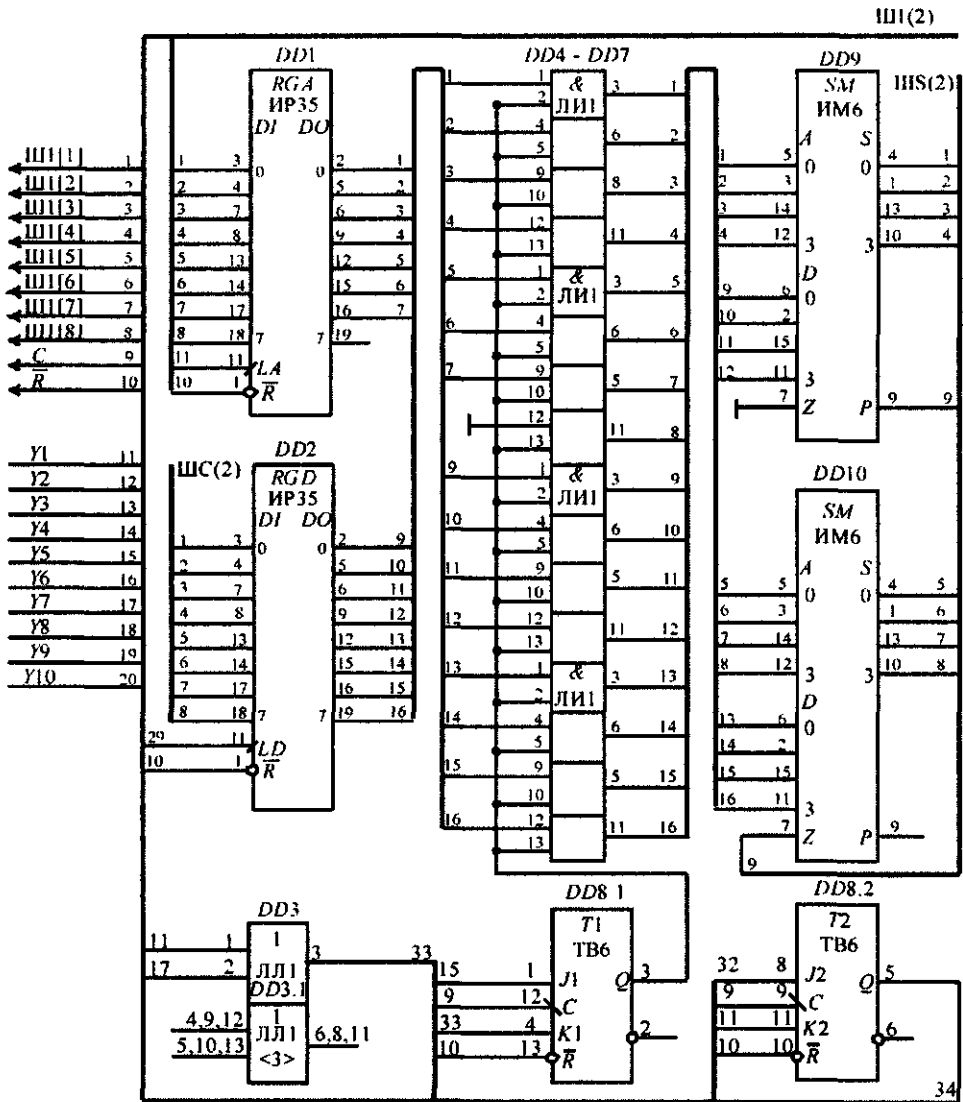


Рис. 9.18. Принципова схема МОБ для операції множення (початок)

- DD1 — реєстр множеного RGA типу IP35;
- DD11 — допоміжний реєстр RGD типу IP35;
- DD12 — реєстр часткової суми RGC типу IP13;
- DD9, DD10 — комбінаційний паралельний суматор SM типу IM6;
- DD13 — лічильник циклів CT2 типу IE10;
- DD4–DD7 — мікросхеми електронних ключів типу ЛІ1;

- допоміжні мікросхеми DD3, DD8, DD14, DD15, DD16 для керування входами функціональних вузлів згідно з логічними рівняннями на основі таблиці відповідності.

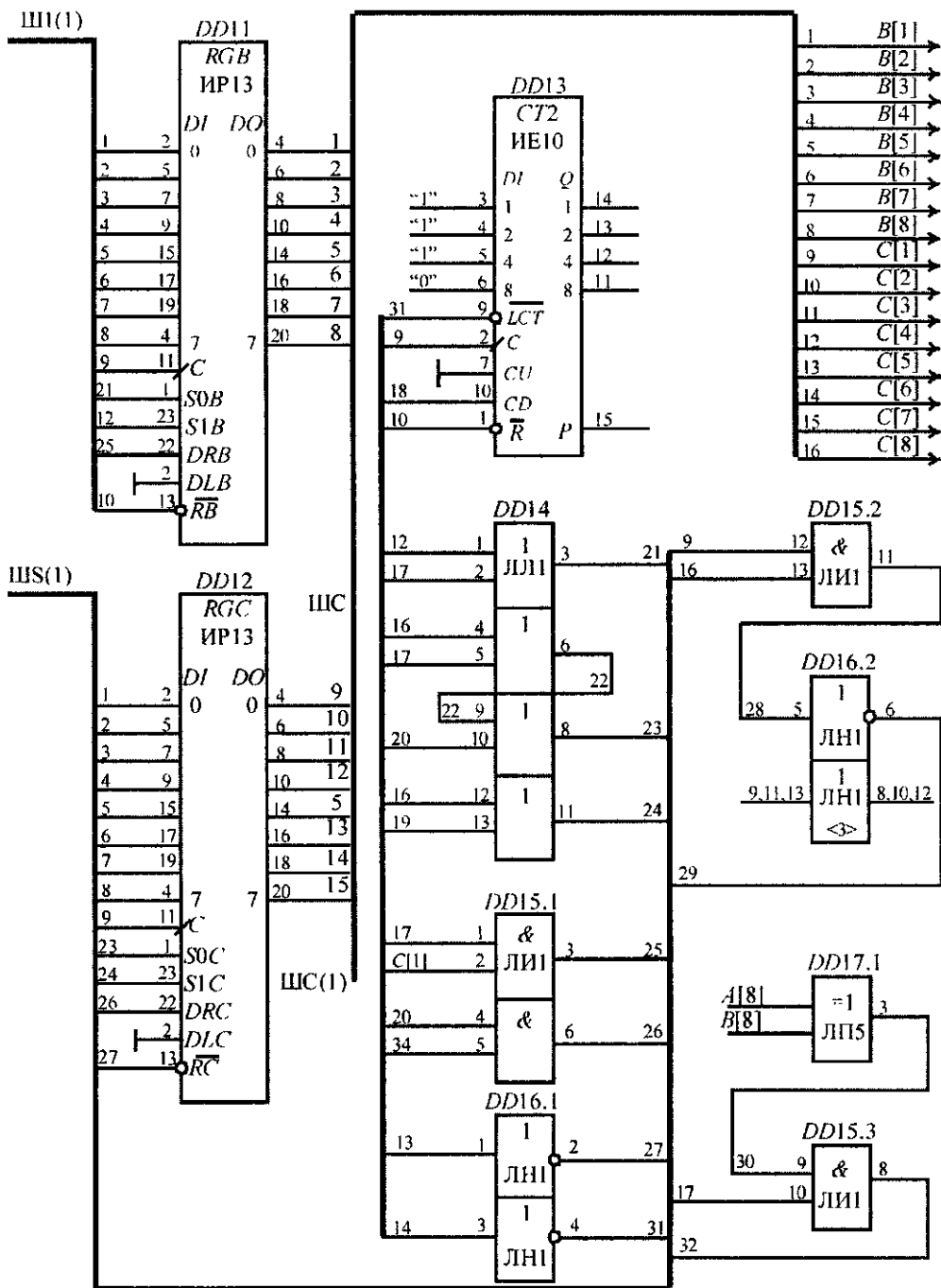


Рис. 9.19. Принципіальна схема МОБ для операції множення (продовження)

9.4.6. Модуль-керуючого блока

Проектування модуля МКБ на основі автомата Мілі з пам'яттю на *RS*-тригерах виконується у такій послідовності.

1. Розмічається закодований граф мікропрограми множення (див. рис. 9.17) і визначається максимальна кількість станів пам'яті та потрібна кількість тригерів: $L = 8, n = \log_2 8 = 3$.
2. Будується граф автомата Мілі, який інтерпретує мікропрограму множення (рис. 9.20). Стани пам'яті кодуються таким чином: $z_1 = \overline{0}_3 \overline{0}_2 \overline{0}_1$, $z_2 = \overline{0}_3 \overline{0}_2 0_1$, $z_3 = 0_3 \overline{0}_2 \overline{0}_1$, ..., $z_6 = 0_3 0_2 0_1$.

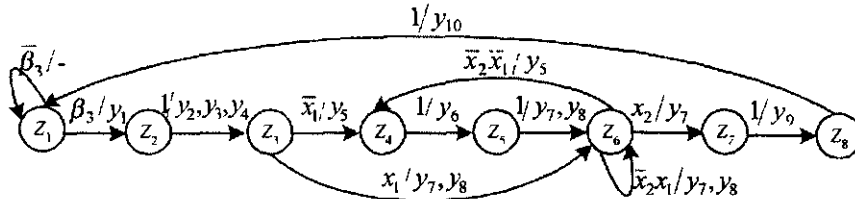


Рис. 9.20. Граф автомата Мілі для мікропрограми множення

3. Записується структурна таблиця переходів і виходів автомата Мілі з пам'яттю на *RS*-тригерах (табл. 9.8). Логічні умови закодовані так: $\beta_3 = K[3]$; $x_1 = \overline{\beta}[1]$; $x_2 = (CT = 0)$.

Таблиця 9.8

z_i	$k(z_j)$	z_j	$k(z_j)$	$x(z_i z_j)$	$y(z_i z_j)$	S	R
z_1	000	z_1	000	$\overline{\beta}_3$	-	-	-
		z_2	001	β_3	y_1	S_1	-
z_2	001	z_3	010	1	y_2, y_3, y_4	S_2	R_1
		z_4	011	\overline{x}_1	y_5	S_1	-
z_3	010	z_6	101	x_1	y_7, y_8	S_3, S_1	R_2
		z_4	011	\overline{x}_1	y_5	S_1	-
z_4	011	z_5	100	1	y_6	S_3	R_2, R_1
z_5	100	z_6	101	1	y_7, y_8	S_1	-
		z_4	011	$\overline{x}_2 \overline{x}_1$	y_5	S_2	R_3
z_6	101	z_7	110	x_2	y_7	S_2	R_1
		z_4	011	$\overline{x}_2 \overline{x}_1$	y_5	S_2	R_3
z_7	110	z_8	111	1	y_9	S_1	-
z_8	111	z_1	000	1	y_{10}	-	R_3, R_2, R_1

4. На підставі даних структурної таблиці записуються системи рівнянь:

- для вихідних сигналів:

$$y_1 = z_1 \beta_3; \quad y_2 = y_3 = y_4 = z_2; \quad y_5 = z_3 \overline{x}_1 \vee z_6 \overline{x}_2 \overline{x}_1; \quad y_6 = z_4;$$

$$y_7 = z_3 x_1 \vee z_5 \vee z_6 \overline{x}_2 x_1 \vee z_6 x_2 = z_3 x_1 \vee z_5 \vee z_6 x_2 \vee z_6 x_1 \quad (\text{використовується тотожність } \overline{x}_2 x_1 \vee x_2 = x_2 \vee x_1);$$

$$y_8 = z_3 x_1 \vee z_5 \vee z_6 \overline{x}_2 x_1;$$

$$y_9 = z_7; \quad y_{10} = z_8.$$

- для функцій збудження входів *RS*-тригерів:

$$S_1 = z_1 \beta_3 \vee z_3 \vee z_5 \vee z_7; \quad R_1 = z_2 \vee z_4 \vee z_6 x_2 \vee z_8;$$

$$S_2 = z_2 \vee z_6 \bar{x}_2 \bar{x}_1 \vee z_6 x_2 = z_2 \vee z_6 \bar{x}_1 \vee z_6 x_2 \quad (\text{використовується тотожність} \\ \bar{x}_2 \bar{x}_1 \vee x_2 = x_2 \vee \bar{x}_1);$$

$$R_2 = z_3 x_1 \vee z_4 \vee z_8;$$

$$S_3 = z_3 x_1 \vee z_4; \quad R_3 = z_6 \bar{x}_2 \bar{x}_1 \vee z_8.$$

У записаних системах рівнянь є спільні двобуквені логічні вирази, для яких вводять позначення:

$$\alpha_1 = z_1 \beta_3; \quad \alpha_2 = z_6 x_2; \quad \alpha_3 = z_6 \bar{x}_1; \quad \alpha_4 = z_3 x_1;$$

$$\alpha_5 = z_6 \bar{x}_2 \bar{x}_1; \quad \alpha_6 = z_3 \bar{x}_1; \quad \alpha_7 = z_5 \bar{x}_1.$$

З урахуванням допоміжних функцій $\alpha_1 - \alpha_7$ та додаткових перетворень за правилом подвійної інверсії і законом де Моргана системи рівнянь подаються у вигляді:

- для функцій збудження входів *RS*-тригерів:

$$S_1 = \overline{\alpha_1 \vee z_3 \vee z_5 \vee z_7} = \overline{\alpha_1} \cdot \overline{z_3} \cdot \overline{z_5} \cdot \overline{z_7};$$

$$R_1 = \overline{z_2 \vee z_4 \vee \alpha_2 \vee z_8} = \overline{z_2} \cdot \overline{z_4} \cdot \overline{\alpha_2} \cdot \overline{z_8};$$

$$S_2 = \overline{z_2 \vee \alpha_2 \vee \alpha_3} = \overline{z_2} \cdot \overline{\alpha_2} \cdot \overline{\alpha_3};$$

$$R_2 = \overline{\alpha_4 \vee z_4 \vee z_8} = \overline{\alpha_4} \cdot \overline{z_4} \cdot \overline{z_8};$$

$$S_3 = \overline{\alpha_4 \vee z_4} = \overline{\alpha_4} \cdot \overline{z_4};$$

$$R_3 = \overline{\alpha_7 \vee z_8} = \overline{\alpha_7} \cdot \overline{z_8};$$

- для вихідних сигналів мікрооперацій:

$$y_1 = \alpha_1; \quad y_2 = y_3 = y_4 = z_2; \quad y_5 = \overline{\alpha_6 \vee \alpha_5} = \overline{\alpha_6} \cdot \overline{\alpha_5}; \quad y_6 = z_4;$$

$$y_7 = \overline{\alpha_4 \vee \alpha_5 \vee \alpha_2} = \overline{\alpha_4} \cdot \overline{\alpha_5} \cdot \overline{\alpha_2};$$

$$y_8 = \overline{\alpha_4 \vee z_5 \vee \alpha_5} = \overline{\alpha_4} \cdot \overline{z_5} \cdot \overline{\alpha_5};$$

$$y_9 = z_7; \quad y_{10} = z_8.$$

9.4.7. Принципіальна схема модуля керуючого блока

Принципіальна схема модуля МКБ містить (рис. 9.21):

- *DD22* — мікросхему чотирьох асинхронних *RS*-тригерів типу TP2 з інверсними входами (пам'ять автомата);
- *DD18*, *DD19.1* — мікросхему HE I типу ЛАЗ, які забезпечують синхронне пряме керування *RS*-тригерами;
- *DD20* — дешифратор типу ИД7, призначений для розшифрування двійкового позиційного коду станів пам'яті;
- *DD23*, *DD24* — дві мікросхеми інверторів типу ЛН1, призначені для інвертування інверсних сигналів з виходів дешифратора;
- *DD19.2*, *DD21* — мікросхеми елементів HE I типу ЛАЗ, призначені для формування допоміжних функцій $\bar{\alpha}_1 - \bar{\alpha}_7$;

- DD25, DD26 – мікросхеми елементів НЕ І типу ЛА1 і ЛА4, які формують сигнали для входів RS-тригерів і вихідні сигнали мікрооперацій.

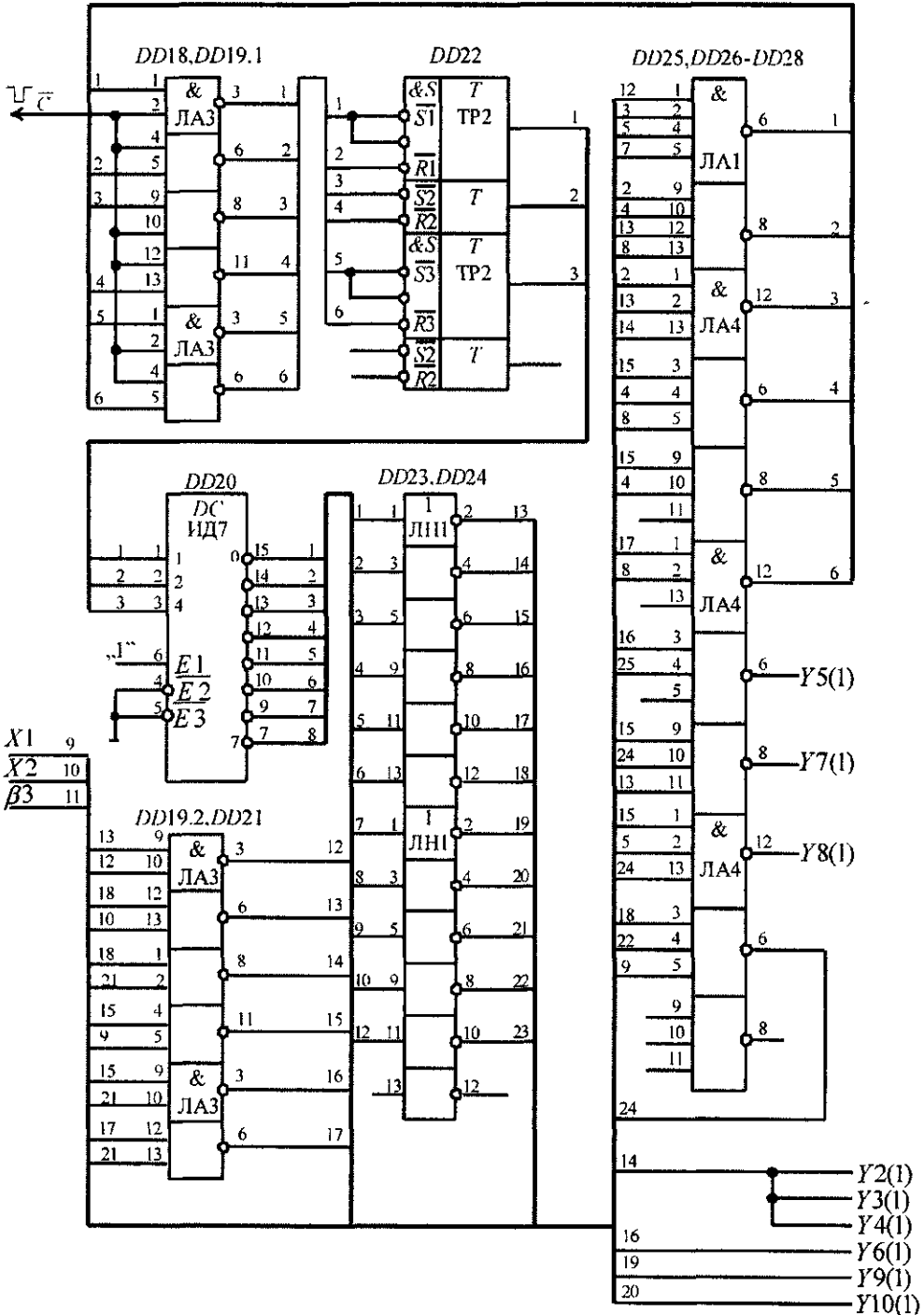


Рис. 9.21. Принципіальна схема модуля МКБ для операції множення

Співвідношення між множиною вихідних сигналів мікрооперацій і керуючими входами мікросхем операційного блока задані табл. 9.9.

Таблиця 9.9

y_i	LA	S0B	S1B	S0C	S1C	LD	\overline{LCT}	CD	J_2	K_2	J_1	K_1	\overline{RC}	\overline{RB}	RGC[8]	RGB[8]
y_1	y_1									y_1		y_1				
y_2		y_2	y_2						F_1							
y_3													$\overline{y_3}$	$\overline{y_3}$		
y_4							$\overline{y_4}$									
y_5											y_5					
y_6				y_6	y_6	$\overline{y_6C}$										
y_7		y_7		y_7								y_7				F_2
y_8								y_8								
y_9					y_9											
y_{10}				y_{10}											$y_{10} \cdot T2$	

Примітка. 1. $F_1 = y_2(A[8] \oplus B[8])$.

2. $F_2 = y_7RGC[1]$.

На підставі даних табл. 9.9 запишемо систему рівнянь, що є основою для побудови другого рівня керування операційним блоком:

$$LA = y_1; \quad S0B = y_2 \vee y_7; \quad S1B = y_2;$$

$$S0C = y_6 \vee y_7 \vee y_{10}; \quad S1C = y_6 \vee y_9;$$

$$LD = \overline{y_6C}; \quad \overline{LCT} = \overline{y_4}; \quad CD = y_8; \quad J_2 = y_2(A[8] \oplus B[8]);$$

$$K_2 = y_1; \quad J_1 = y_3; \quad K_1 = y_1 \vee y_7; \quad \overline{RC} = \overline{y_3}; \quad \overline{RD} = \overline{y_3};$$

$$RGB[8] = y_7 \cdot RGC[1]; \quad RGC[8] = y_{10} \cdot T2.$$

Для побудови другого рівня керування операційним блоком на основі одержаних рівнянь використовуються мікросхеми DD3, DD8, DD14–DD17 (див. рис. 9.19, 9.20).

9.5. Реалізація операції ділення

9.5.1. Початкові дані

- Тип арифметичної операції — ділення двійкових чисел;
- початковий код подання операндів — прямий;
- розрядність — ділене X — 16 біт, дільник Y — 8 біт;
- код виконання мікрооперації у суматорі — доповняльний;
- структура операційного блока — із закріпленими мікроопераціями;
- тип керуючого блока — автомат з пам'яттю на JK-тригерах.

Елементна база, перелік обов'язкового графічного матеріалу та необхідні розрахунки — згідно з пунктом 9.1.2.

9.5.2. Алгоритм ділення цілих чисел

Операція ділення цілих чисел $Z=X/Y$ зводиться до послідовності віднімання дільника Y спочатку від діленого X , а потім від створюваних в процесі ділення залишків R_i .

Залежно від способу віднімання дільника Y розрізняють два основні алгоритми ділення: без відновлення залишку R_i та з відновленням залишку. Обидва способи реалізуються приблизно однаковими апаратними затратами, але для ділення без відновлення залишку потрібно більше мікрооперацій додавання і віднімання. В універсальних комп'ютерах зазвичай використовують ділення без відновлення залишку.

У разі ділення цілих чисел часто ділене X подається в $2n$ -розрядному форматі, а дільник Y — у n -розрядному.

Алгоритм ділення цілих чисел у прямому коді без відновлення залишку реалізується в такій послідовності.

1. У регістри A , B і C послідовно записуються із вхідної шини n -розрядний дільник Y та $2n$ -розрядне ділене X . У лічильник циклів CT заноситься число циклів $n-1$.
2. Ділене X та дільник Y аналізуються на рівність нулю. Якщо ділене $X = 0$, то частці Z присвоюється нульове значення і ділення закінчується. Якщо дільник $Y = 0$, то ділення переривається.
3. Установлюється можливість ділення без переповнення розрядної сітки. Для цього значення діленого подвоюється зсувом вліво на один розряд. Із зсунутого діленого віднімається дільник. Операція віднімання дільника замінюється на його додавання у доповняльному коді і визначається перший залишок R_0 за формулою

$$R_0 = 2|X| - Y = 2|X| + |-Y|_d.$$

Якщо $R_0 < 0$, то ділення можливе; якщо $R_0 \geq 0$, виникає переповнення розрядної сітки і ділення припиняється.

4. Якщо ділення можливе, виконуються такі основні дії:
 - частковий залишок в регістрі B і зміст регістра C зсуваються вліво на один розряд (тобто подвоюються);
 - із зсунутого залишку віднімається дільник, якщо попередній залишок $R_{i-1} \geq 0$, або додається, якщо $R_{i-1} < 0$. Це визначається рекурентним співвідношенням

$$R_i = \begin{cases} 2R_{i-1} + |Y|, & \text{якщо } R_{i-1} < 0; \\ 2R_{i-1} + |-Y|_d, & \text{якщо } R_{i-1} \geq 0, \end{cases}$$

де $i = 1, 2, \dots, n-1$.

- якщо відбувається зсув вліво, в молодший розряд регістра C записується цифра частки r_i згідно із співвідношенням

$$r_i = \begin{cases} 0, & \text{якщо } R_{i-1} < 0; \\ 1, & \text{якщо } R_{i-1} \geq 0. \end{cases}$$

Це означає, що поточна цифра частки є інверсією знака залишку;

- зміст лічильника циклів CT зменшується на одиницю;

- п. 4 повторюється до обнуління лічильника циклів CT .

Значення першого (старшого) розряду частки відводиться для записування знака результату на підставі виразу $r_0 = A[n] \oplus B[n]$.

5. Залишок ділення розміщується у регістрі B на місці старших розрядів діленого, а частка — в регістрі C . Дільник, залишок і частка мають формат n -розрядного числа із знаком. Залишок має мати той же знак, що і ділене, нульові залишки і частки завжди додатні. Якщо знак останнього залишку від'ємний, то він коректується додаванням до нього модуля дільника, після цього залишку присвоюється знак діленого.

9.5.3. Функціональна схема АЛП для мікропрограми ділення

Функціональну схему АЛП для виконання мікропрограми ділення показано на рис. 9.22.

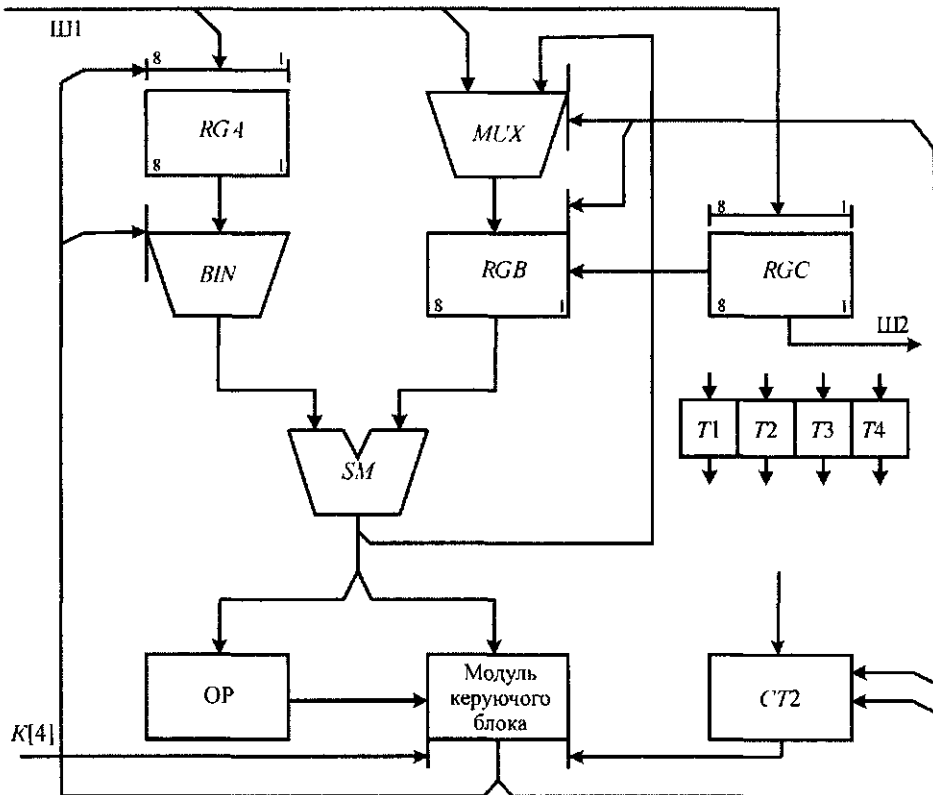


Рис. 9.22. Функціональна схема АЛП для операції ділення

Показана схема АЛП містить:

- регістр RGA для приймання восьмирозрядного дільника Y ;
- регістри RGB і RGC для приймання 16-розрядного діленого X ;
- комбінаційний восьмирозрядний суматор SM ;
- мультиплексор MUX , який забезпечує записування інформації в регістр RGB із вхідної шини Ш1 або з виходів суматора SM ;

- схему *BIN* для подання прямого або оберненого коду дільника на входи суматора *SM*;
- лічильник кількості циклів *CT*;
- схему вироблення ознак результату *OP*;
- тригери *T1–T4* для створення другого рівня керування модулем операційного блока *МОБ5*;
- місцевий керуючий блок *МКБ5* на основі автомата Мілі з пам'яттю на *JK*-тригерах.

9.5.4. Мікропрограма ділення цілих чисел без відновлення залишку

Початок. Якщо $K[4]$, то M_1 , інакше — чекати

M_1 $y_1 : RGA := Y; T1.T2.T3.T4 := 0.0.0.0$ <завантаження дільника і обнулення тригерів>

$y_2 : RGB := X1; T1 := B[n]$ <завантаження старшої частини діленого; присвоєння тригеру $T1$ знака діленого>

$y_3 : RGC := X2$ <завантаження молодшої частини діленого>

$y_4 : RGB.RGC := L(B.C).T3$ <зсув вліво на один розряд змісту регістрів RGB і RGC . Передача значення старшого розряду регістра RGC в молодший розряд регістра RGB , а в молодший розряд регістра RGC — змісту тригера $T3$ >

$y_5 : T3 := ЗНАК = A[n] \oplus T1$ <записування в тригер $T3$ знака результату ділення>

$y_6 : CT := n - 1$ <завантаження в лічильник CT кількості циклів>

$y_7 : SM := B + [-A]_д$ <віднімання дільника в доповняльному коді від значення даних в регістрі RGB >

$y_8 : ADR.RGB := 1.SM$ <завантаження результату віднімання в регістр RGB . Входу адресації ADR мультіплектора MUX надається значення одиниці>

$y_9 : T4 := SM[n]$ <присвоєння знака залишку тригеру $T4$ >

Якщо $T4$, то M_2 , інакше

$y_{10} : T_{\Pi} := \text{III}$ <присвоєння ознаки переповнення тригеру T_{Π} в ЦПК>.

Перейти до M_7 (кінець)

M_2 $y_4 : RGB.RGC := L(B.C).T3$.

M_3 Якщо $T4$, то M_4 , інакше

$y_7 : SM := B + [-A]_д$

$y_{12} : T3.T4 := 0.0$ <обнулення тригерів $T3, T4$ > Перейти до M_5

M_4 $y_{11} : SM := B + |A|$ <додавання дільника до залишку>

$y_{12} : T3.T4 := 0.0$

M_5 $y_8 : ADR.RGB := 1.SM$

$$y_9 : T4 := SM[n]$$

$y_{13} : T3 := \overline{SM}[n]$ <присвоєння тригеру $T3$ інверсного значення знака суматора — формування цифри частки>

$$y_{14} : CT := CT - 1 \text{ <декремент змісту лічильника>}$$

Якщо $CT \neq 0$, то M_5 , інакше M_5

M_5 $y_{15} := RGC := L(C)T3$ <зсув вліво змісту регістра RGC і записування останньої молодшої цифри частки>

Якщо $B[n]$, то M_6 , інакше

$$y_{11} : SM := B + |A|$$

$$y_8 : ADR.RGB := 1.SM$$

M_6 Якщо $B[n] = T1$, то M_7 . Інакше

$$y_{15} := RGB := L(B)0$$
 <зсув вліво залишку>

$$y_{16} : RGB := T1.R(B)$$
 <зсув вправо частки і присвоєння їй знака діленого>

M_7 Кінець.

Змістовний і закодований графі мікропрограми ділення показано на рис. 9.23 і 9.24.

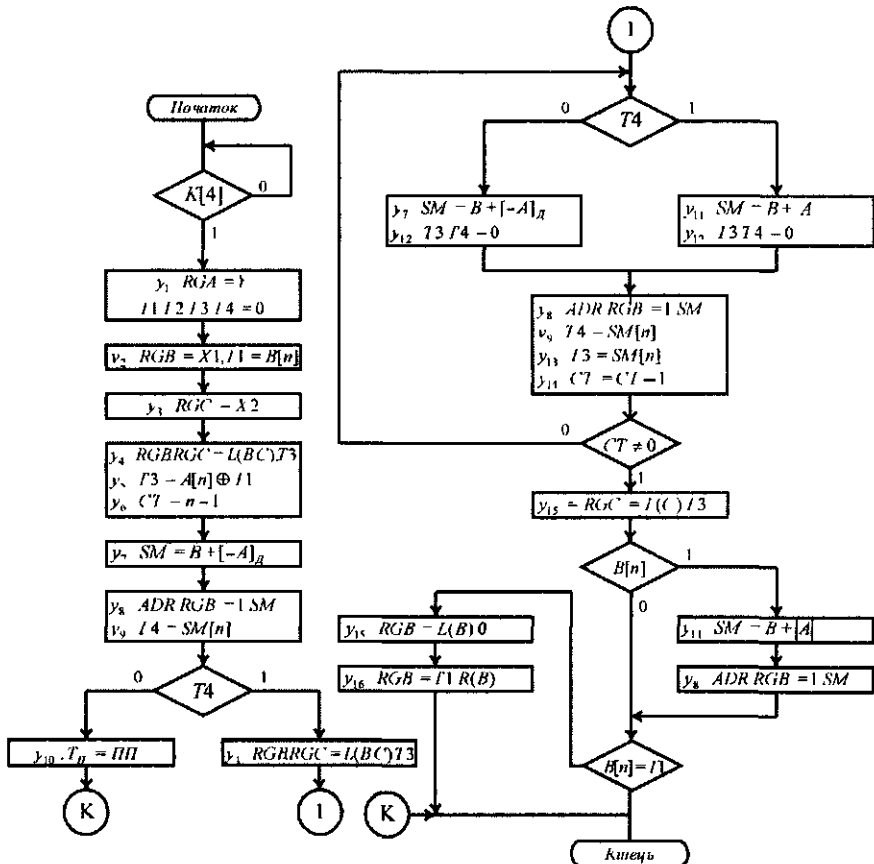


Рис. 9.23. Змістовний графі мікропрограми ділення

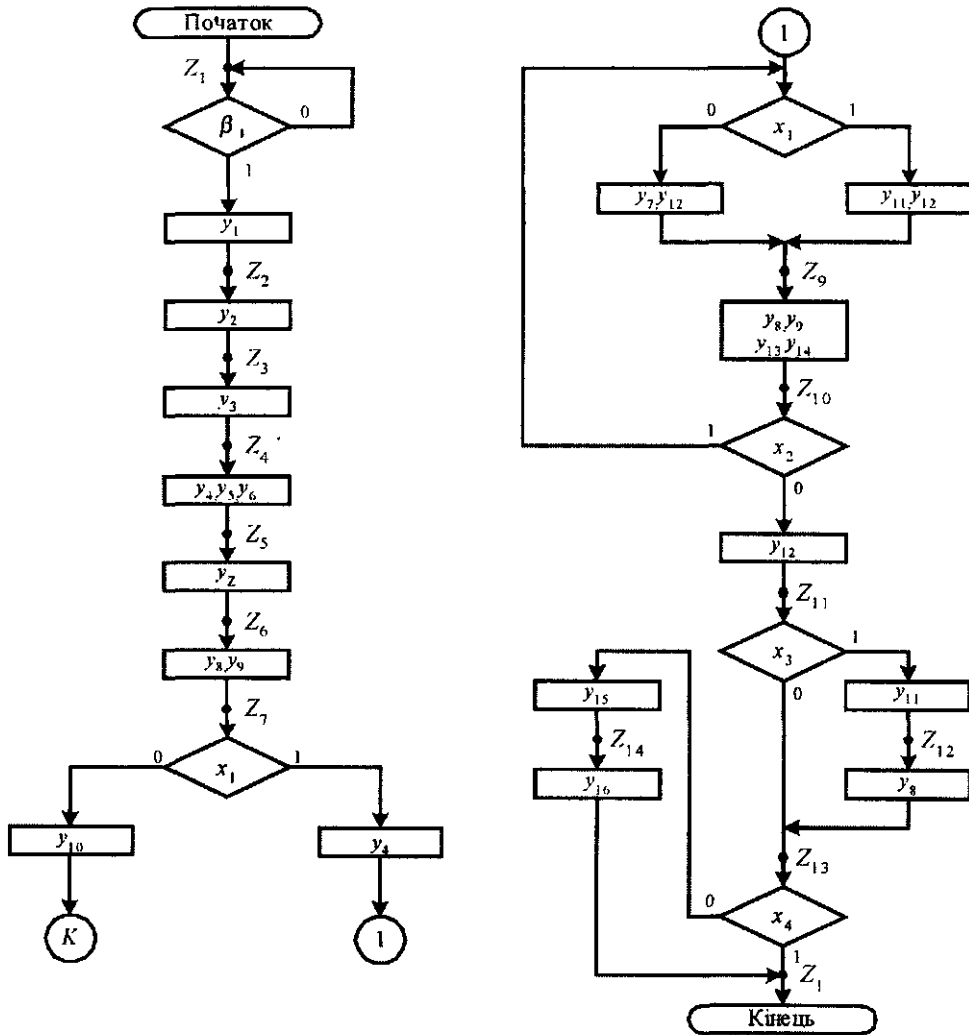


Рис. 9.24. Закодований і розмічений граф мікропрограми ділення

9.5.5. Принципіальна схема модуля операційного блока для мікропрограми ділення

Принципіальна схема модуля МОБ (рис. 9.25, 9.26) містить:

- $DD1$ — восьмирозрядний регістр дільника RGA типу $IP35$;
- $DD2, DD3$ — двоканальний восьмирозрядний мультиплексор MUX на двох мікросхемах типу $KP16$;
- $DD4, DD5, DD32$ — мікросхеми логічних елементів "виключальне ЧИ" типу $ЛП5$;
- $DD6, DD9$ — восьмирозрядні регістри діленого RGB і RGC типу $IP13$;
- $DD7, DD8$ — восьмирозрядний суматор SM на двох мікросхемах типу $IM6$;
- $DD10, DD11, DD26, DD27$ — вісім JK -тригерів типу $TB6$;

- DD12–DD15 — чотири мікросхеми логічних елементів НЕ І типу ПА3;
- DD16 — дешифратор типу ИД3;
- DD17, DD18, DD33 — мікросхеми інверторів типу ЛН1;
- DD19, DD29 — мікросхеми логічних елементів І типу ЛИ1;
- DD20–DD22 — мікросхеми логічних елементів НЕ І типу ПА2;
- DD23, DD25 — мікросхеми логічних елементів НЕ І типу ПА4;
- DD24 — мікросхема логічних елементів НЕ І типу ПА1;
- DD28 — двійковий чотирирозрядний лічильник циклів типу ИЕ10;
- DD30, DD31, DD34 — мікросхеми логічних елементів ЧИ типу ЛЛ1.

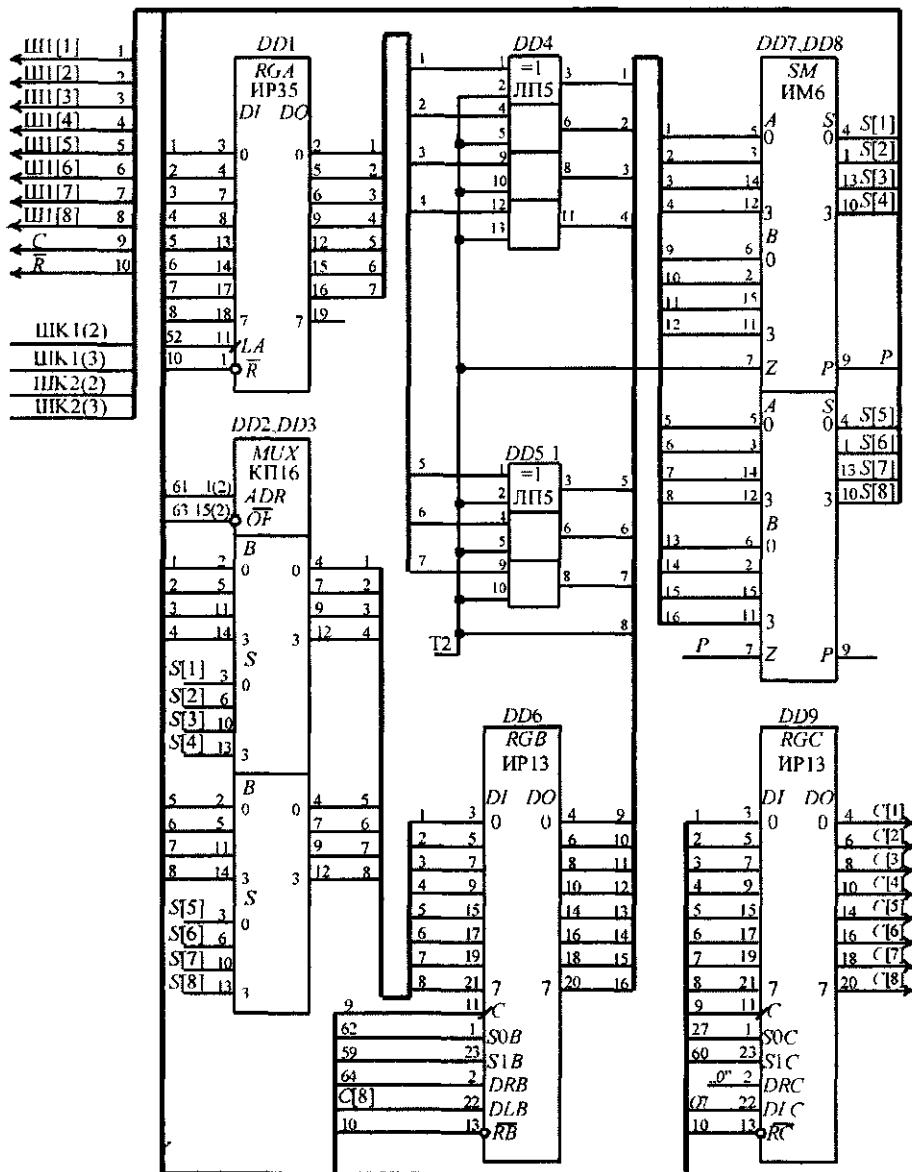


Рис. 9.25. Принципіальна схема МОБ для мікропрограми ділення (початок)

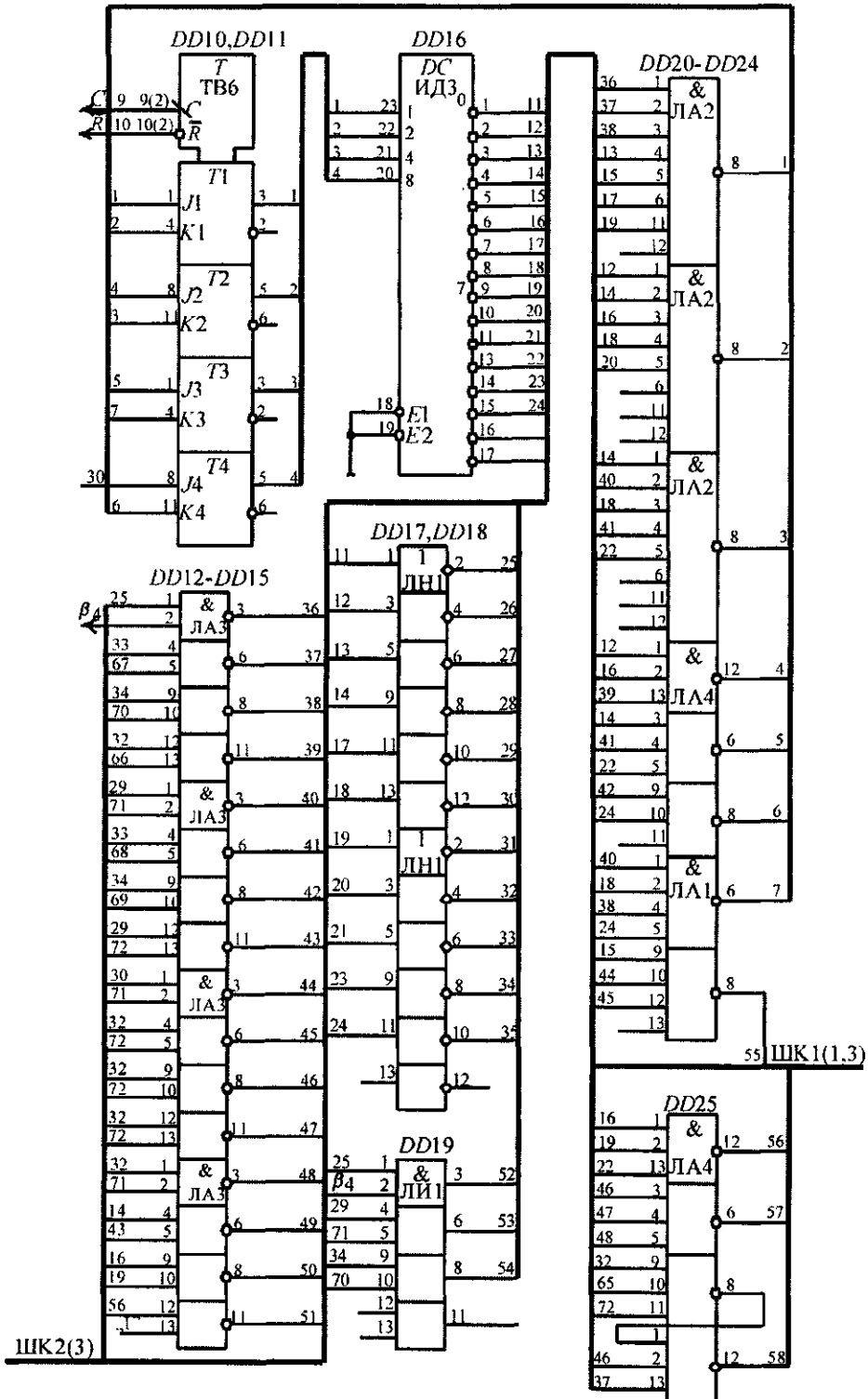


Рис. 9.26. Принципіальна схема МОБ для мікропрограми ділення (продовження)

9.5.6. Схема другого рівня керування операційним блоком

Традиційно при проектуванні довільного цифрового обчислювального пристрою керуючий автомат Мілі або Мура розглядається як перший рівень керування. Це справедливо і для керуючих автоматів на основі програмовної логіки.

Сигнали з виходів керуючого автомата ініціюють в операційному блоці виконання деякої множини мікрооперацій. Зазвичай кожний сигнал мікрооперації поступає на вхід відповідного керуючого входу функціонального вузла.

У разі використання сучасних мікросхем середнього рівня інтеграції (регістрів, лічильників, мультиплексорів і т. ін.) для реалізації складних мікропрограм типу множення, ділення та інших операцій проектування ускладнюється. Це зумовлено такими факторами:

- один керуючий сигнал може одночасно поступати на декілька входів однієї мікросхеми;
- один і той самий сигнал мікрооперації може одночасно поступати на входи декількох різних мікросхем;
- регістри в серії КР1533 побудовані на D -тригерах, що майже виключає можливість записування в них інформації за окремими розрядами, наприклад, знака добутку або знака ділення в старший розряд.

В такому випадку можливо використовувати схему другого рівня керування, яка будується в такій послідовності:

- записується таблиця відповідностей між входами керування мікросхем операційного блока і сигналами мікрооперацій ϕ_i ;
- на основі таблиці відповідності записуються диз'юнкції сигналів мікрооперацій для кожного керуючого входу мікросхем операційного блока;
- на підставі одержаних логічних рівнянь визначається необхідний склад мікросхем і будується схема другого рівня керування.

Відповідність між керуючими входами мікросхем операційного блока МОБ5 наведено в табл. 9.10.

У табл. 9.10 показано керуючі входи мікросхем, на які подаються сигнали мікрооперацій:

- $J5-J8$ — установлення тригерів $T5-T8$;
- $K5-K8$ — обнулення тригерів $T5-T8$;
- $S1B, S0B$ — задання режимів універсального регістра RGB (00 — зберігання даних, 11 — записування даних, 01 і 10 відповідно — зсув вправо і вліво);
- $S1C, S0C$ — задання режимів роботи регістра RGC ;
- \overline{OE} — дозвіл передачі даних мультиплексору;
- ADR — адресує канал мультиплексування: $ADR = 0, \overline{OE} = 0$ — передаються дані із вхідної шини; $ADR = 1, \overline{OE} = 0$ — комутуються дані з виходів суматора;
- \overline{LCT} — завантаження лічильника числом циклів — константою $0111_{(2)} = 7_{(10)}$;
- LA — завантаження регістра RGA ;
- KA — сприйняття сигналу переповнення;
- CD — вхід віднімання лічильника.

Таблиця 9.10

y_i	T5		T6		T7		T8		RGA		RGB		RGC		MUX		CT2		SM	T _П
	K5	J5	K6	J6	K7	J7	K8	J8	LA	SIB	SOB	SIC	SOC	ADR	OE	LCT	CD	z ₁	J _П	
y_1	y_1		y_1		y_1															
y_2		$y_2 \cdot B[8]$								y_2	y_2			\bar{y}_2	\bar{y}_2					
y_3												y_3	y_3							
y_4										y_4		y_4								
y_5					$y_5 \cdot 3H$															
y_6																\bar{y}_6				
y_7			y_7																T2	
y_8										y_8	y_8			y_8	y_8					
y_9								$y_9 \cdot SM[8]$												
y_{10}																				y_{10}
y_{11}			y_{11}																T2	
y_{12}					y_{12}		y_{12}													
y_{13}						$y_{13} \cdot SM[8]$														
y_{14}																				y_{14}
y_{15}										y_{15}										
y_{16}											y_{16}									

На підставі даних табл. 9.10 записується система логічних рівнянь, які є основою для схемної реалізації другого ступеня керування ОБ:

$$\begin{aligned}
 J5 &= y_2 B[8]; & K5 &= y_1; & ADR &= \bar{y}_2 \vee y_8; \\
 J6 &= y_7; & K6 &= y_1 \vee y_{11}; & \overline{OE} &= \bar{y}_2 \vee \bar{y}_8; \\
 J7 &= y_5 \cdot 3H \vee y_{13} \cdot \overline{SM}[8]; & K7 &= y_1 \vee y_{12}; & Z1 &= T2; \\
 J8 &= y_9 \cdot SM[8]; & K8 &= y_1 \vee y_{12}; & LA &= y_1; \\
 SIB &= y_2 \vee y_4 \vee y_8 \vee y_{16}; & SOC &= y_3; & \overline{LCT} &= \bar{y}_6; \\
 SOB &= y_2 \vee y_8 \vee y_{16}; & & & KA &= y_{10}. \\
 SIC &= y_3 \vee y_4; & & & &
 \end{aligned}$$

9.5.7. Принципіальна схема автомата Мілі для операції ділення

На рис. 9.24 виконано розміщення графу мікропрограми ділення для автомата Мілі. На основі закодованої і розміченої мікропрограми будується граф автомата Мілі для реалізації операції ділення (рис. 9.27).

Максимальна кількість станів автомата $L = 14$.

Стани кодуються логічними виразами таким чином:

$$\begin{aligned}
 z_1 &= \bar{Q}_4 \bar{Q}_3 \bar{Q}_2 \bar{Q}_1; & z_5 &= \bar{Q}_4 Q_3 \bar{Q}_2 \bar{Q}_1; & z_9 &= Q_4 \bar{Q}_3 \bar{Q}_2 \bar{Q}_1; & z_{13} &= Q_4 Q_3 \bar{Q}_2 \bar{Q}_1; \\
 z_2 &= \bar{Q}_4 \bar{Q}_3 \bar{Q}_2 Q_1; & z_6 &= \bar{Q}_4 Q_3 \bar{Q}_2 Q_1; & z_{10} &= Q_4 \bar{Q}_3 \bar{Q}_2 Q_1; & z_{14} &= Q_4 Q_3 \bar{Q}_2 Q_1; \\
 z_3 &= \bar{Q}_4 \bar{Q}_3 Q_2 \bar{Q}_1; & z_7 &= \bar{Q}_4 Q_3 Q_2 \bar{Q}_1; & z_{11} &= Q_4 \bar{Q}_3 Q_2 \bar{Q}_1; & z_{15} &= Q_4 Q_3 Q_2 \bar{Q}_1; \\
 z_4 &= \bar{Q}_4 \bar{Q}_3 Q_2 Q_1; & z_8 &= \bar{Q}_4 Q_3 Q_2 Q_1; & z_{12} &= Q_4 \bar{Q}_3 Q_2 Q_1; & z_{16} &= Q_4 Q_3 Q_2 Q_1.
 \end{aligned}$$

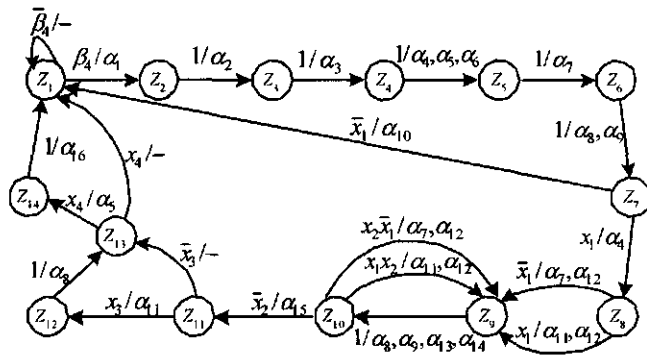


Рис. 9.27. Граф автомата Мілі для операції ділення

На основі графу автомата Мілі для операції ділення будується пряма структурна таблиця переходів і виходів (табл. 9.11).

Таблиця 9.11

z _i	K(z _i)	z _j	K(z _j)	X(z _i z _j)	Y(z _i z _j)	JK	
						J	K
z ₁	0000	z ₁	0000	$\bar{\beta}_3$	-	-	-
	0000	z ₂	0001	β_3	α_1	J1	-
z ₂	0001	z ₃	0010	1	α_2	J2	K1
z ₃	0010	z ₄	0011	1	α_3	J1	-
z ₄	0011	z ₅	0100	1	$\alpha_4, \alpha_5, \alpha_6$	J3	K2, K1
z ₅	0100	z ₆	0101	1	α_7	J1	-
z ₆	0101	z ₇	0110	1	α_8, α_9	J2	K1
z ₇	0110	z ₈	0111	x_1	α_4	J1	-
	0110	z ₁	0000	\bar{x}_1	α_{10}	-	K3, K2
z ₈	0111	z ₉	1000	x_1	α_{11}, α_{12}	J4	K3, K2, K1
	0111	z ₉	1000	\bar{x}_1	α_7, α_{12}	J4	K3, K2, K1
z ₉	1000	z ₁₀	1001	1	α_8, α_9	J1	-
					α_{13}, α_{14}		
z ₁₀	1001	z ₁₁	1010	\bar{x}_2	α_{15}	J2	K1
	1001	z ₉	1000	$x_1 \cdot x_2$	α_{11}, α_{12}	-	K1
	1001	z ₉	1000	\bar{x}_1	α_7, α_{12}	-	K1
z ₁₁	1010	z ₁₂	1011	x_3	α_{11}	J1	-
		z ₁₃	1100	\bar{x}_3	-	J3	K2
z ₁₂	1011	z ₁₃	1100	1	α_8	J3	K2, K1
z ₁₃	1100	z ₁₄	1101	\bar{x}_4	α_{15}	J1	-
	1100	z ₁	0000	x_4	-	-	K4, K3
z ₁₄	1101	z ₁	0000	1	α_{16}	-	K4, K3, K1

За даними структурної табл. 9.11 записуємо системи логічних рівнянь:

• для функцій збудження:

$$\begin{aligned}
 J1 &= z_1\beta_4 \vee z_{11}x_3 \vee z_{13}\bar{x}_4 \vee z_3 \vee z_5 \vee z_7 \vee z_9; & K1 &= z_2 \vee z_4 \vee z_6 \vee z_8 \vee z_{10}; \\
 J2 &= z_2 \vee z_6 \vee z_{10}\bar{x}_2; & K2 &= z_4 \vee z_7\bar{x}_1 \vee z_8 \vee z_{11}\bar{x}_3 \vee z_{12}; \\
 J3 &= z_4 \vee z_{12} \vee z_{11}\bar{x}_3; & K3 &= z_7\bar{x}_1 \vee z_8 \vee z_{13}\bar{x}_4 \vee z_{14}; \\
 J4 &= z_8; & K4 &= z_{13}x_4 \vee z_{14}.
 \end{aligned}$$

• для вихідних сигналів:

$$\begin{aligned}
 \alpha_1 &= z_1\beta_4; & \alpha_2 &= z_2; & \alpha_3 &= z_3; & \alpha_4 &= z_4 \vee z_7x_1; & \alpha_5 &= z_4; & \alpha_6 &= z_4; \\
 \alpha_7 &= z_5 \vee z_8\bar{x}_1 \vee z_{10}x_1; & \alpha_8 &= z_6 \vee z_9 \vee z_{12}; & \alpha_9 &= z_6 \vee z_9; & \alpha_{10} &= z_7\bar{x}_1; \\
 \alpha_{11} &= z_8x_1 \vee z_{10}x_2x_1 \vee z_{11}x_3; & \alpha_{12} &= z_8x_1 \vee z_{10}x_2 \vee z_{10}\bar{x}_1; \\
 \alpha_{13} &= z_9; & \alpha_{14} &= z_9; & \alpha_{15} &= z_{13}\bar{x}_4.
 \end{aligned}$$

На основі одержаних систем рівнянь будувється логічна схема автомата Мілі для операції ділення (рис. 9.28).

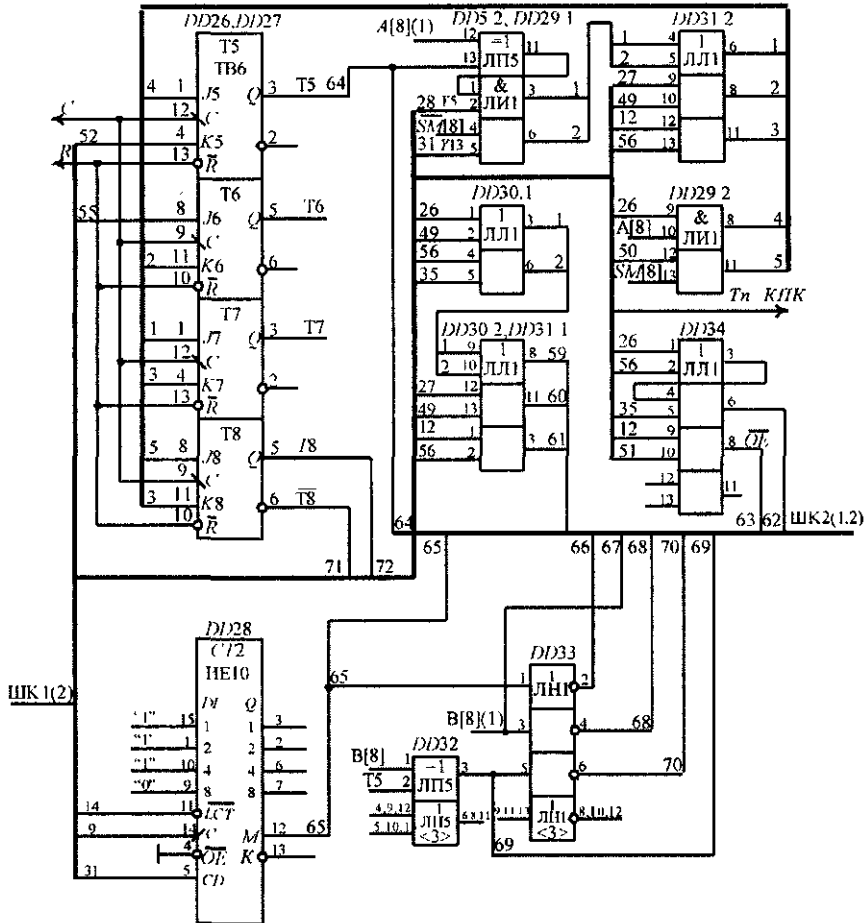


Рис. 9.28. Принципіальна схема модуля МКБ

В даному випадку пристрій АЛП реалізується на основі синхронного принципу керування, за якого:

- тривалість машинного такту T_C для всіх мікрокоманд, поданих операторними вершинами мікропрограми ділення, постійна;
- усі мікрооперації поділяються на дві групи: короткі (приймання і пересилання операндів, зсуви, мультиплексування та ін.) і довгі (додавання або віднімання дільника від діленого);
- тривалість машинного такту T_C обирається з умови виконання за один такт довгої мікрооперації;
- тривалість синхросигналу t_{CC} обирається з умови виконання коротких мікрооперацій;
- тривалість машинного такту пов'язана з тривалістю синхросигналу співвідношенням $T_C = (2...3)t_{CC}$.

Уся множина операторних вершин мікропрограми ділення (див. рис. 9. 23) послідовно розбивається на групи K_1, K_2, \dots, K_7 з певною кількістю операторів з однаковими ймовірностями звернення P_1, P_2, \dots, P_7 . Групи K_4 і K_5 виконуються в циклі $n-1=7$ раз.

З урахуванням цього можна записати:

$$K_1 = 6 \text{ (оператори 2, 3, 4, 5, 6, 7), } P_1 = 1;$$

$$K_2 = 1 \text{ (оператор 1), } P_2 = 0,9;$$

$$K_3 = 14 \text{ (оператори 12, 14), } P_3 = 0,5;$$

$$K_4 = 14 \text{ (оператори 13, 14), } P_4 = 0,5;$$

$$K_5 = 1 \text{ (оператор 16), } P_5 = 1;$$

$$K_6 = 2 \text{ (оператори 18, 19), } P_6 = 0,5;$$

$$K_7 = 2 \text{ (оператори 21, 22), } P_7 = 0,5;$$

$$K_8 = 1 \text{ (оператор 9), ймовірність переповнення приймається малою } P_8 = 0,1.$$

Тривалість t_{CC} синхросигналу визначається двома складовими:

$$t_{CC} = t_y + t_{mo},$$

де $t_{кр} \leq 5t_p$ — максимальний час формування керуючих сигналів, $t_p \approx 10$ нс — середня затримка поширення сигналу в логічних елементах, $t_{mo} \leq 30$ нс — максимальний час виконання короткої мікрооперації в операційному блоці. З урахуванням цих даних запишемо:

$$t_{CC} = 5t_p + 30 = 50 + 30 = 80 \text{ нс};$$

$$T_C = 2t_{CC} = 2 \cdot 80 = 160 \text{ нс.}$$

Тривалість виконання мікропрограми ділення $T_C = 24T_C = 24 \cdot 160 \text{ нс} \approx 3,86 \text{ мкс.}$

Контрольні запитання

1. Напишіть мікропрограму додавання цілих чисел.
2. Поясніть фрагмент схеми АЛП для додавання цілих чисел.
3. Напишіть мікропрограму віднімання цілих чисел.
4. Поясніть фрагмент схеми АЛП для віднімання цілих чисел.
5. Напишіть мікропрограму множення цілих чисел.
6. Поясніть фрагмент схеми АЛП для множення цілих чисел.
7. Напишіть мікропрограму ділення цілих чисел.
8. Поясніть фрагмент схеми АЛП для ділення цілих чисел.

Розділ 10

Мікропроцесори

10.1. Загальна характеристика процесорів і мікропроцесорів

10.1.1. Процесори

Універсальні комп'ютери поділяються на три функціонально зв'язані апаратні частини: процесор, пам'ять і периферійні пристрої.

Процесор — це основна функціональна частина комп'ютера, яка інтерпретує й виконує команди, тобто безпосередньо реалізує програмно-керований процес обробки даних. Процесор, який виконує в обчислювальній системі основні функції, називають **центральною** (ЦП). Спеціалізований процесор, призначений для керування зовнішніми пристроями (накопичувачами, дисплеями, принтерами та ін.) називають **контролером**.

Процесор характеризується архітектурою, до якої відносять:

- список арифметико-логічних операцій (система команд);
- типи і формати команд і даних;
- організацію адресного простору пам'яті і периферійних пристроїв;
- способи адресації команд і даних;
- функції складових частин і структуру зв'язків з іншими пристроями машин та режими роботи.

Процесор складається з пристрою керування, арифметико-логічного пристрою та блоку інтерфейсу (БІФ) для з'єднання із зовнішнім середовищем — пам'яттю, периферійними пристроями (рис. 10.1).

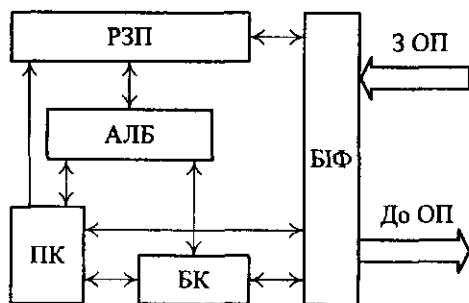


Рис. 10.1. Структура процесора

Оброблення даних здійснюється в АЛП, який містить арифметико-логічний блок АЛБ, блок РЗП, блок контролю БК і місцевий блок керування при децентралізованому керуванні.

Арифметико-логічний блок має універсальний двійковий комбінаційний суматор, двійково-десятьковий суматор або схему десяткової корекції, регістри для тимчасового зберігання двох операндів і результату операцій та регістр прапорців. Для підвищення продуктивності в

АЛП можуть включати спеціалізовані вузли-зсувачі, помножувачі, схеми прискореного переносу та інші пристрої.

Ряд процесорів мають по два і більше АЛП. Розрядність АЛП визначає розрядність всього процесора.

В РЗП зберігаються початкові дані, проміжні та кінцеві результати, адреси даних, константи, які необхідні в процесі виконання команди. Всі операції в АЛП реалізуються як просторово-часові послідовності мікрооперацій над двійковими словами, кожна з яких є сукупністю булевих операцій над бітами слів. В АЛП реалізуються такі типи мікрооперації

- передачі слів між регістрами та регістрами і пам'яттю,
- додавання двох слів, декремент (мінус 1) або інкремент (плюс 1) слова,
- арифметичні, логічні та циклічні зсуви вправо чи вліво,
- порозрядні логічні операції ЧИ, І, виключальне ЧИ та порівняння операндів,
- перетворення кодів слів — інверсія, доповнення, розширення та ін

Пристрій керування ПК керує процесом оброблення даних, забезпечує основні режими роботи (початкових установлень, очікування, переривання, прямого доступу до пам'яті, діагностики і контролю) та взаємодію всіх пристроїв комп'ютера. Для виконання цих функцій ПК має в своєму складі регістр і дешифратор команд, програмний лічильник для задання адреси наступної команди, блок керування та схеми синхронізації, діагностики й контролю.

До складу процесора можуть входити спеціальні системні засоби (служба часу, засоби міжпроцесорного зв'язку, пульт керування та ін.)

Пристрій керування послідовно зчитує код команди з пам'яті і розміщує його в регістр команд (інструкцій). Блок керування дешифрує команду і формує послідовності керуючих сигналів. Для виконання однієї мікрооперації в АЛП необхідний один керуючий сигнал. В одному машинному такті реалізується сукупність мікрооперацій — мікрокоманда. Множина мікрокоманд створює мікропрограму команди. Кожна команда має свою мікропрограму, час виконання якої називається **командним циклом**.

Розрізняють апаратні, мікропрограмні та комбіновані блоки керування. Апаратні блоки керування побудовані на основі схемної логіки, а мікропрограмні — програмованою логікою (мають пам'ять мікропрограм). Комбіновані блоки керування використовують обидва способи їхньої реалізації.

При централізованому керуванні один ПК керує процесом оброблення команд і даних у всій машині. При децентралізованому керуванні ПК формує основні керуючі сигнали, а опрацюванням даних керує місцевий блок керування, розміщений в АЛП.

Зазначимо, що всі команди в комп'ютері реалізуються на основі принципу мікропрограмного керування, тобто виконання мікропрограм. В той же час ПК з програмованою логикою також називають мікропрограмними. Методологічно ці поняття слід розрізняти.

10.1.2. Мікропроцесори

Мікропроцесор являє собою процесор, складові частини якого мініатюризовані та розміщені в одній або декількох мікросхемах. Історія мікропроцесорів почалася в 1971 р., коли фірма *Intel* розробила перший у світі чотирирозрядний процесор 4004, а через рік — восьмирозрядний 8008. На основі мікропроцесорів створено нову елементну базу — мікропроцесорні засоби, на яких будують сучасні комп'ютери. У наш час терміни "процесор" і "мікропроцесор" сприймають як синоніми.

Мікросхема, яка виконує функції мікропроцесора або його частини, називається **мікропроцесорною**. Сукупність мікропроцесорних та інших мікросхем, які сумісні за конструктивно-технологічним виконанням і призначені для спільного використання, називається **мікропроцесорним комплектом (МПК)**

До характеристик мікропроцесорних ІМС відносяться розміри кристала і кількість транзисторів у ньому, тип корпусу і кількість виводів

Найбільш важливими статичними і динамічними електричними параметрами мікропроцесорів як мікроелектронних виробів є

- кількість джерел живлення та їхня напруга,
- струм і потужність споживання,
- кількість серій синхроімпульсів, їхні частота і амплітуда; рівні логічних сигналів;
- вхідна і вихідна ємності, навантажувальна здатність;
- час затримки розповсюдження сигналів, число операцій в секунду над операандами, які зберігаються в регістрах-акумуляторах.

Залежно від умов експлуатації до мікропроцесорних ІМС можуть пред'являтися спеціальні вимоги щодо додержання

- температурного діапазону роботи і режиму зберігання,
- стійкості до радіаційних і електромагнітних впливів, вібрацій та ударів;
- маси і габаритів,
- інтенсивності відмов, напрацювання на відмову, надійності функціонування.

Мікропроцесор, призначений для використання функціонально повного набору операцій, є універсальним, інакше — спеціалізованим. За конструкцією розрізняють такі мікропроцесори:

- **однокристалні** — виконані у вигляді єдиної мікросхеми (рис. 10.2, а);
- **багатокристалні (модульні)** — побудовані з мікросхем, кожна з яких виконує функції пристрою комп'ютера (рис. 10.2, б),
- **розрядно-нарощувальні (секційні)**; розрядність їхня може збільшуватися за рахунок декількох мікропроцесорних секцій, об'єднаних спільними шинами (рис. 10.2, в)

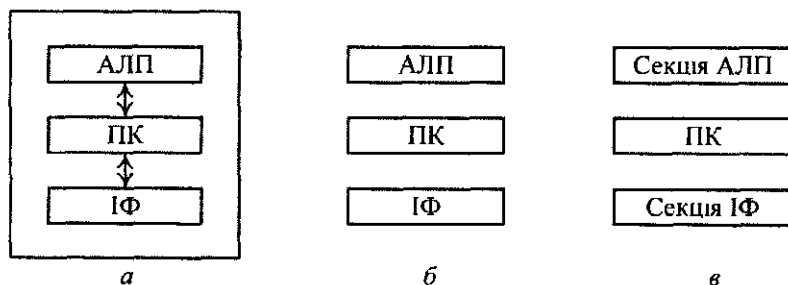


Рис. 10.2. Конструкції мікропроцесорів:

а — однокристалні; б — багатокристалні; в — секційні

Усі мікропроцесори мають засоби сполучення із зовнішніми пристроями — інтерфейси (ІФ).

Для виробництва мікропроцесорів використовують усі види схемотехніки (технології). ТТЛШ, ЕЗЛ, І²Л, n-МОН, p-МОН, КМОН та ін.

Залежно від режиму роботи розрізняють такі процесори:

- однопрограмні (виконують одну програму) і багатопрограмні (мають засоби для одночасного виконання кількох програм);
- мультипроцесори (системи, в яких одночасно можуть бути активними декілька процесорів);
- конвеєрні (команди виконуються послідовно рядом пристроїв, причому різні пристрої можуть одночасно обробляти відповідні частини декількох команд);
- матричні (мають спеціальну архітектуру, розраховану на оброблення числових масивів);
- співпроцесори (арифметичні розширювачі) — призначені для розширення списку команд ЦП; самостійно не використовуються;
- периферійні – виконують функції введення-виведення інформації
- асоціативні процесори, в яких характер обробки даних визначається змістом самих даних.

За видом оброблюваної інформації розрізняють цифрові (звичайні) та аналогові мікропроцесори. В аналогових мікропроцесорах на вході використовують АЦП для перетворення аналогових величин в цифровий код, а на виході — схеми ЦАП, які перетворюють цифрові дані в аналогові.

10.1.3. Мікропроцесорні комплекти

Для побудови сучасних комп'ютерів використовують мікропроцесорні засоби — комплекси апаратно і програмно сумісних ІМС різного ступеню складності (СІС, ВІС, НВІС та ультра-ВІС). Основою мікропроцесорних засобів є мікропроцесорні комплекти МПК і базові кристали, ВІС пам'яті (рис. 10.3).

В нашій країні розроблений і випускається мікроелектронною промисловістю широкий набір серій МПК. Основні параметри більшості МПК (серія, підгрупа і вид базового мікропроцесора, технологія, швидкодія F в MIPS для операцій типу реєстр-реєстр, розрядність n в бітах, споживана потужність P_{CC} у ватах та число мікросхем в серії) наведені в табл. 10.1.

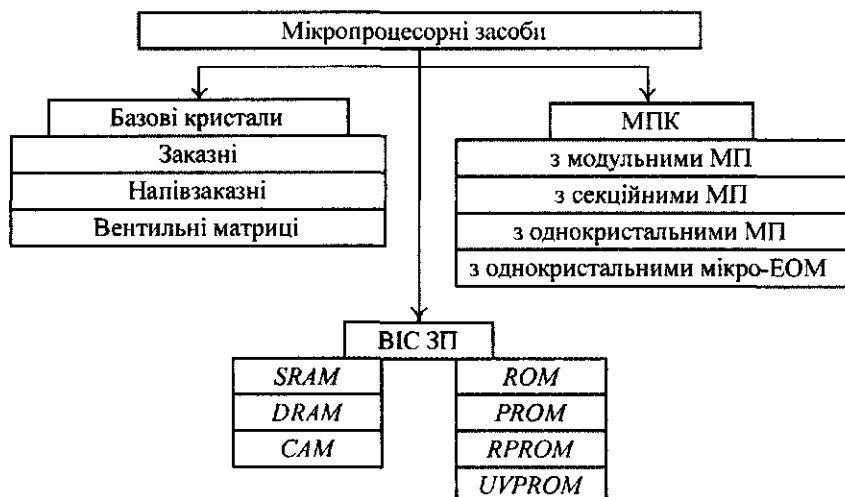


Рис. 10.3. Склад мікропроцесорних засобів

Таблиця 10.1

Серія МПК	Тип ІМС	Технологія	F , MIPS	n , біт	$P_{\text{сст}}$, Вт	Число ВІС, шт.
K580	BM80	n -МОН	0,5	8	1,5	21
K581	BE1	І ² Л	0,4		0,9	6
K582	BM1	І ² Л	1,5		0,3	
K583	BC1	І ² Л	0,1	8	0,6	11
K584	BM1A	І ² Л	0,5	4	0,25	4
K586	BM1	n -МОН	0,5		0,9	
K587	BM2	КМОН	0,5		0,005	4
K588	BC2A	КМОН	0,5	16	0,05	21
K589	BC02	ТТЛШ	1,0	2	1,0	8
K1800	BC01	ЕЗЛ	3,0	4	1,7	10
K1801	BM1	n -МОН	0,5	16	1,4	9
K1802	BC1	ТТЛШ	0,8	8	1,4	15
K1804	BC1	ТТЛШ	0,8	4	1,5	29
K1808	BM1	І ² Л	0,3	8	0,05	4
K1810	BM86	n -МОН	0,7	16	0,9	22
K1811	BM1	n -МОН	0,5	16	1,1	5
K1813	BE1	n -МОН	1,2	28		1
K1815	ВФ1	І ² Л	0,5	16	0,3	9
K1816	BE51	n -МОН	1,0	8	0,7	
K1820	BE1	n -МОН	0,5	8	0,6	2
K1821	BM85	n -МОН	0,5	8	1,0	12
K1827	BE1	n -МОН	0,5	16		11
K1838	BM1	ЕЗЛ	5,0	16		11

Залежно від виду і типу базової програмної ВІС виділяють такі МПК:

- з одним або кількома однокристальними мікропроцесорами: серії KP580, KP581, K1810, K1518, K1821, KP1828, KM1831, K1838;
- з багатокристальними (модульними) мікропроцесорами: K581, K588;
- із секціонованими (розрядно-модульними) мікропроцесорами: K583, K1800, K1802, K1804, K1822;
- з однокристальними мікро-ЕОМ: K1813, K1816, K1820, K1827.

Однокристальні мікропроцесори мають фіксовану структуру і систему команд; призначені для побудови мікро- та міні-комп'ютерів широкого застосування. Однокристальні мікро-ЕОМ вбудовуються в технічні системи, в яких вони виконують функції контролю, керування і діагностики.

Модульні мікропроцесори орієнтовані на реалізацію конкретного класу алгоритмів (оброблення цифрових сигналів, графічні перетворення та ін.). Секціоновані мікропроцесори забезпечують побудову високопродуктивних універсальних комп'ютерів методом нарощування необхідного числа секцій.

До складу МПК входить широкий спектр спеціалізованих мікропроцесорних ІМС, в тому числі:

- процесорні елементи (секції АЛП); блоки регістрів загального і спеціального

призначення; багаторежимні буферні регістри (порти, фіксатори); схеми прискорених перенесень і розширювачі обчислювальних функцій;

- схеми синхронізації та генератори тактових імпульсів; блоки керування обміном з пам'яттю та формування адрес мікрокоманд;
- контролери: клавіатури, відеотерміналів, накопичувачів на магнітних дисках і стрічках, введення–виведення даних, прямого доступу до пам'яті і переривань;
- контролери шин і протоколів, шинні формувачі;
- програмовні таймери і схеми арбітражу.

За характером роботи в часі мікропроцесори поділяються на синхронні (час виконання команд в них постійний) і асинхронні, де кожна команда реалізується за час її фактичного виконання. Мікропроцесори мають різні пристрої керування із схемною або програмною логікою та комбіновані.

Поява мікропроцесорів — це еволюційний процес у технології виробництва мікросхем, а для обчислювальної техніки — перехід у новий якісний стан.

До техніко-економічних переваг мікропроцесорів відносять:

- високу швидкість і надійність, малу потужність споживання, малі розміри і масу, низьку вартість;
- простоту технічного обслуговування, складання і розширення мікропроцесорних систем;
- можливість побудови потужних мікропроцесорних комплексів з паралельною обробкою інформації;
- реалізацію розподілених систем контролю, діагностики і керування.

Достоїнства мікропроцесорів забезпечили масове застосування їх у науці, техніці та нетрадиційних областях господарства.

Розвиток мікропроцесорів у часі ілюструє табл. 10.2 для виробів фірми *Intel*.

Таблиця 10.2

Назва МП	Рік випуску	Технологія	n , біт	Адресний простір	F , MIPS	N_{TP}
4004	1971	p -МОН	4	640 байт	0,1	2800
4040	1972	p -МОН	4		0,15	3500
8008	1973	p -МОН	8	16 Кбайт	1,0	4500
8080	1974	n -МОН	8	64 Кбайт	5,0	5000
8085	1975	n -МОН	8	64 Кбайт	5–10	6500
8086	1978	n -МОН	16	1 Мбайт	5–10	29000
80186	1980	КМОН	16			
80286	1982	КМОН	16	16 Мбайт		134000
80386	1985	КМОН	32	4 Гбайт		275000
80486	1991	КМОН	32	4 Гбайт		1,2 млн
Pentium	1993	КМОН	32	4 Гбайт	75–200	3,1 млн
Pentium Pro	1995	КМОН	32	4 Гбайт	180–200	5,5 млн
Pentium MMX	1997	КМОН	32	4 Гбайт	166–200	4,5 млн
Pentium II	1997	КМОН	32	4 Гбайт	233–450	–
Pentium III	1998	БіКМОН	32	4 Гбайт	450–933	–
Itanium (Merced)	2000	БіКМОН	64			

Примітка: МП — мікропроцесор, $N_{\text{ТР}}$ — число транзисторів у МП.

Конструктивно-технологічний розвиток мікропроцесорів відбувається в таких напрямках.

1. Використовують нові технології, наприклад, БіКМОН, в яких комбінуються біполярні транзистори (для збільшення швидкості) та КМОН структури (для зменшення споживаної потужності та підвищення щільності компоновки). Якщо в перших мікропроцесорах відстань між сусідніми лініями дорівнювала 10 мкм, то в останніх виробках вона дорівнює 0,18 мкм.
2. Зростає рівень інтеграції: від 2800 транзисторів в чипах перших мікропроцесорів до 10–12 млн і більше в останніх виробках типу *Itanium*. За законом Мура характеристики мікросхем мають поліпшуватися в два рази кожні 18 місяців при збереженні вартості. За прогнозом експертів у 2012 р. на чипі буде розташовуватись 1,4 млрд транзисторів, які мають працювати на частоті 2700 МГц.
3. Збільшується розрядність оброблюваних даних: від чотирьох — у перших мікропроцесорах до 64 — в *Itanium* (зросла в 16 разів).
4. Прискорюється зміна поколінь мікропроцесорів. Якщо до 1994 р. нові покоління з'являлися кожні три роки, то тепер цей цикл зменшився до двох років. Вартість сучасного заводу, який випускає чипи з розмірами транзисторів 0,25 мкм становить \$ 2 млрд, а через 5–10 років такий комплекс буде коштувати \$ 4 млрд.

10.1.4. Архітектура мікропроцесорів

Архітектуру мікропроцесора характеризують: список команд та їхні формати; способи адресації; розрядність і ємність адресованої пам'яті; структура регістрів та їхні функції та ін.

Історично першими склалися такі основні архітектури мікропроцесорів: з акумулятором, з РЗП, зі стековою організацією та комбіновані (рис. 10.4).

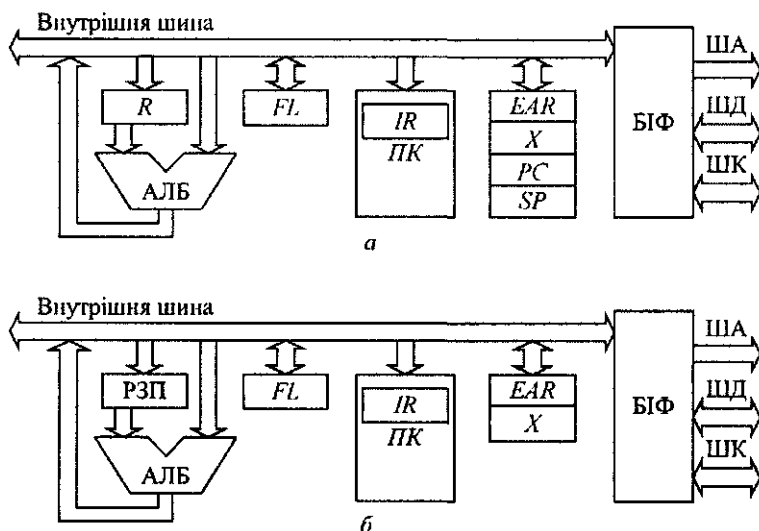


Рис. 10.4. Структура мікропроцесорів: а — з акумулятором і стеком; б — з РЗП

Структура всіх мікропроцесорів містить такі однакові пристрої, вузли і блоки, об'єднані спільною внутрішньою шиною даних: АЛБ; ПК; *IR* — реєстр команд (інструкцій); *FL* — реєстр ознак (прапорців); *EAR* — реєстр виконавчої адреси; БІФ — блок інтерфейсу з вихідними шинами адреси, даних і керування.

В архітектурі мікропроцесора з акумуляторами (рис. 10.4, а) додатково використовують індексний реєстр *X* і покажчик стека *SP*; реєстр *R* виконує функції акумулятора *A*. При виконанні арифметико-логічних операцій перший операнд попередньо розміщується в акумуляторі, а другий — з ОП безпосередньо поступає на вхід АЛБ. Результат операції розміщується в акумуляторі, а в реєстр *FL* автоматично записуються ознаки операції.

В архітектурі мікропроцесора з РЗП (рис. 10.4, б) звичайно використовують від восьми до 16 реєстрів, кожний з яких може виконувати функції акумулятора. Збільшення числа РЗП значно зменшує кількість звернень до ОП, що підвищує продуктивність комп'ютера. Але збільшення числа реєстрів призводить до втрат машинного часу у випадках переривання програм: при цьому необхідно зберегти зміст РЗП в пам'яті (звичайно у стеку), а потім відновити його після обробки переривання. Тому при виборі кількості РЗП враховують цей фактор.

У мікропроцесорі зі стековою архітектурою (рис. 10.4, а) відсутні акумулятор і РЗП, а *R* виконує функції реєстра тимчасового зберігання даних ОП на час виконання операції. Читання даних і їхнє записування у стекову пам'ять здійснюють за допомогою покажчика стека *SP*. Усі операції з даними виконують відповідно до польського запису: операнди розміщують у стеку в послідовності виконання над ними дій. У стек послідовно завантажують числа *a*, *b* і *c* (рис. 10.5).

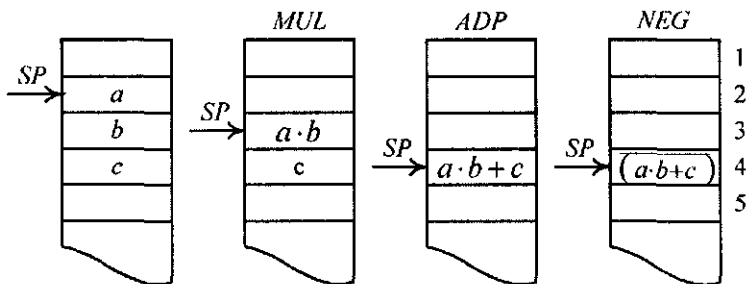


Рис. 10.5. Ілюстрація роботи стека

За командою множення *MUL* одержуємо добуток $a \cdot b$, який записується в адресі *b*. За командою додавання *ADD* маємо $a \cdot b + c$, який записується в адресі *c*. Після команди інвертування *NEG* в комірці за адресою *c* записується обернений код результату $a \cdot b + c$.

В мікропроцесорах з комбінованою архітектурою об'єднують значною мірою властивості архітектур з акумулятором, РЗП і стеком. До них відносять однокристальний мікропроцесор 8080 (1974 р., фірма *Intel*, аналог у нашій країні — КР580ВМ80А), більш досконалу його модель 8085А (аналог — КР1821ВМ85А), а також модель Z80 фірми *Zilog* (рис. 10.6).

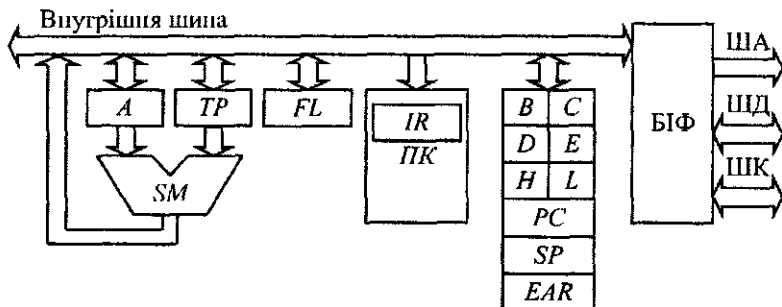


Рис. 10.6. Структура мікропроцесора з комбінованою архітектурою (близькою до моделі 8085A)

Комбінована структура містить:

- складний комбінований суматор SM (в ньому виконується більшість арифметичних і логічних операцій), реєстри A і TP ; разом вони створюють АЛБ;
- блок РЗП (реєстри B, C, D, E, H, L), програмний лічильник PC , покажчик стека SP , реєстр виконавчої адреси EAR ;
- блок ІФ з ША, ШД і ШК;
- пристрій керування ПК з реєстром команд IR .

Сукупність обчислювальних засобів, куди входять один або декілька мікропроцесорів та напівпровідникова пам'ять і засоби інтерфейса, називається **мікропроцесорною системою** (МПС).

10.1.5. Програмно керований обмін інформацією

Між ядром машини і периферією реалізуються такі види програмно керованого обміну інформацією:

- синхронний і асинхронний обміни в послідовних і паралельних двійкових кодах, які виконуються за схемою, зображеною на рис. 10.7, а;
- з перериванням програми за запитом ПП.

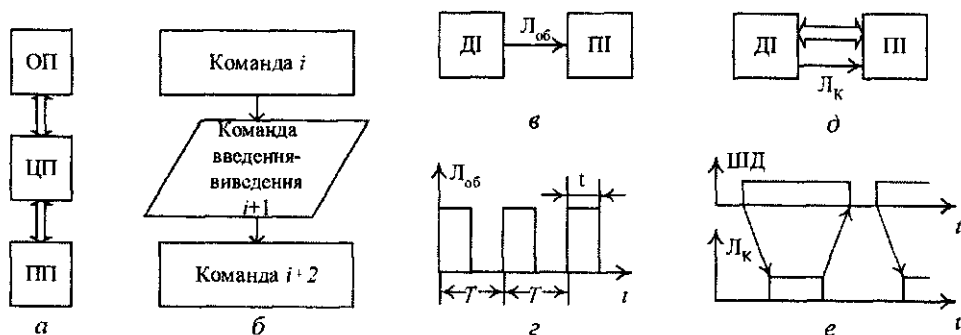


Рис. 10.7. Синхронний обмін: а — схема, б — алгоритм;

в, г — послідовний обмін без стробування; д, е — паралельний обмін із стробуванням

При синхронній передачі джерело інформації ДІ завжди готове до обміну відповідно до алгоритму (рис. 10.7, б). Джерело інформації виставляє і утримує значення

даних на лінії послідовного обміну $L_{об}$ протягом часу τ , який складається з тривалості затримки розповсюдження сигналу на лінії, його розпізнавання і фіксації в регістрі приймача ПІ (рис. 10.7, в, г).

При синхронному паралельному обміні часто використовують сигнал квитування, який передається і приймається по окремій лінії L_K і визначає інтервал часу надійного приймання даних приймачем (рис. 10.7, д, е).

Асинхронний обмін виконується при готовності (*Ready*) зовнішнього пристрою до обміну даними відповідно до алгоритму (рис. 10.8, а). При асинхронній передачі паралельного коду по ШД використовують метод квитування, в якому поєднуються спільна дія сигналу квитування L_K від джерела ДІ та сигналу підтвердження приймання $L_{П}$ від приймача ПІ (рис. 10.8, б, в).

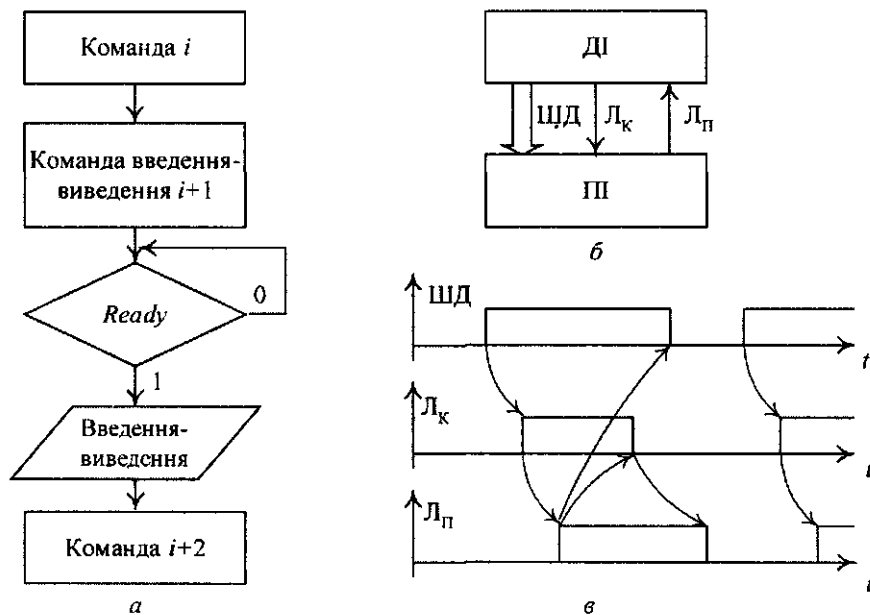


Рис. 10.8. Асинхронний обмін:

а — алгоритм; б — схема; в — часові діаграми роботи з квитуванням

Основним недоліком синхронного і асинхронного обміну є значне завантаження процесора операціями введення-виведення, що призводить до суттєвого зменшення продуктивності комп'ютера. Тому синхронний і асинхронний обміни використовують при передачі одиночних байтів чи слів. Обмін в режимі переривання здійснюють апаратно за ініціативою зовнішнього пристрою чи програмно — командою переривання *INT*. Процесор, одержавши апаратний запит на переривання, закінчує поточну команду, пересилає в ОП зміст своїх регістрів і переходить на підпрограму обслуговування переривання. Після її закінчення процесор відновлює зміст своїх регістрів і продовжує виконання перерваної програми (рис. 10.9, а).

Апаратно режим переривання забезпечується контролером переривань (КПР), до якого підключаються ПП. Вихід *INT* контролера подається на відповідний вхід ЦП, а на ШД пересилається початкова адреса підпрограми обслуговування (рис. 10.9, б). При програмному перериванні адреса підпрограми подається в самій команді переривання *INT*.

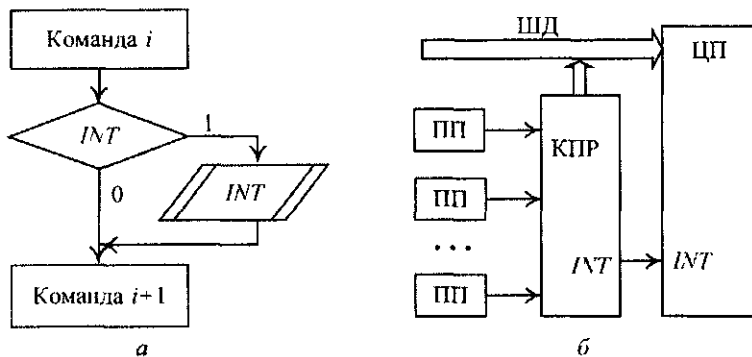


Рис. 10.9. Обмін за перериванням: а — алгоритм; б — схема підключення контролера КПР

Прямий доступ до пам'яті (ПДП) використовують для швидкого обміну масивами інформації між основною пам'яттю і периферією. При цьому процесор (мікропроцесор) звільняється від безпосереднього керування операціями введення-виведення.

ПДП реалізується відповідно до алгоритму (рис. 10.10, а).

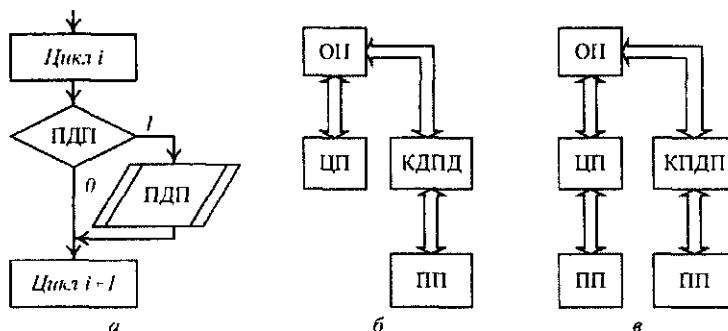


Рис. 10.10. Режим ПДП: а — алгоритм; б, в — схеми підключення контролерів КПДП

В міні- і мікрокомп'ютерах прямим доступом керує контролер прямого доступу до пам'яті (КПДП). Перед початком обміну процесор пересилає в КПДП таку інформацію (програмування контролера):

- початкову адресу області пам'яті, яка бере участь у обміні;
- напрямок операції обміну — введення чи виведення;
- кількість байтів, які підлягають передачі.

Особливість режиму ПДП — обмін даними може бути між машинними циклами в команді ("Захват циклу"), а також після закінчення команди.

Контролер ПДП керує обміном даними між ОП і ПП без участі процесора (рис. 10.10, б). При необхідності в комп'ютері використовують програмно-керований обмін окремими байтами (він не вимагає програмування КПДП) і обмін масивами у режимі ПДП (рис. 10.10, в).

В універсальних комп'ютерах обмін інформацією між ОП і ПП забезпечують спеціальні пристрої — КВВ або просто канали (рис. 10.11).

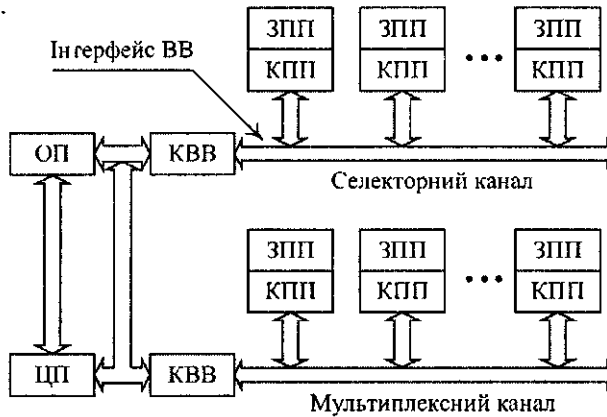


Рис. 10.11. Структура комп'ютера

В каналах використовують два режими обміну інформацією: монопольний та розподілу в часі. Монопольний режим реалізують селекторним каналом, а розподілу в часі — мультиплексним каналом. Засоби каналу, які призначені для обслуговування одного ПП, називаються підканалом.

Селекторні канали обслуговують швидкодіючі пристрої, в першу чергу ЗЗП (накопичувачі на дисках і магнітних стрічках). Селекторний канал має один підканал. Після встановлення зв'язку він монопольно обслуговує тільки один ПП: інші пристрої чекають закінчення даної програми обміну. У малих комп'ютерах використовують один селекторний канал, у великих — до шести.

Мультиплексний канал (один в машині) паралельно обслуговує сотні повільно діючих ПП в режимі розподілу часу (клавіатура, принтери, перфоратори та ін.).

10.2. Однокристальні восьмирозрядні мікропроцесори

Однокристальні восьмирозрядні мікропроцесори мають фіксовану систему команд. Вони розраховані на різноманітне застосування у складі МПС.

Відомі такі основні типи восьмирозрядних однокристальних мікропроцесорів: КР580ВМ80А (і8080), КР1821ВМ85 (і8085), 6800 (фірми *Motorola*), Z-80 (фірми *Zilog*) та інші. Розглянемо детально мікропроцесор КР1821ВМ85А.

10.2.1. Загальна характеристика мікропроцесора К1821ВМ85А

Мікропроцесор К1821ВМ85А (далі скорочено ВМ85) — це однокристальний восьмирозрядний процесор з фіксованою системою команд. Він має класичну архітектуру з одним акумулятором та окремими шинами адреси і даних. Основна область застосування мікропроцесора ВМ85 — контролери, які працюють за фіксованою програмою. Мікросхема ВМ85 виконана за *n*-МОП технологією і випускається у 48-контактному керамічному корпусі з двостороннім вертикальним розташуванням виводів (тип *DIP*).

Схема інтерфейсу містить блоки послідовного обміну даними (ПОД) та оброблення переривань (БПР). Всі функціональні компоненти мікропроцесора об'єднані внутрішньою восьмирозрядною шиною.

В АЛБ виконують арифметико-логічні операції над двійковими та двійково-десятьковими операндами. Акумулятор *A* побудований на двоступеневих тригерах. При виконанні більшості арифметико-логічних операцій використовують складний комбінаційний суматор *SM*. При цьому в акумуляторі зберігається перший операнд, а в регістрі *TP* — другий; результат операції записується в акумулятор.

Регістр *TP* бере участь в операціях міжрегістрових передач, а акумулятор використовується також при паралельному та послідовному введенні-виведенні даних. В регістр прапорців *FL* після виконання більшості операцій автоматично записуються такі ознаки:

- *CY* — прапорець перенесення, установлюється при наявності перенесення із старшого біта результату (в операціях віднімання він стає прапорцем позитивності);
- *S* — прапорець знака, збігається зі значенням старшого біта результату;
- *Z* — прапорець нуля, установлюється при нульовому результаті;
- *P* — прапорець паритету, установлюється при наявності в результаті парного числа одиниць;
- *AC* — допоміжний прапор, установлюється при наявності перенесення між тетрадами результату.

В суматорі *SM* виконуються всі арифметико-логічні операції та зсуви. Схема ДК призначена для корекції результатів додавання двійково-десятькових чисел у двійковому суматорі *SM*.

Регістри *B*, *C*, *D*, *E*, *H* та *L* призначені для зберігання даних (як операндів, так і результатів) в процесі виконання програми. В ряді операцій вони об'єднуються в регістрові пари *BC*, *DE* і *HL*, які зберігають двобайтові дані або покажчики адрес операндів у пам'яті.

Програмний лічильник *PC* призначений для зберігання адреси наступної команди, яка буде виконуватися. Зміст *PC* автоматично збільшується на константу, яка дорівнює числу байтів у команді.

Покажчик стеку *SP* зберігає адресу комірки стеку, до якого було останнє звернення. При записуванні в пам'ять зміст *SP* зменшується, а при читанні — збільшується.

Регістр *PAID* дозволяє збільшувати або зменшувати на одиницю зміст програмного лічильника *PC*, покажчик стеку *SP* і регістрових пар.

Регістр команд *R-K* використовується для приймання і зберігання на час виконання операції першого байта команди, який містить код операції. Зміст *R-K* розшифровується дешифратором команд *ДшК* і передається в блок БКС. Останній виробляє сигнали синхронізації та керування обчислювальним процесом у комп'ютері.

Блок керування перериваннями (БПР) переключає мікропроцесор з виконання однієї програми на іншу за допомогою зовнішніх сигналів переривання. Введені такі входи і відповідні їм рівні переривань:

- *TRAP* — має найвищий пріоритет і не може відключатися чи маскуватися; призначений на випадок появи похибки, яку не можна усунути (наприклад, при пошкодженні джерела живлення або появи несправності в шинах);

- *RST 7.5*, *RST 6.5*, *RST 5.5*, *INT* — сигнали переривання на цих входах можуть маскуватися (програмно забороняється їхнє сприймання). Вхід *INT* призначений для обслуговування векторних переривань за командами *RSTn* і *CALL n* з використанням контролера КР580ВН59.

На початку кожного машинного циклу буфер БА виставляє і утримує на лініях шини *A15 – A8* старший байт адреси пам'яті, а буфер БАД передає на короткий час молодший байт адреси на лінії *AD7 – AD0*, який за сигналом *AEN* має записуватися в зовнішній регістр-фіксатор. Після цього двонаправлений буфер БАД перемикається на записування або зчитування даних. Периферійні пристрої адресуються восьмирозрядним кодом, який одночасно поступає на дві частини шини адреси. При необхідності буфери БА і БАД перемикаються в *Z*-стан.

Блок ПОД за командою *RIM* здійснює послідовне введення даних, а за командою *SIM* — послідовне виведення.

Умовне графічне позначення мікропроцесора *VM85* показано на рис. 10.13. Позначення виводів мікропроцесора збігаються з іменами сигналів на них.

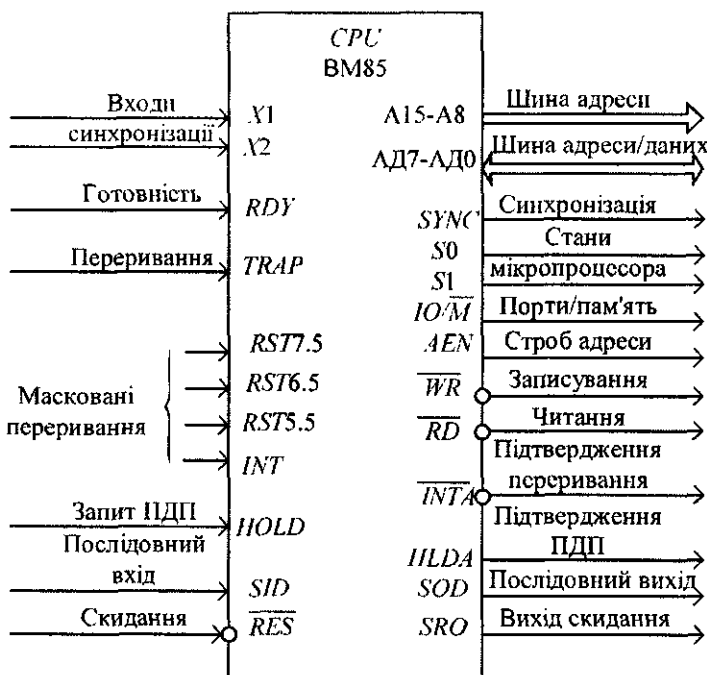


Рис. 10.13. Умовне графічне позначення мікропроцесора *VM85* і функції виводів

Порівняно з програмно сумісним мікропроцесором КР580ВМ80А мікросхема *VM86* має такі відмінності: двофазна синхронізація замінена на однофазну; збільшена в два рази частота синхронізації; замість трьох джерел живлення використовується одне; добавлені дві команди послідовного введення–виведення.

10.2.3. Функціонування мікропроцесора *VM85* у часі

Функціонування мікропроцесора *VM85* у часі визначається внутрішнім генератором тактових (синхронізуючих) імпульсів (ГТІ). Збудження його може задаватися

зовнішнім кварцовим резонатором, RC- чи LC-ланцюгами, або зовнішнім генератором, які підключаються до входів $X1$, $X2$ мікропроцесора (рис. 10.14)

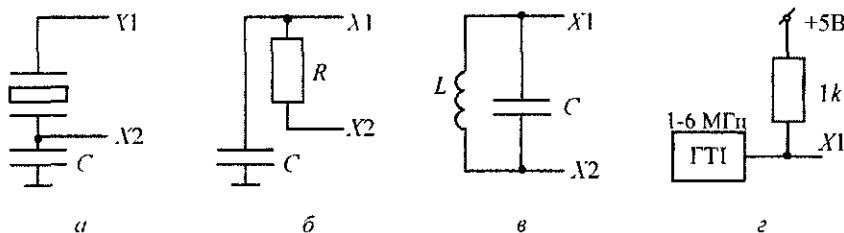


Рис. 10.14. Керування синхронізацією:

а — кварцовим резонатором; б — RC-ланцюгом; в — LC-ланцюгом; г — зовнішнім ГТТ

Внутрішній ГТТ працює і видає вихідний сигнал $SYNC$, частотою вдвоє меншою, ніж на входах $X1$ і $X2$ (рис. 10.15).



Рис. 10.15. Діаграми сигналів $X1$ і $SYNC$

Період проходження сигналів $SYNC$ визначає машинний такт T_C , з яким пов'язаний стійкий стан мікропроцесора. Команди виконуються за машинними циклами. Залежно від типу команди для її виконання потрібно один, два, три, чотири або п'ять машинних циклів (МЦ). У кожному МЦ (за виключенням циклу "Шина вільна") здійснюється звертання до пам'яті чи зовнішнього пристрою.

Мікропроцесор має сім типів можливих МЦ, які визначаються комбінацією трьох сигналів стану $S0$, $S1$, IO/M і трьох керуючих сигналів: читання RD , записування WR та підтвердження переривання $INTA$ (табл. 10.3).

Таблиця 10.3

Тип машинного циклу	Сигнали стану і керування					
	IO/M	$S1$	$S0$	\overline{RD}	\overline{WR}	\overline{INTA}
Вибірка коду операції	0	1	1	0	1	1
Читання пам'яті	0	1	0	0	1	1
Записування в пам'ять	0	0	1	1	0	1
Введення з порта	1	1	0	0	1	1
Виведення в порт	1	0	1	1	0	1
Підтвердження переривання	1	1	1	1	1	0
Шина вільна:						
команда BAD	0	1	0	1	1	1
підтвердження RST , $TRAP$	1	1	1	1	1	1
зупинка	Z	0	0	Z	Z	1

Робота процесора починається з подання сигналу скидання $RES = 0$, після чого обнуляються програмний лічильник PC , регістри PtK та FL , забороняється приймання запитів переривань. На виході CRO встановлюється сигнал лог. 1, який використовують для обнуління зовнішніх пристроїв.

При роботі мікропроцесора 8085 можливі три режими: очікування, захоплення циклу і зупинки. Режим очікування призначений для узгодження обміну інформацією між швидкодіючим процесором та повільними зовнішніми пристроями. Якщо вони не готові до обміну, то на вході RDY установлюється сигнал лог. 0 тривалістю $T_w = nT_c$, де n — довільне число тактів. Обмін продовжується за умови, що $RDY = 1$.

Режим захоплення циклу забезпечує прямий доступ до ОП від зовнішніх пристроїв. Для цього ПП виставляє на вході $HOLD$ сигнал лог. 1. Після закінчення поточного циклу мікропроцесор переключає свої шини в Z -стан і виставляє на вході $HLDA$ сигнал лог. 1 дозволу на захоплення циклу. За умови, що $HOLD = 0$, мікропроцесор продовжує свою роботу.

У режим зупинки процесор переходить після виконання команди HLT . У цьому режимі обслуговуються також запити переривання і захоплення циклу. Вихід з режиму зупинки здійснюється після скидання (початкового встановлення).

Блок керування і синхронізації (БКС) забезпечує виконання в тактах типових машинних дій, які наведені в табл. 10.4.

Таблиця 10.4

Номер такту	Типові дії в такті
T_1	Видача адреси на ША
T_2	Аналіз сигналів RDY , $HOLD$
T_w	Перехід в режими очікування при $RDY = 0$
T_3	Обмін даними. Виконання операцій
T_4, T_5	Виконання внутрішніх дій для ряду операцій

Типові машинні цикли читання і записування показані на рис. 10.16.

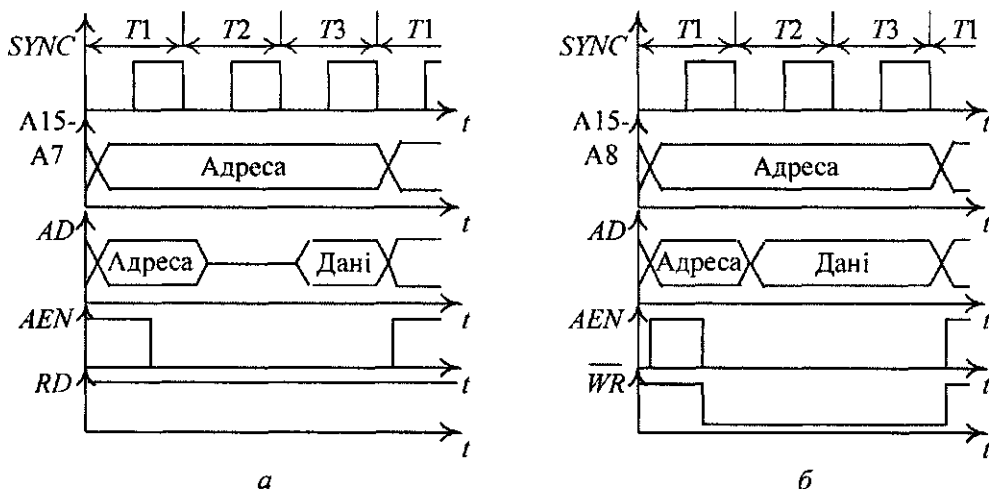


Рис. 10.16. Діаграми машинних циклів: а — читання; б — записування

При тактовій частоті $F_c = 5$ МГц тривалість такту $T_c = 0,2$ мкс.

10.2.4. Формати команд і даних. Способи адресації

Формат команд мікропроцесора залежить від типу операції і може бути одно-, дво- або трибайтовим (рис. 10.17). Перший байт містить КОП, інші — дані чи адресу операнда. Дво- і трибайтові команди зберігаються в послідовних комітках ОП. Для читання команди з пам'яті вказують адресу її першого байта.

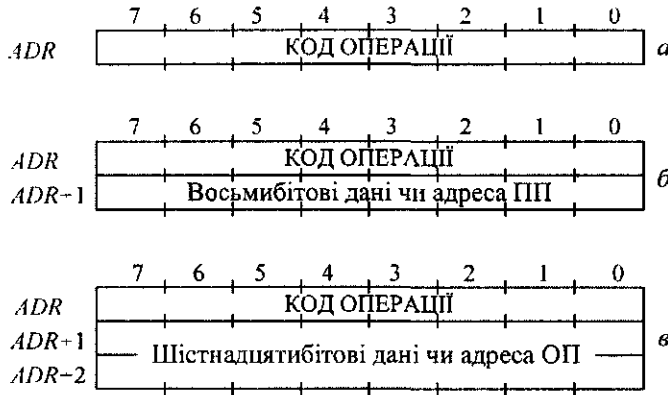


Рис. 10.17. Формат команд: а — однобайтовий; б — двобайтовий; в — трибайтовий

Символічно зміст другого байта команди записують як $\langle B2 \rangle$, третього — $\langle B3 \rangle$, а в форматі конкатенації — як $\langle B3 \rangle \langle B2 \rangle$.

У мікропроцесорі дані представляються в форматі восьмибітного двійкового слова (числа). Є команди, що оперують з 16-бітовими даними, які запам'ятовуються в двох послідовних комітках пам'яті. В арифметичних операціях мікропроцесор інтерпретує байт даних як число із знаком в доповнювальному коді. Старший біт *D7* такого числа є знаковим (0 — для додатного числа, 1 — для від'ємного). Інші сім бітів створюють значення числа.

Числа в форматі байта в доповняльному коді змінюються в діапазоні: для додатних чисел від 0 до плюс 127; для від'ємних чисел від мінус 1 до мінус 128. Від'ємні числа зберігаються в ОП в доповняльному коді. Якщо знакова інтерпретація чисел не обумовлена, то число вважається додатним в діапазоні 0–255.

В мікропроцесорі *VM85* використовують такі способи адресації операндів:

- неявний, який не вимагає операндів (наприклад, *HLT* — зупинка, *EI* — дозвіл переривання);
- прямий — адреса операнда визначається значеннями другого і третього байтів команди;
- безпосередній — операнд задається безпосередньо в команді;
- регістровий — як операнд використовують зміст регістра, вказаного в першому байті команди;
- непрямої — адреса операнда визначається змістом регістрової пари *BC*, *DE* і *HL*.

Адресація з покажчиком стека *SP* називається непрямою з автоіндексуванням.

При виконанні деяких команд застосовуються комбіновані адресації, наприклад, регістрова для акумулятора і непряма для комірки пам'яті за адресою $M = (H, L)$.

10.2.5. Мова асемблеру

Систему команд мікропроцесора описують на мові асемблеру.

Мова асемблеру — це машинно-орієнтований засіб програмування. Програма на мові асемблеру називається **початковою**. Асемблер трансліює початкову програму в символах у двійковій машинній коді. Така програма називається **об'єктною**, її можна завантажувати в пам'ять для виконання.

Програма готується у вигляді послідовності речень, які називаються операторами мови асемблеру. Кожний оператор записується в одному окремому рядку і після трансляції породжує одну машинну команду. Рядок оператора має чотири поля, в які записують мітку, мнемокод (код операції), операнд(и) та коментарі. На практиці поля оператора вирівнюються програмістом.

Усі оператори записуються за допомогою символів коду *ASCII*.

Мітки відділяються від поля мнемоніки двокрапкою:

МІТКА: *MOV A, B*; Команда пересилки

Мітки починаються з літери і відрізняються першими п'яти символами. Не можна записувати однакову мітку в декількох операторах. Але в одному операторі можна записати зразу декілька різних міток. Мітками помічають тільки ті оператори, на яких є посилання з інших операторів програми. При трансляції кожній мітці присвоюється 16-розрядна адреса першого байта поміченої команди:

JMP МІТКА; Команда переходу
MOV B, C;

.....

МІТКА: *INR A*; Помічений оператор

У полі операції записують мнемоніку команди. Мнемоніка відділяється від поля операндів як мінімум одним пробілом.

У полі операндів розміщують один або два операнди. В командах без операндів це поле вільне. Як операнди використовують імена реєстрів і реєстрових пар, мітки або безпосередні дані. Двійкові дані закінчуються символом *B*, вісімкові — *Q*, десяткові — *D* (може опускатися), шістнадцяткові — *H*; якщо в шістнадцятковому числі перший знак є літера, то перед нею пишуть цифру 0. Приклад завантаження даних:

МІТКА: *MVI A, 0EH*; шістнадцяткових
MVI B, 15; десяткових
MVI C, 15Q; вісімкових
MVI L, 1111B; двійкових

В операнд можна записувати один або два символи коду *ASCII*, які беруться в лапки. Після трансляції запису символів підставляються їхні двійкові коди.

Значення операндів можна задавати виразами, які обчислюються впродовж трансляції:

МІТКА: *MVI B, (30+40)/2*
MOV 7/2+4, 04H

У полі коментарів, яке відділяють від попереднього поля крапкою з комою, вміщують опис призначення оператора. У мові асемблеру використовують директиви, які призначені для передачі інформації, що використовується при трансляції.

До основних директив відносяться

- керування трансляцією *ORG END*,
- визначення даних та резервування в пам'яті області даних *DB, DW, DS*

Директива *ORG* (початок) використовується для встановлення початкової адреси програми

МІТКА *ORG 0800H*

При трансляції асемблер присвоює своєму лічильнику адресу значення *0800H*, за цією адресою в програмі буде розміщений перший байт команди або даних. Якщо на початку програми директива *ORG* відсутня, то лічильнику адреси присвоюється нульове значення. Якщо в директиві *ORG* є мітка, то їй присвоюється адреса комірки пам'яті, яка йде за останнім байтом об'єктного коду попереднього оператора.

Директива *END* (кінець) записується в останньому рядку початкової програми і повідомляє асемблеру про її закінчення.

МІТКА *END*

Якщо в директиві *END* записана мітка, то їй присвоюється адреса байта, який йде за останнім байтом об'єктного коду попереднього оператора.

Директиву *DB* (визначення байтів) використовують для записування в пам'ять масиву одnobайтових даних.

МІТКА *DB СПИСОК*

Операнд *СПИСОК* може бути послідовністю байтів даних, розділених комами, або символів, взятих в лапки. Якщо в директиві *DB* використовується мітка, то їй присвоюється адреса першого байта масиву даних. Використання міток у директиві *DB* дозволяє організувати символічну адресацію даних.

МІТКА1 *DB 15, 1, 24, 50*, Запам'ятати чотири байти даних

МІТКА2 *DB "Рядок"*, Запам'ятати п'ять символів

У першій директиві число 15 запам'ятовується за адресою *МІТКА1*, число 1 — за адресою *МІТКА1+1* і т. д.

Директиву *DW* (визначення слова) використовують для записування в пам'ять масиву двобайтових даних.

МІТКА *DW СПИСОК*

Операнд *СПИСОК* задає послідовність 16-розрядних даних. При виконанні директиви *DW* молодший байт першого даного запам'ятовується за поточною адресою (адреса мітки), а старший байт — за адресою на одиницю більшою. Потім ці дві виконуються для таких даних:

DATA DW 1100H, 0222H

У цьому прикладі число *1100H* запам'ятовується за адресою *DATA, DATA+1*, а число *0222H* — за адресою *DATA+2, DATA+3*. Початкову адресу може задавати директива *ORG*.

Директиву *DS* (визначення пам'яті) використовують для резервування області пам'яті для даних.

МІТКА *DS ЧИСЛО*

Значення операнда *ЧИСЛО* визначає кількість комірок (байтів) пам'яті, які резервуються для запам'ятовування даних. Зміст цієї області не визначається. Якщо в

директиві *DS* використовують мітку, то їй присвоюють адресу першого байта області пам'яті, яка резервується:

TABLE: DS 64; Резервують 64 байт даних

10.2.6. Система команд мікропроцесора *VM85*

Загальна характеристика

Система команд *VM85* містить 80 команд, які розбиті на п'ять функціональних груп:

- пересилки;
- арифметичні операції;
- логічні операції;
- передачі керування;
- керування стеком, введенням–виведенням та станом мікропроцесора.

Команди описуються символічними назвами. Кожна команда записується в одному рядку, який у загальному випадку має три поля: мнемоніки, операндів та коментарів. У полі мнемоніки записують символічну назву команди, яка замінює машинний двійковий код операції. Більшість мнемонік — це аббревіатури англійських слів, які характеризують основні функції команди, наприклад:

- *MOV (MOVE)* — передати, переслати;
- *ADD (ADD)* — додавати;
- *SUB (SUBtract)* — відняти;
- *LDA (Load Accumulator)* — завантажити акумулятор;
- *STA (Store Accumulator)* — запам'ятати акумулятор;
- *ACI (Add with Carry Immediate)* — додавання з перенесенням, безпосереднє;
- *JNZ (Jump if Non Zero)* — перейти, якщо не нуль.

Зазвичай довжина мнемоніки не перевищує чотирьох позицій, а сусіднє поле справа відділяється мінімум одним пробілом. У полі операнда записують дані — один або два операнди. Це можуть бути імена регістрів, регістрових пар, безпосередні дані чи адреси. У ряді команд операнди відсутні, наприклад, *HLT* — зупинка машини. Поле коментарів відділяється крапкою з комою, в ньому дають пояснювальний текст.

Далі система команд *VM85* описується за функціональними групами. Командам у групах дані назви команд, їхній запис у форматі мови асемблера та машинного коду, а також коментарі. Для опису команд використовується символіка:

- *R, R1, R2* — один з регістрів *A, B, C, D, E, H і L*;
- *M* — комірка пам'яті, адресу якої задає зміст регістрової *HL*-пари, тобто $M = (H, L)$;
- *RP* — одна з регістрових пар: *BC, DE, HL* чи *SP*;
- *(R)* або *(RP)* — зміст регістра чи регістрової пари;
- [*<B3><B2>*], [*<RP>*] — зміст комірки пам'яті, що адресується двома байтами пам'яті або змістом регістрової пари;
- \leftrightarrow , \leftarrow — оператори обміну і пересилки відповідно;
- *D8, D16* — безпосередні восьми- або 16-бітові дані;

- *ADR* — назва 16-бітової адреси, яка подається у другому і третьому байтах команди;
- *SSS* і *DDD* — коди регістра-джерела і регістра-приймача.

Команди пересилок

Команди пересилок мікропроцесора VM85 наведені в табл. 10.5, в якій число тактів виконання операції позначено $N_{\text{вик}}$.

Таблиця 10.5

Команда	Коментар	Машинний код	$N_{\text{вик}}$
Пересилка			
<i>MOV R1, R2</i>	$(R1) \leftarrow (R2)$	01DDDSSS	4
<i>MOV R, M</i>	$(R) \leftarrow (M)$	01DDD110	7
<i>MOV M, R</i>	$(M) \leftarrow (R)$	01110SSS	7
Безпосереднє завантаження			
<i>MVI R, D8</i>	$(R) \leftarrow D8$	01DDD110	7
		<B2>	
<i>MVI M, D8</i>	$(M) \leftarrow D8$	00110110	10
		<D8>	
<i>LXI RP, D16</i>	$(RP) \leftarrow D16$	00RP0001	10
		<B2>	
		<B3>	
Завантаження акумулятора			
<i>LDA ADR</i>	$(A) \leftarrow (ADR)$	00111010	13
		<B2>	
		<B3>	
<i>LDAX RP</i>	$(A) \leftarrow [(RP)]$	00RP1010	7
Запам'ятовування акумулятора			
<i>STA ADR</i>	$(ADR) \leftarrow (A)$	00110010	13
		<B2>	
		<B3>	
<i>STAX RP</i>	$[(RP)] \leftarrow (A)$	00RP0010	7
Пряме завантажування HL-пари			
<i>LHLD ADR</i>	$(L) \leftarrow [(ADR)]$ $(H) \leftarrow [(ADR+1)]$	00101010	16
		<B2>	
		<B3>	
Запам'ятовування HL-пари			
<i>SHLD ADR</i>	$[\langle B3 \rangle \langle B2 \rangle] \leftarrow (L)$ $[\langle B3 \rangle \langle B2 \rangle + 1] \leftarrow (H)$	00100010	16
		<B2>	
		<B3>	
Обмін змістом HL і DE пар			
<i>XCHG</i>	$(H) \leftrightarrow (D)$ $(L) \leftrightarrow (E)$	11101011	4

Команди пересилок виконують такі дії:

- обмін даними між регістрами $MOV R1, R2$, регістрами і пам'яттю $MOV M, R$, або $MOV R, M$;
- безпосереднє завантаження регістрів $MVI R, D8$, комірки пам'яті $MVI M, D8$, регістрової пари $LXI RP, D16$;
- пряме завантаження $LDA ADR$ і запам'ятовування $STA ADR$ акумулятора;
- непряме завантаження $LDA RP$ і запам'ятовування $STAX RP$ акумулятора;
- пряме завантаження $LHLD ADR$ і запам'ятовування $SHLD ADR$ регістрової HL -пари;
- обмін даними $HCHG$ регістрових HL і DE пар.

Команди пересилок не змінюють значень прапорців у регістрі ознак.

Приклади команд пересилок:

$MOV A, B$; Переслати зміст регістра B в акумулятор A ;

$MVI A, 15H$; Записати в акумулятор число $15H$;

$MVI H, 10/2+2$; Записати в регістр H число 7 ;

$LXI H, 1000H$; Завантажити в регістрову HL -пару адресу $1000H$;

$MOV M, A$; Запам'ятати за адресою $1000H$ зміст акумулятора, якщо $(H, L) = 1000H$;

$STA 2000H$; Запам'ятати за адресою $2000H$ зміст акумулятора.

Команди арифметичних операцій мікропроцесора 8085

Команди арифметичних операцій мікропроцесора 8085 наведені в табл. 10.6.

Таблиця 10.6

Команда	Коментар	Машинний код	$N_{\text{БК}}$
Додавання з акумулятором			
$ADD R$	$(A) \leftarrow (A) + (R)$	1 0 0 0 0 S S S	4
$ADD M$	$(A) \leftarrow (A) + (M)$	1 0 0 0 0 1 1 0	7
$ADI D8$	$(A) \leftarrow (A) + D8$	1 1 0 0 0 1 1 0	7
		<B2>	
Додавання з акумулятором з перенесенням			
$ADC R$	$(A) \leftarrow (A) + (R) + (CY)$	1 0 0 0 1 S S S	4
$ADC M$	$(A) \leftarrow (A) + (M) + (CY)$	1 0 0 0 1 1 1 0	7
$ACI D8$	$(A) \leftarrow (A) + D8 + (CY)$	1 1 0 0 1 1 1 0	7
		<B2>	
Віднімання з акумулятора			
$SUB R$	$(A) \leftarrow (A) - (R)$	1 0 0 1 0 S S S	4
$SUB M$	$(A) \leftarrow (A) - (M)$	1 0 0 1 0 1 1 0	7
$SUI D8$	$(A) \leftarrow (A) - D8$	1 1 0 1 0 1 1 0	7
		<B2>	
$SBB R$	$(A) \leftarrow (A) - (R) - (CY)$	1 0 0 1 1 S S S	4
$SBB M$	$(A) \leftarrow (A) - (M) - (CY)$	1 0 0 1 1 1 1 0	7
$SBI D8$	$(A) \leftarrow (A) - D8 - (CY)$	1 1 0 1 1 1 1 0	7
		<B2>	7

Таблиця 10.6. Продовження

Команда	Коментар	Машинний код	$N_{\text{вик}}$
Інкремент			
<i>INR R</i>	$(R) \leftarrow (R) + 1$	00 D D D 1 0 0	5
<i>INR M</i>	$(M) \leftarrow (M) + 1$	0 0 1 1 0 1 0 0	10
<i>INX RP</i>	$(RP) \leftarrow (RP) + 1$	0 0 R P 0 0 1 1	5
Декремент			
<i>DCR R</i>	$(R) \leftarrow (R) - 1$	0 0 D D D 1 0 1	5
<i>DCR M</i>	$(M) \leftarrow (M) - 1$	0 0 1 1 0 1 0 1	10
<i>DCX RP</i>	$(RP) \leftarrow (RP) - 1$	0 0 R P 1 0 1 1	5
Подвійне додавання регістрової пари			
<i>DAD RP</i>	$(HL) \leftarrow (HL) + (RP)$	0 0 R P 1 0 0 1	10
Десяткова корекція акумулятора			
<i>DAA</i>		0 0 1 0 0 1 1 1	4

Арифметичні команди виконують операції додавання, віднімання, інкремента, декремента та ін.

Операції додавання і віднімання виконуються у доповняльному коді над змістом акумулятора і операнда. Результат операції зберігається в акумуляторі і залежно від нього встановлюються всі прапорці. Операнди представляються у форматі цілого числа із знаком в доповняльному коді в діапазоні мінус 128 – плюс 127. При цьому старший біт є знаковим.

За командами додавання *ADD R*, *ADD M*, *ADI D8* зміст вказаного регістра *R*, комірки пам'яті за адресою *M* чи безпосередні дані додаються до змісту акумулятора і результат записується в акумулятор. Команди додавання *ADC R*, *ADC M*, *ACI D8* виконують додавання з урахуванням значення біта перенесення (*CY*).

Приклад виконання команди додавання *ADD C*:

Акумулятор	$(A) = \underline{1111\ 0001} = -15_{10}$
Регістр	$(C) = 0101\ 1100 = +92_{10}$
Результат	$(A) = 0100\ 1101 = 77_{10}$
Прапорці:	$CY = 1; AC = 0; S = 0; P = 1; Z = 0.$

За командами віднімання *SUB R*, *SUB M*, *SUI D8* зміст вказаного регістра *R* чи комірки пам'яті за адресою або безпосередні дані віднімаються від змісту акумулятора і результат записують в акумулятор. За командами віднімання *SBB R*, *SBB M*, *SBI D8* ці ж дії виконуються з урахуванням значення біта позики (*CY*). Якщо при відніманні позика є (тобто зменшуване менше від'ємника), то із старшого розряду суматора перенесення не буде і біт *CY* устанавлюється в стан "1". Якщо ж це перенесення було (позики не треба), то *CY* біт скидається в стан "0".

Приклад виконання команди віднімання *SUB H*:

Акумулятор	$(A) = \underline{1,111\ 0001} = -15_{10}$
Регістр	$(H) = \underline{1,111\ 1110} = -2_{10}$
Результат	$(A) = \underline{1,111\ 0011} = -13_{10}$
Прапорці:	$CY = 0; AC = 0; S = 1; P = 1.$

Команди додавання і віднімання із врахуванням перенесень (позики) дозволяють організувати обробку багатобайтових чисел послідовно за байтами, які зберігаються в суміжних комірках пам'яті й адресуються, починаючи з молодшого байта

Для збільшення або зменшення змісту регістрів чи комірки пам'яті на одиницю використовуються відповідно команди інкремента $INR R$, $INR M$ та декремента $DCR R$, $DCR M$. Ці операції відносно до результатів операції встановлюють прапорці S , Z , P , AC , біт перенесення (позики) не змінюється

Команди інкременту $INX RP$ або декремента $DCX RP$ відповідно збільшують або зменшують на одиницю зміст регістрових пар RP

Команда подвійного додавання $DAD RP$ підсумовує зміст двох регістрових пар фіксованої HL та RP , що вказується в команді. Результат записується в HL -парі, встановлюється тільки прапорець CY , інші ознаки не змінюються

Приклад виконання команди $DAD BC$:

Регістрова пара (HL)	$= 0000\ 1100\ 0000\ 0000 = 3072_{10}$
Регістрова пара (BC)	$= +1111\ 1111\ 1111\ 1111 = -1_{10}$
Результат (HL)	$= 0000\ 1011\ 1111\ 1111 = 3071_{10}$

У форматі байта можна записати дві двійково-кодовані десяткові цифри (BCD -числа). При додаванні BCD -чисел операнди підсумовуються як двійкові числа. Потім проміжний результат коректується командою DAA за правилами

- якщо молодша тетрада більш за дев'ять або встановлений прапорець $AC = 1$, то зміст акумулятора збільшується на шість і до старшої тетради додається одиниця,
- якщо після цього старша тетрада акумулятора більш за дев'ять, $CY = 1$, то до неї додають цифру 6

Зазначимо, що в командах додавання і віднімання результат в акумулятор записується в такті $T2$ циклу $M1$ нової команди, це підвищує продуктивність мікропроцесора

Команди логічних операцій мікропроцесора $BM85$

Команди логічних операцій мікропроцесора $BM85$ наведені в табл. 10.7

Таблиця 10.7

Команда	Коментар	Машинний код	$N_{\text{вик}}$
Логічне І з акумулятором			
$ANA R$	$(A) \leftarrow (A) \wedge (R)$	1 0 1 0 0 S S S	4
$ANA M$	$(A) \leftarrow (A) \wedge (M)$	1 0 1 0 0 1 1 0	7
$ANI D8$	$(A) \leftarrow (A) \wedge D8$	1 1 1 0 0 1 1 0	7
		<B2>	
Логічне ЧИ з акумулятором			
$ORA R$	$(A) \leftarrow (A) \vee (R)$	1 0 1 1 0 S S S	4
$ORA M$	$(A) \leftarrow (A) \vee (M)$	1 0 1 1 0 1 1 0	7
$ORI D8$	$(A) \leftarrow (A) \vee D8$	1 1 1 1 0 1 1 0	7
		<B2>	

Таблиця 10.7. Продовження

Команда	Коментар	Машинний код	$N_{\text{вик}}$
“Виключальне ЧИ” з акумулятором			
<i>XRA R</i>	$(A) \leftarrow (A) \oplus (R)$	1 0 1 0 1 S S S	4
<i>XRA M</i>	$(A) \leftarrow (A) \oplus (M)$	1 0 1 0 1 1 1 0	7
<i>XRI D8</i>	$(A) \leftarrow (A) \oplus D8$	1 1 1 0 1 1 1 0	7
		<B2>	
Порівняння з акумулятором			
<i>CMP R</i>	$(A) - (R)$	1 0 1 1 1 S S S	4
<i>CMP M</i>	$(A) - (M)$	1 0 1 1 1 1 1 0	7
<i>CPI D8</i>	$(A) - D8$	1 1 1 1 1 1 1 0	7
		<B2>	
Зсув акумулятора			
<i>RLC</i>	$A_m \leftarrow A_{m-1}$	0 0 0 0 0 1 1 1	4
<i>RRC</i>	$A_m \rightarrow A_{m-1}$	0 0 0 0 1 1 1 1	4
Зсув акумулятора з перенесенням			
<i>RAL</i>		0 0 0 1 0 1 1 1	4
<i>RAR</i>		0 0 0 1 1 1 1 1	4
Інверсія			
<i>CMA</i>	$(A) \leftarrow (\overline{A})$	0 0 1 0 1 1 1 1	4
<i>CMC</i>	$(CY) \leftarrow (\overline{CY})$		4
Встановлення перенесення в положення “1”			
<i>STC</i>	$(CY) \leftarrow 1$	0 0 1 1 0 1 1 1	4

Логічні команди виконують порозрядні операції І, ЧИ, “Виключальне ЧИ”, а також операції порівняння, зсуву, інвертування акумулятора і біта перенесення, встановлення біта перенесення.

За командами логічного І (*ANA R*, *ANA M*, *ANI D8*) або логічного ЧИ (*ORA R*, *ORA M*, *ORI D8*) та виключального ЧИ (*XRA R*, *XRA M*, *XRI D8*) виконуються відповідні порозрядні логічні операції над змістом:

- акумулятора і вказаного регістра *R*;
- акумулятора і комірки пам'яті за адресою *M*;
- акумулятора і безпосередніх даних *D8*.

Результат записується в акумулятор.

Приклади логічних порозрядних операцій І, ЧИ, “Виключальне ЧИ”

Зміст акумулятора і регістра *B* до операції: $(A) = 10011011$, $(B) = 11001101$.
Результати операції:

ANA B; $(A) = 10001001$, $S = 1$, $Z = 0$, $P = 0$

ORA B; $(A) = 11011111$, $S = 1$, $Z = 0$, $P = 0$

XRA B; $(A) = 01010110$, $S = 0$, $Z = 0$, $P = 1$

За командами порівняння *CMP R*, *CMP M*, *CPI D8* із змісту акумулятора віднімається операнд, але результат операції не зберігається; при цьому значення даних,

що порівнюються, не змінюються і за правилами віднімання встановлюються всі прапорці.

В командах порівняння дані інтерпретуються як числа в доповняльному коді в діапазоні мінус 128 – плюс 127. Результат порівняння однобайтових чисел із знаками трактується так:

$$\text{Якщо } \left\{ \begin{array}{l} (A) = (R) \\ (A) \neq (R) \\ (A) > (R) \\ (A) < (R) \end{array} \right\}, \text{ маємо } \left\{ \begin{array}{l} CY = 0, Z = 1 \\ Z = 0 \\ CY = 0 \\ CY = 1 \end{array} \right\}.$$

Коли дані, які порівнюються, розглядаються як додатні числа без знака від нуля до 255 і їхні старші біти різні, то програміст має трактувати результат порівняння так:

$$\text{Якщо } \left\{ \begin{array}{l} (A) > (R) \\ (A) < (R) \end{array} \right\}, \text{ маємо } \left\{ \begin{array}{l} CY = 0 \\ CY = 1 \end{array} \right\}.$$

Команди зсуву призначені для циклічного зсуву змісту акумулятора на один розряд вліво (команди *RLC*, *RAL*) або вправо (команди *RRC*, *RAR*). Команди *RAL* та *RAR* включають у полі зсуву прапорець перенесення (рис. 10.18.)

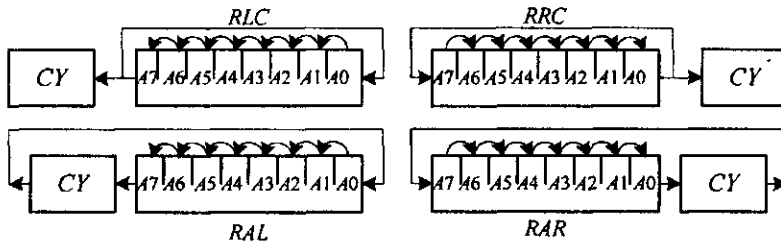


Рис. 10.18. Виконання операцій зсуву

До логічних команд відносять команди *CMA* — інвертування змісту акумулятора, *STC* — встановлення в положення "1" прапорця перенесення *CY* та його інвертування *CMC*.

Команди передачі керування мікропроцесора 8085

Команди передачі керування мікропроцесора 8085 наведені в табл. 10.8.

Таблиця 10.8

Команда	Коментар	Машинний код	$N_{\text{вик}}$
Безумовний перехід			
<i>JMP ADR</i>	$(PC) \leftarrow ADR$	11000011	10
		<B2>	
		<B3>	
Умовний перехід			
<i>JMP COND ADR</i>	Якщо <i>CCC</i> , то $(PC) \leftarrow B3 > B2$, $(PC) \leftarrow (PC) + 3$	11CCC100	10
		<B2>	
		<B3>	

Таблиця 10.8. Продовження

Команда	Коментар	Машинний код	$N_{\text{вик}}$
Безумовний виклик підпрограми			
<i>C ALL ADR</i>	$(PC) \leftarrow ADR$	11001101	17
		<B2>	
		<B3>	
Умовний виклик підпрограми			
<i>C COND ADR</i>	Якщо <i>CCC</i> , то $(PC) \leftarrow \langle B3 \rangle \langle B2 \rangle$, $(PC) \leftarrow (PC) + 3$	11CCC101	17
		<B2>	
		<B3>	
<i>RET</i>	$(PC) \leftarrow (STACK)$	11001001	11
Повернення за умовою			
<i>R COND</i>	Якщо <i>CCC</i> , то $(PC) \leftarrow (STACK)$ $(PC) \leftarrow (PC) + 1$	11CCC000	17/11
Непрямий перехід			
<i>PCHL</i>		11101001	5
Повторний запуск (рестарт)			
<i>RST n</i>		11AAA111	

Команди передачі керування змінюють природну послідовність виконання команд програми (табл. 10.8.), але не змінюють значень прапорців. Розрізняють безумовні та умовні команди передачі керування.

За командою *JMP ADR* виконується безумовна передача керування за адресою <B3><B2>, яка завантажується в програмний лічильник *PC*; старий зміст лічильника втрачається. Наприклад:

JMP 4040H; Перехід за адресою 4040H

При виконанні умови здійснюється перехід за новою адресою, якщо інакше, то виконується наступна команда. Для зручності програмування передбачені перевірки умов за одиничним і нульовим значеннями прапорців *CY*, *Z*, *S*, *P*. Мнемоніка команд умовного переходу створюється додаванням букви *J* до назви умови: *JC* — перехід, якщо *CY* = 1; *JNC* — перехід, якщо *NC* = 1 і т.д. Наприклад:

JNC 4040H; Перехід, якщо *NC* = 1

В умовних командах передачі керування перевіряється поточне значення умови, яке вказується в трибітовому полі *CCC* коду операції (табл. 10.9).

Таблиця 10.9

Умова	Назва умови	Код <i>CCC</i>	Мнемоніка		
			перехід	виклик	повернення
<i>CY</i> =1	Перенесення	011	<i>JC</i>	<i>CC</i>	<i>RC</i>
<i>NC</i> =1	Немає перенесення	010	<i>JNC</i>	<i>CNC</i>	<i>RNC</i>
<i>Z</i> =1	Нульовий результат	001	<i>JZ</i>	<i>CZ</i>	<i>RZ</i>
<i>NZ</i>	Ненульовий результат	000	<i>JNZ</i>	<i>CNZ</i>	<i>RNZ</i>
<i>M</i>	Від'ємний результат	111	<i>JM</i>	<i>CM</i>	<i>RM</i>
<i>P</i>	Додатний результат	110	<i>JP</i>	<i>CP</i>	<i>RP</i>
<i>PE</i>	Парний результат	101	<i>JPE</i>	<i>CPE</i>	<i>RPE</i>
<i>PO</i>	Непарний результат	100	<i>JPO</i>	<i>CPO</i>	<i>RPO</i>

За командою *CALL ADR* виконується безумовний виклик підпрограми за адресою $\langle B3 \rangle \langle B2 \rangle$, яка завантажується в програмний лічильник. Зміст *PC* тимчасово запам'ятовується в стеку як адреса повернення. Наприклад:

CALL 4040H; Виклик підпрограми за адресою 4040H

Команди умовного виклику підпрограм передають керування за адресою $\langle B3 \rangle \langle B2 \rangle$ тільки при виконанні умови, записаної в полі *CCC* коду операції. При цьому адреса повернення запам'ятовується в стеку. Якщо задана умова не виконується, то виконується наступна за послідовністю команда. Мнемоніка команди умовного виклику підпрограми будується додаванням літери *C* до назви умови (табл. 10.9); *CC* — виклик, якщо $CY = 1$; *CNC* — виклик, якщо $NC = 0$ і т.д.

Команда безумовного повернення з програми передає керування за адресою повернення, яка зберігається в двох верхніх комірках стеку. Команди умовного повернення з підпрограм виконуються при виконанні умови в полі *CCC* (табл. 10.9).

Команда повторного запуску (рестарт) *RSTn* призначена для обробки переривання і введення контрольних точок (розривів) при налагодженні програм. У коді операції рестарт 11AAA111 три біти $n = AAA$ називаються вектором; їх задає програміст або схема переривання. За командою *RSTn* поточний зміст програмного лічильника *PC* засилається у стек, а в лічильник завантажується код $PC \leftarrow 0000\ 0000\ 00AA\ A000 = 8 \cdot n$.

Таким чином, залежно від значень вектора *AAA* реалізуються передачі керування за адресою: *RST 0* → 0000H; *RST 1* → 0008H; *RST 7* → 0038H. Зазвичай у восьми байтах, які відводяться для кожної команди, записують початкову адресу підпрограми обслуговування переривань. Вихід з підпрограм обслуговування переривань виконується за командою повернення *RET*.

Команди *CALL* і *C COND* (при виконанні умови) виконуються за п'ять циклів і 17 тактів: якщо умова виклику не задовольняється, то ці команди реалізуються за три цикли і 11 тактів. Команди *JMP* і *J COND* виконуються за три цикли і 11 тактів. Команди *RET* і *R COND* (при виконанні умови) реалізуються за три цикли і п'ять тактів. Команда *RST* виконується за три цикли і 11 тактів.

Команди керування введенням-виведенням, стеком і станами мікропроцесора 8085

Команди керування введенням-виведенням, стеком і станами мікропроцесора 8085 наведені в табл. 10.10.

Таблиця 10.10

Команда	Коментар	Машинний код	$N_{\text{вик}}$
Введення даних з порту			
<i>IN PORT</i>	$(A) \leftarrow (PORT)$	11011011	10
		$\langle B2 \rangle$	
Виведення даних в порт			
<i>OUT PORT</i>	$(PORT) \leftarrow (A)$	11010011	10
		$\langle B2 \rangle$	
Записування у стек			
<i>PUSH RP</i>	$(STACK) \leftarrow (RP)$	11RP0101	11

Таблиця 10.10. Продовження

Команда	Коментар	Машинний код	$N_{\text{внк}}$
Читання стеку			
<i>POP RP</i>	$(RP) \leftarrow (STACK)$	11RP0001	10
Обмін із стеком			
<i>XTHL</i>	$(L) \leftrightarrow (SP) \quad (H) \leftrightarrow ((SP)+1)$	11100011	17
Завантаження <i>SP</i>			
<i>SPHL</i>	$(SP) \leftarrow (HL)$	11111001	5
Дозвіл переривання			
<i>EI</i>	$INTE \leftarrow 1$	11111011	4
Заборона переривання			
<i>DI</i>	$INTE \leftarrow 0$	11110011	4
Відсутність операцій			
<i>NOP</i>		00000000	4
Зупинка			
<i>HLT</i>		01110110	7
Читання маски			
<i>RIM</i>	$(A) \leftarrow (IM)$	00100000	4
Встановлення маски			
<i>SIM</i>	$(IM) \leftarrow (A)$	00110000	4

Для обміну даними з процесором зовнішні пристрої мають на своїх входах і виходах регістри, які називаються **портами введення–виведення**.

За командою *IN PORT* байт даних з порту зовнішнього пристрою пересилається в акумулятор. За командою *OUT PORT* пересилається байт даних з акумулятора в зовнішній порт; *PORT* — це символічне позначення адреси порту, яка знаходиться в діапазоні 0–255.

За командою *PUSH RP* зміст однієї з регістрових пар *BC*, *DE*, *HL* чи *PSW*, яка адресується кодом *RP* в полі команди, записується в стек. У комірку пам'яті за адресою $(SP)-1$ пересилається зміст старшого регістра пари, а за адресою $(SP)-2$ — зміст молодшого регістра пари. Зміст цього значення стеку *SP* зменшується на два.

За командою *POP RP* два сусідніх байти з вершини стеку пересилаються до однієї з регістрових пар *BC*, *DE*, *HL* чи *PSW*, які адресуються кодом у полі *RP*. В молодший регістр пари завантажуються зміст комірки з адресою $[SP]$, а в старший — з адресою $[(SP)+1]$. Після цього значення покажчика стеку *SP* збільшується на два.

За командою *SPHL* зміст *HL*-пари переписується в покажчик стека *SP*.

За командою *XTHL* виконується обмін змістом між *HL*-парою та двома верхніми комірками стеку, які адресуються *SP* (рис. 10.19).

Програмне керування запитами переривань від зовнішніх пристроїв здійснюється командами *EI* (дозвіл переривання) і *DI* (заборона переривання). За командою *NOP* не виконується операція, оскільки вона використовується для формування часових затримок або для резервування місця в програмі. За цією командою зміст лічильника *PC* збільшується на одиницю і, таким чином, вказує на адресу наступної команди. За командою зупинки *HLT* значення *PC* збільшується на одиницю і мікропроцесор переходить в режим зупинки до появи запиту на переривання від зовнішнього пристрою. Запит на переривання обслуговується, якщо командою *EI* до цього

був встановлений внутрішній тригер переривання; після цього мікропроцесор знову переходить в режим зупинки. Для продовження звичайної роботи необхідно виконати скидання мікропроцесора. Усі команди керування не змінюють значень прапорців.

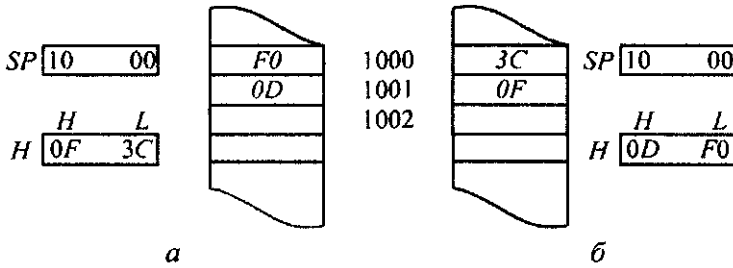


Рис. 10.19. Виконання команди XTHL: а — до виконання команди; б — після виконання команди

За командою *SIM* зміст акумулятора пересилається в регістр маски переривання і одночасно значення старшого розряду акумулятора передається в однобітний порт виводу *SOD*. За командою *RIM* в акумулятор пересилається зміст маски переривання і одночасно в старший розряд акумулятора передається значення послідовного порту введення.

10.3. Однокристальні шістнадцятирозрядні мікропроцесори

Однокристальні 16-розрядні мікропроцесори мають: розширену систему команд (в тому числі множення, ділення одно- і двобайтних чисел); збільшений адресний простір до 1–16 Мбайт; додаткові способи адресації.

Відомі такі мікропроцесори даного класу: K1810BM86 (*i8086*), K1810BM88 (*i8088*), Z-800 (фірми *Zilog*), 680XO (фірми *Motorola*) та інші. Розглянемо детально мікропроцесор K1810BM86A.

10.3.1. Загальна характеристика мікропроцесора K1810BM86A

Мікропроцесор K1810BM86A (далі *BM86*) використовують як центральний мікропроцесор в керуючих, інформаційних та контрольно-вимірюючих МПС. Мікропроцесор *BM86* (аналог *i8086*) має такі характеристики:

- схемотехнологія — *n*-МОП, розмір кристала — 5,5 × 5,5 мм, містить близько 29000 транзисторів;
- однофазна синхронізація імпульсами частотою 5 МГц від зовнішнього генератора;
- випускається в корпусі типу *DIP* з 40 виводами;
- джерело живлення — плюс 5 В, споживана потужність — 1,7 Вт;
- число базових команд — 111;
- кількість операцій додавання типу "регістр" — близько 1,6 млн/с.

Структурна схема мікропроцесора *BM86* (рис. 10.20) має арифметико-логічний пристрій, шинний інтерфейс та пристрій керування і синхронізації.

Пристрій керування і синхронізації (ПКС) має у своєму складі такі схеми:

- буфера адреси/стану (БАС);
- оброблення запитів на переривання;
- керування прямим доступом до пам'яті;
- вироблення сигналів записування і читання пам'яті чи зовнішніх портів введення-виведення;
- забезпечення внутрішньої синхронізації.

Регістри черги команд $R1$ – $R6$ є восьмирозрядні, а всі інші — шістнадцятирозрядні.

Функції арифметико-логічного пристрою

Арифметико-логічний пристрій призначений для зберігання даних і виконання операцій над ними. Блок АЛБ реалізує арифметико-логічні операції, в тому числі зсуви над восьми- і шістнадцятирозрядними операндами.

Регістри загального призначення (або регістри даних) AX , BX , CX , DX використовуються для зберігання операндів і результатів операцій. Вони програмно доступні й використовуються в усіх арифметико-логічних операціях. У багатьох командах регістри даних мають свої специфічні призначення: AX — акумулятор; BX — регістр бази; CX — лічильник циклів; DX — регістр даних. Кожний з регістрів даних складається з двох частин, до яких можна звертатися за іменами AH , BH , CH , DH (старший байт — *High*) і AL , BL , CL , DL (молодший байт — *Low*). Це забезпечує обробку байтів даних, а також програмну сумісність з мікропроцесорами $VM80$, $VM85$.

Регістри-показчики SP , BP та індексні регістри використовують у арифметико-логічних операціях та при визначенні адрес операндів. Показчики стека SP і бази BP містять відносні адреси в границях сегмента стекової пам'яті. Індексні регістри SI та DI забезпечують автоінкрементну адресацію в операціях з рядками даних.

У регістрі ознак FL використовують тільки дев'ять бітів, причому п'ять ознак еквівалентні ознакам мікропроцесорів $VM80$, $VM85$: SF — знак результату; ZF — нульовий результат; AF — перенесення між тетрадами молодшого байта; PF — парний результат; CF — перенесення із старшого розряду результату (рис. 10.21).

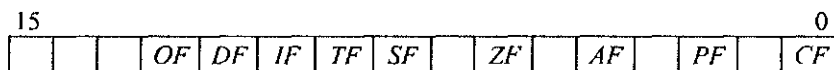


Рис. 10.21. Регістр ознак

До нових ознак відносяться: TF — дозвіл крокового режиму (трасування); IF — дозвіл або заборона маскованих переривань; DF — керування напрямком автоадресації: $DF = 1$ — індексні регістри автоматично декрементуються, $DF = 0$ — інкрементуються; OF — переповнення розрядної сітки.

Функції блоку інтерфейса з шиною

Даний блок здійснює зв'язки з модулями МПС, вибірку команд і даних з пам'яті, формує чергу команд. Шість восьмирозрядних регістрів черги команд $R1$ – $R6$ забезпечують тимчасове зберігання шести байтів кодів команд. Блок інтерфейсу слідує за станом черги і при необхідності поповнює її, коли інші модулі системи не займа-

ють пам'яті. При виконанні команд передачі керування черга скидається і після завершення переходу заповнюється знову.

Мікропроцесор 8086 може адресувати пам'ять ємністю до 1 Мбайт. Адресний простір логічно розбивається на сегменти по 64 Кбайт кожний. Початкова адреса сегмента розміщується в сегментних регістрах:

CS — регістр програмного сегмента пам'яті; його зміст разом зі змістом покажчика задає адресу наступної команди; DS — регістр сегмента даних; SS — регістр сегмента стека; ES — додатковий сегментний регістр даних.

Таким чином, зміст сегментного регістра — це базова адреса сегмента.

Фізичну адресу (ФА) пам'яті, яка поступає на 20-розрядну ША, одержують додаванням двох частин: адреси необхідного сегмента (Seg), яка збільшується на чотири вліво (множиться на 16), та виконавчої (ефективної) адреси (EA):

$$ФА = 16 \cdot Seg + EA.$$

Формування ФА в блоці ШІФ показано на рис. 10.22.

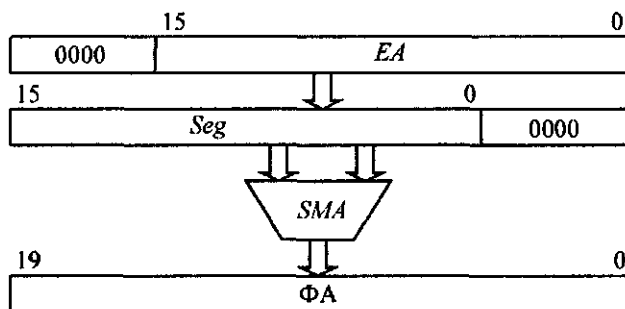


Рис. 10.22. Формування ФА

Виконавча адреса EA може бути константою (зміщенням в команді), змістом регістра чи комірки пам'яті або сумою декількох значень (наприклад, двох регістрів і константи), але ця сума є 16-розрядною (перенесення ігнорується). Таким чином, значення ФА ніколи не переходить границі 64 Кбайт. Сегментація пам'яті по 64 Кбайт підвищує захист програмних файлів один від одного, але незручна при написанні великих програм. У 32-розрядних мікропроцесорах обмеження на об'єм сегментів пам'яті зняті.

Мікропроцесор може звертатися як до одного байта пам'яті, так і до слова з двох байтів або до подвійного слова — з чотирьох байтів.

Пристрій керування і синхронізації

При описуванні мікросхеми 8086 назва виводу (контакту, лінії) збігається з назвою сигналу, що передається по ньому.

Для збільшення комунікаційних можливостей мікропроцесора використовують:

- мультиплексування в часі, при чому сигнали на одному і тому ж виводі мають різне функціональне призначення;
- зміну конфігурації;
- двонаправлені виводи обміном даних;
- третій стан (Z -стан), в який переводиться частина виводів у режимі захвату локальної шини і оброблення переривань.

Мікросхема має вивід MN/\overline{MX} для вибору конфігурації: мінімальної – вивід підключається до джерела живлення U_{CC} і максимальної — вивід з'єднують з шиною землі. Мінімальна конфігурація використовується в простих системах, при цьому мікропроцесор **BM86** сам виробляє всі необхідні керуючі сигнали. Максимальну конфігурацію вибирають при побудові складних МПС, а керуючі сигнали одержують за допомогою зовнішніх контролерів.

Сукупність ліній, які зв'язують мікропроцесор з буферними регістрами адреси, формувачами даних та контролерами, створюють **локальну шину**. Сигнали з виходів указаних модулів створюють **системну шину (магістраль)**.

Цикл обміну інформацією з магістраллю складається з чотирьох машинних тактів $T1-T4$ (в деяких командах вводиться такт $T5$). У такті $T1$ на локальну шину $A_{D15-A_{D0}}$, $A_{19-A_{16}}$ видається адреса пам'яті або порта, потім шина переключається на обмін даними в циклі записування (такти $T2, T3, T4$), або в циклі читання (такти $T3, T4$). Для узгодження з повільно діючою пам'яттю або з портами між тактами $T3$ і $T4$ можуть вводиться додаткові такти очікування T_W , протягом яких дані в магістралі не змінюються.

Частина сигналів мікропроцесора **BM86** аналогічна із сигналами мікропроцесорів **BM80** і **BM85**: CLK — синхронізація; RES — скидання; RDY — готовність до обміну; \overline{RD} — читання; \overline{WR} — записування; ALE — строб адреси; $DEN, DT/R, HOLD$ і $HOLDA, INT$ і $INTA, M/IO$.

У мікросхемі мікропроцесора **BM86** введені нові виводи і сигнали:

- \overline{BHE} — дозвіл на передачу старшого байта адреси;
- \overline{MNI} — вхід сигналу немаскованого переривання;
- \overline{TEST} — стан сигналу переривання \overline{WAIT} (при $\overline{TEST} = 0$ мікропроцесор переходить у стан очікування до появи логічного нуля, використовується для узгодження роботи із співпроцесором);
- \overline{LOCK} — блокування системної шини (при виконанні мікропроцесором команди з префіксом \overline{LOCK} на виводі \overline{WR} формується логічний нуль, який забороняє доступ до магістралі інших модулів МПС до закінчення команди).

На двофункціональних виводах після закінчення основних сигналів формується інформація про тип циклу S_2, S_1, S_0 : 000 — підтвердження переривання; 001/010 — записування / читання ПП; 011 — зупинка; 101/110 — записування / читання пам'яті. Сигнали S_3, S_4 вказують на сегментний регістр, який використовується в даному циклі: 00 — ES , 01 — SS , 10 — CS , 11 — DS .

Система команд мікропроцесора **BM86**

Мікропроцесор має 133 команди, які складають п'ять функціональних груп:

- пересилки даних типу регістр-регістр, регістр-пам'ять, пам'ять-пам'ять, циклічні передачі, введення-виведення;
- арифметичні: додавання, віднімання, порівняння і корекції;
- логічні: порозрядні диз'юнкції, кон'юнкції, за модулем два, інверсії, арифметичні та логічні зсуви вправо чи вліво;
- передачі керування і безумовного і умовного переходу, визову підпрограм та повернення з них, багатократного виконання, програмні та зовнішні переривання;

- керування мікропроцесором: зупинка, встановлення і скидання прапорців, підтримка функціонування схеми в мультипроцесорних системах.

Мнемоніка більшості команд мікропроцесора 8086 збігається з символікою команд мікропроцесорів 8080 та 8085. Команди мають довжину 1–6 байт. Код операції міститься в одному або в двох перших байтах. Розряд W в першому байті команди вказує на тип операнда: $W = 0$ — 1 байт; $W = 1$ — 2 байт (слова): У двомісних командах джерело даних (правий операнд) може бути константою, тоді приймач даних (лівий операнд) є регістром або пам'яттю. У всіх інших випадках один з двох операндів міститься в регістрі, а другий — в регістрі або пам'яті.

Вид адресації задають у другому байті (постбайті), який має три поля: методу адресації MOD , регістра REG і регістра або пам'яті R/M . Наступні один або два байти в команді можуть містити зміщення ($DISP$) в команді або безпосередні дані. Зміщення подається восьмирозрядним ($DISP8$) чи 16-розрядним ($DISP16$) числом із знаком в доповняльному коді (рис. 10.23, а, б). Дані також записуються у виді байта $DATA8$ або слова $DATA16$ (рис. 10.23, в, г). У командах із шести байтів можуть бути послідовно записані зміщення і дані.

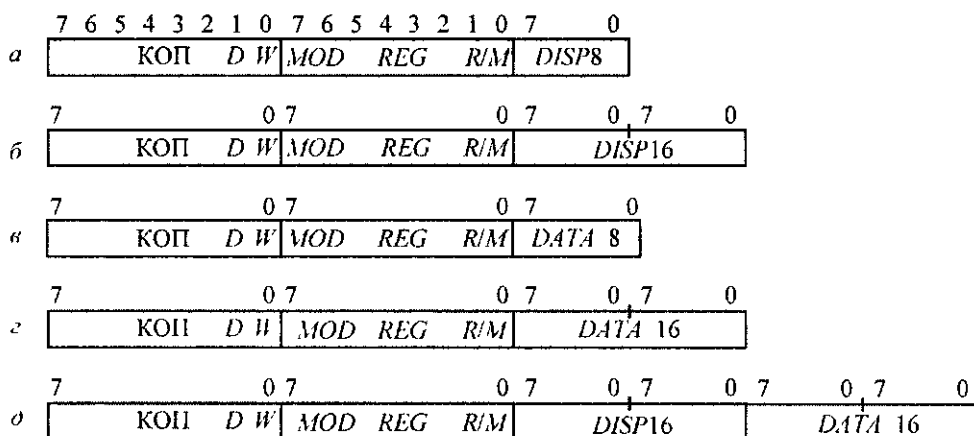


Рис. 10.23. Формат команд: а, б — із зміщенням;
в, г — з безпосередніми даними; д — із зміщенням та даними

Двобітове поле MOD показує умови обчислення виконавчої адреси EA : 00 — без зміщення; 01, 10 — відповідно із зміщенням $D8$ або $D16$; 11 — операнд знаходиться в регістрі. Трибітове поле R/M вказує разом з MOD спосіб формування адреси EA , яка розміщується в пам'яті чи регістрі (табл. 10.11).

Таблиця 10.11

R/M	MOD				
	00	01	10	11	
				$W = 0$	$W = 1$
000	$(BX) + (SI)$	$(BX) + (SI) + D8$	$(BX) + (SI) + D16$	AL	AX
001	$(BX) + (DI)$	$(BX) + (DI) + D8$	$(BX) + (DI) + D16$	CL	CX
010	$(BP) + (SI)$	$(BP) + (SI) + D8$	$(BP) + (SI) + D16$	DL	DX
011	$(BX) + (DI)$	$(BX) + (DI) + D8$	$(BX) + (DI) + D16$	BL	BX
100	(SI)	$(SI) + D8$	$(SI) + D16$	AH	SP

Таблиця 10.11. Продовження

R/M	MOD				
	00	01	10	11	
				W = 0	W = 1
101	(DI)	(DI) + D8	(DI) + D16	CH	BP
110	Прямий	(BP) + D8	(BP) + D16	DH	SI
111	(BX)	(BX) + D8	(BX) + D16	BH	DI

Трибітове поле вказує ім'я регістра, який містить операнд (табл. 10.12). Зазначимо, що при двох операндах у команді кожен з них може адресуватися своїм способом.

Таблиця 10.12

REG	W = 1	W = 0
000	AX	AL
001	CX	CL
010	DX	DL
011	BX	BL
100	SP	AH
101	BP	CH
110	SI	DH
111	DI	BH

Таким чином, постбайт визначає дві адреси: регістрів у полі *REG* і виконавчу *EA* в полі *R/M* за допомогою *MOD*. В командах пересилки одна з них є адресою джерела даних, друга — адресою приймача. При $d = 0$ поле визначає джерело даних, а адреса *EA* — приймач даних; при $d = 1$ — навпаки. В двомісних операціях при $d = 1$ адреса *EA* задає повний операнд і приймач даних, а поле *REG* — правий операнд; при $d = 0$ — навпаки. В ряді операцій (з одним операндом або з безпосередніми даними) поле КОП використовують для розміщення КОП.

Способи адресації

У мікропроцесорі використовують такі способи адресації.

- Неявний:

NOP — немає операцій;

HLT — зупинка;

LOCK — на виході установлюється сигнал блокування шини на час виконання наступної команди.

- Прямий регістровий (операнд в регістрі):

ADD AL, BL; (AL) ← (AL) + (BL) — обидва операнди мають пряму регістрову адресацію

AND AX, BX; (AX) ← (AX) ∧ (BX)

JMP CX — безумовний перехід на адресу (*CX*)

- Безпосередній (другий операнд в команді):

ADD AX, DATA16; (AX) ← (AX) + DATA16

SUB BL, DATA8; (BL) ← (BL) + DATA8

- Пряма адресація (виконавча адреса *EA* знаходиться в команді):

OR 1213, DATA16; (1213) ← (1213) ∨ DATA16

- Відносна адресація (адреса визначається відносно регістра *IP*):

JMP 2530; перехід безумовний на адресу (IP) ← (IP) + 2530

- Індексна адресація (з регістрами *SI* *DI*)
 $MOV [SI], 2CA0H, (SI) \leftarrow 2CA0H$
 $MOV [DI], 5555H, (DI) \leftarrow 5555H$
- Індексна адресація зі зміщенням
 $MOV 5[DI], 5501H, ((DI) + 5) \leftarrow 5501H$
- Базово-індексна адресація
 $MOV AX, [BX][SI], (AX) \leftarrow ((BX) + (SI))$
- Базово-індексна адресація зі зміщенням
 $MOV BX, 100[BX][SI], (BX) \leftarrow ((BX)(SI) + 100)$
- Адресація з багаторазовим повторенням
 $REP MOVSB STR1, STR$ — зміст комірки пам'яті сегмента *DS*, до якого звертаються за адресою в регістрі *SI*, пересилається в комірку пам'яті допоміжного сегмента *ES* на адресу *DI*. Зміст індексних регістрів *SI* та *DI* автоматично декрементується або інкрементується залежно від значень *D*

Префікс *REP* перед командою *MOVSB* викликає декрементування змісту *CX* і багатократне повторення команди до того часу, доки $CX = 0$. Потім здійснюється перехід до наступної команди.

Для передачі співпроцесору коду операції чи операнда призначена команда *ESC*. Після цієї команди мікропроцесор не виконує операцій, а тільки зчитує операнди з пам'яті і видає їх на шину.

При виконанні команд з умовчання використовують сегменти коду *CS*, даних *DS*, стека *SS* та допоміжного *ES*. При необхідності змінити сегмент у команді записують префікс (ім'я нового сегмента) замість сегмента даних використовують допоміжний сегмент.

Мікропроцесор K1810BM88 є практично повним програмним і апаратним аналогом процесора 8088. Але обмін даними виконується байтами по восьмирозрядній ШД. Така модифікація забезпечує сумісність з широким набором інтерфейсних схем.

10.4. Арифметичні співпроцесори

10.4.1. Співпроцесор K1810BM87

Арифметичні співпроцесори призначені для підвищення продуктивності ЦП в 100 і більше разів при виконанні операцій з багаторозрядними цілими й дійсними числами. Співпроцесор працює тільки в парі з визначеним мікропроцесором. Сполучення мікропроцесора і співпроцесора звичайно виконують простим об'єднанням їхніх виводів без додаткових мікросхем. Взаємодія двох мікросхем підтримується за допомогою контролера переривання. Розглянемо принцип побудови і функціонування співпроцесора на основі співпроцесора K1810BM87.

Мікросхема K1810BM87 (далі *BM87*) — це однокристальний 80-розрядний арифметичний співпроцесор (АСП), виконаний за *n*-МОН схемотехнологією. Він розміщений на кристалі розміром 5,5 × 5,5 мм, містить 65000 транзисторів, споживає потужність не більше 3 Вт від джерела живлення плюс 5 В. Схема АСП випускається в 40-виводному корпусі, має однофазну синхронізацію частотою 5 МГц від зовнішнього генератора.

Структура АСП містить операційний пристрій і пристрій ШІФ.

Операційний пристрій має у своєму складі:

- групу з восьми 80-розрядних регістрів, організованих в стек, які служать для зберігання операндів в розширеному форматі;
- модуль оброблення мантиси, який має 68-розрядний суматор, регістр зсуву і регістр результату та виконує задані операції над мантисами операндів і формує ознаки;
- модуль оброблення порядку, який виконує дії над значеннями полів порядку початкових операндів;
- постійний запам'ятовуючий пристрій констант, що зберігає в розширеному форматі сім числових констант, які часто використовуються;
- блок керування, пов'язаний з виходом черги байтів команд і побудований за мікропрограмним принципом. Він дешифрує чергову команду, формує керуючі сигнали, які визначають режим роботи АСП.

Пристрій шинного інтерфейсу містить групу допоміжних регістрів, буферний регістр, а також регістри черги команд і операндів.

Співпроцесор працює тільки в парі з ЦП ВМ86, установленим у максимальний режим ($MN/\overline{MX} = 0$), як показано на рис. 10.24.

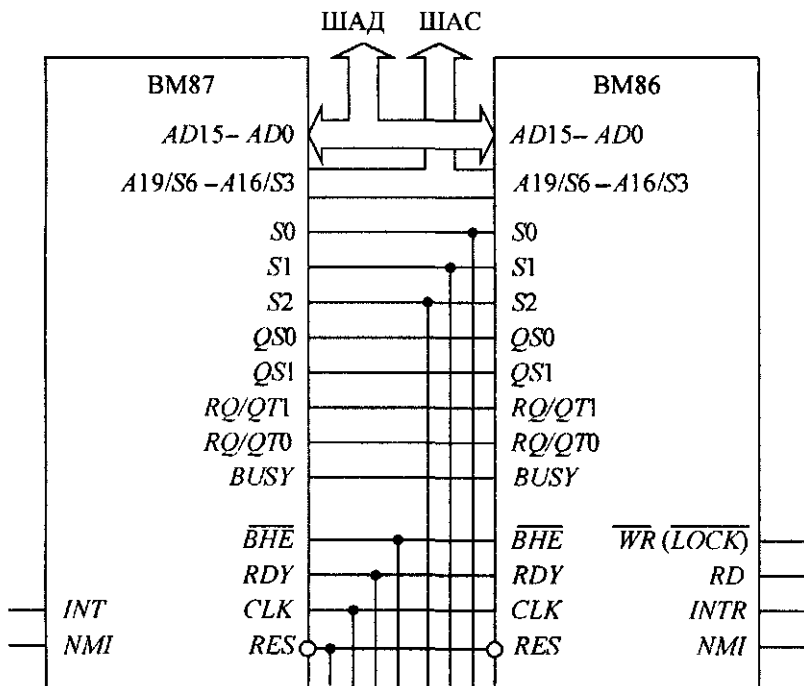


Рис. 10.24. Сполучення мікросхем ВМ86 і ВМ87

Команди з пам'яті вибирає тільки ЦП. Співпроцесор знаходиться в активному режимі, слідує за роботою ЦП і також записує в свою чергу команди. Арифметичний співпроцесор аналізує код стану черги команд по лініях $QS1$, $QS0$. Як тільки на цих лініях з'явиться код першого байта команди $ESC = 11011$, співпроцесор переходить в активний режим.

Якщо команда *ESC* вказує, що операнд є джерелом і розміщений в пам'яті, то АСП перехоплює адресу і записує операнд у свій регістр, а байти даних розміщує за чергою команд. Якщо в команді *ESC* адресу визначає приймач операнда, то співпроцесор після виконання команди записує результат по перехопленій адресі.

Система команд АСП ВМ87 містить 69 мнемонік і побудована на основі команди *ESC*, яка має машинний код 11011, розміщений на початку кожної команди співпроцесора. Мнемоніка кожної команди починається з букви *F* (*float* — плаваючий), що дозволяє розрізнити в програмі на мові асемблера команди мікропроцесора і співпроцесора.

Система команд АСП ВМ87 за функціональними ознаками розбивається на чотири групи

- передачі даних — завантаження, запам'ятовування, обміну,
- арифметичні — порівняння, аналізу, додавання, віднімання, множення і ділення,
- спеціальні — обчислювання квадратного кореня, тангенса, котангенса, логарифму та ін ,
- керування — ініціалізації співпроцесора, дозволу / заборони переривання, оброблення особливих ситуацій

Арифметичний співпроцесор ВМ87 оперує даними семи форматів цілими двійковими числами (два байти, чотири і вісім байтів), цілими двійково-десятьковими числами (10 байтів), дійсними числами в короткому, довгому і розширеному форматах

10.5. Суперскалярні 32-розрядні мікропроцесори із *CISC*-архітектурою

10.5.1. Скалярні й суперскалярні мікропроцесори

Процесори з одним конвеєром називаються **скалярними**. Процесори, які мають два і більше конвеєрів, називаються **суперскалярними**, вони можуть виконувати декілька команд за один машинний такт.

Сучасні 32-розрядні мікропроцесори мають архітектуру *CISC*, *RISC*, *Alpha*, *Power* та інші. Типовими виробами з *CISC*-архітектурою (складні команди, багато способів адресації, мало регістрів загального призначення) є мікропроцесори фірми *Intel*, в тому числі скалярні

- 80386, випущений у 1985 році, містить 275000 транзисторів, технологія — 1,2 мкм, адресний простір — 4 Гбайт, внутрішня черга команд — 16 байт. Це перший 32-розрядний процесор фірми *Intel*,
- 486, появився у 1989 р, число транзисторів — 1,2 млн, технологія — 1 мкм, має вбудований співпроцесор

До суперскалярних мікропроцесорів фірми *Intel* відносяться

- *Pentium*, розроблений у 1993 р, містить 3,1 млн транзисторів, технологія — 0,8 мкм, має два конвеєри,
- *Pentium Pro*, розроблявся одночасно з процесором *Pentium*. В одному корпусі розміщено 5,5 млн транзисторів ядра і 15,5 млн транзисторів для вторинного кеша об'ємом 256 Кбайт,

- *Pentium MMX*, розроблений в 1997 р. Має 4,5 млн транзисторів, технологія — 0,35 мкм. Забезпечує паралельне оброблення групи операндів однією командою, що прискорює виконання мультимедійних задач;
- *Pentium II*, появився в 1997 р. Це — скорочений варіант ядра *Pentium Pro*, в який ввели підтримку *MMX*. Кристали ядра і статичної пам'яті вторинного кеша розміщені на друкованій платі-картриджі й закриті спільною кришкою.

У 2000 р. фірма *Intel* випустила 64-розрядний процесор *Itanium* (кодована назва розробки *Merced*).

Перераховані моделі мікропроцесорів мають модифікації, які відрізняються частотою роботи (від 66 до 500 МГц і більше), енергоспоживанням, вартістю, областю використання (стаціонарні, мобільні, портативні та ін.).

Ряд фірм (*AMD*, *Cyrix*, *IBM*) випускають процесори, сумісні з перерахованими моделями *Intel* і мають свої характерні особливості. Фірми *DEC*, *Motorola*, *Texas Instruments*, *IBM* мають розробки більш потужних процесорів з *RISC*-архітектурою.

10.5.2. Структура мікропроцесора *Pentium*

Мікропроцесор *Pentium* виготовлений за 0,8-мікронною БіКМОН-технологією з комбінованим використанням біполярних транзисторів (для підвищення швидкодії) і КМОН-структур — для зменшення споживаної потужності та збільшення щільності компоновки. Перший процесор *Pentium* був розрахований на частоти роботи 60 і 66 МГц, випускався у корпусі типу *PGA-273* (матриця 21*21), мав напругу живлення плюс 5 В і споживав потужність 16 Вт.

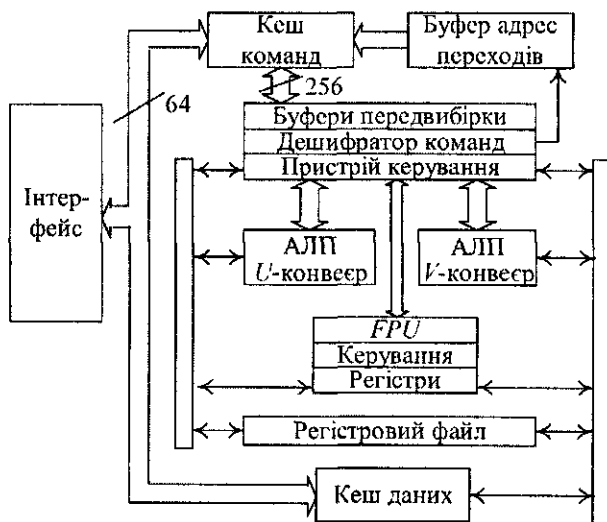


Рис. 10.25. Структура мікропроцесора *Pentium*

Структура мікропроцесора *Pentium* включає (рис. 10.25):

- виконавчі *U*- і *V*-конвеєри з АЛП кожний;
- пристрій обчислень з плаваючою комою *FPU*;
- регістровий файл;
- роздільні кеш команд і кеш даних;

- дешифратор команд і пристрій керування;
- буфери передвибірки з опередженням;
- буфери прогнозування адреси переходів;
- шинний інтерфейс з 64-розрядною ШД.

Цілочислові *U*- і *V*-конвеєри мають таку п'ятиступеневу структуру (рис. 10.26):

- *PF* (*Pre Fetch*) — попередня вибірка (передвибірка);
- *D1* (*Decode Stage1*) — декодування команд і визначення можливості запуску на виконання зразу двох команд;
- *D2* (*Decode Stage2*) — генерація адрес операндів;
- *EX* (*Execute*) — буфер оберненого запису завершення виконання команди і можлива зміна стану процесора;
- *X1*, *X2*, *WF* — використовуються *FPU*.

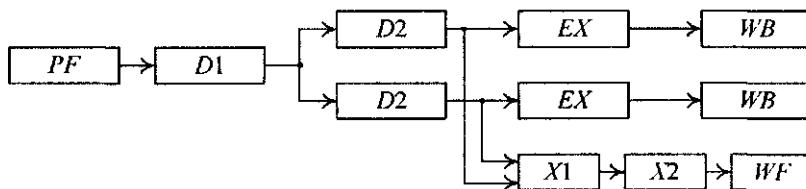


Рис. 10.26. Конвеєри мікропроцесора Pentium

Кожен конвеєр має свій АЛП, сукупність пристроїв генерації адрес на ступені *D2* з використанням багатоканального суматора та інтерфейса з кешами. Процес видачі на виконання одночасно двох команд називається парним запуском; він ніяк не обмежує програмування задач. Обидва конвеєри функціонально подібні, але головний *U*-конвеєр виконує всі цілочислові операції, а *V*-конвеєр — тільки прості та деякі команди з плаваючою комою. Конвеєри працюють одночасно і при парному запуску спочатку в *U*-конвеєр подається більш складна команда, а потім у *V*-конвеєр поступає простіша команда. Якщо дві команди запустити не можна, то працює *U*-конвеєр, а *V*-конвеєр простоює. Якщо виконання команди зупиняється на будь-якому ступені, то команда в другому конвеєрі зупиняється на тому ж ступені (за окремими виключеннями).

Парні запуски можливі при відсутності залежності між двома командами, наприклад, при зверненні до одного і того ж регістра. Команди *FPU* не запускаються в парі з цілочисловими командами.

Кожний конвеєр має 64-розрядний буфер запису *WB*, з якого інформація передається в пам'ять в порядку її генерації процесорним ядром.

Математичний співпроцесор *FPU* реалізований на триступеневому конвеєрі з плаваючою комою, який прибудований до цілочислового конвеєра. Команди *FPU* проходять по спільному конвеєру до ступеня *EX*, після цього вони проходять по ступенях *X1*, *X2*, *WF* конвеєра *FPU*. Пристрій *FPU* підтримує операції з операндами 32-, 64- і 80-розрядного формату.

Для зберігання поточних операндів селекторів сегментів пам'яті та інформації про стан процесора передбачений регістровий файл, який містить вісім загальних регістрів, шість сегментних регістрів та два регістри станів і керування.

Окремі кеші команд і даних об'ємом 8 Кбайт кожний виключають конфлікти при одночасному звертанні до команд і даних на стадії попередньої вибірки. Кожний кеш — це два двоканальні асоціативні модулі з довжиною рядка 32 байт і шириною зовнішньої шини 8 байт. Пам'ять кожного кеша розбита на вісім банків з чергуванням по границях у 4 байт. Кеш даних має два інтерфейси, по одному для кожного конвеєра.

Якщо конвеєрам необхідні дані з різних банків, то вони обслуговуються одночасно в одному такті. З кожним кешем зв'язаний свій асоціативний буфер трансляцій, який перетворює логічні адреси у фізичні.

Інформація в рядках кеша даних має такі ознаки: модифікована; приналежна тільки даному кешу; розподільна з іншими кешами; недійсна. Це забезпечується протоколом *MESI* (*Modified – Exclusive – Shared – Invalid* — протокол підтримки когерентності пам'яті при наявності кеша і названий за станами рядків). Кеш даних може настроюватися на алгоритм наскрізного або оберненого записування.

Кеш команд захищений від випадкових записів і його рядки помічаються як спільні з другими кешами команд або як недійсні.

Блок передвибірки команд має чотири 32-байтові буфери, які працюють разом з буфером адрес переходів *BTB* (*Branch Target Buffer*). Вибірка проводиться послідовно до появи команди переходів, після чого блок *BTB* прогнозує — буде перехід чи ні. Наприклад, після виконання кожного програмного циклу (а це може бути сотні разів) обчислюється умова — повернутися до початку циклу або вийти з нього. Ці два рішення (шляхи) називаються прогнозуванням адрес переходів.

Блок *BTB* запам'ятовує адресу і команду першого входження в цикл і передбачає, що з появою команди переходу цикл буде продовжуватися. Якщо прогноз правильний, то перехід виконується без затримки. В іншому випадку використовуються додаткові три-чотири такти для обчислення умови дійсного переходу і зчитування наступної команди.

Зовнішня шина має 32 лінії адреси чи 64 лінії даних, але всередині обробляються 32-розрядні дані. Ця шина забезпечує можливість пакетних пересилок (швидкість до 528 Мбайт/с) і циклів пакетного узгодження інформації між кешами і ОЗП при використанні алгоритму оберненого записування.

10.5.3. Функціональні реєстри

Архітектура 32-розрядних CISC-мікропроцесорів 386, 486, *Pentium*, *Pentium Pro*, *Pentium II* і *Celeron* характеризується спільною базовою програмною моделлю, яка охоплює набір реєстрів процесорів, формат команд і способи адресації, типи даних, організацію пам'яті, введення-виведення, переривання та ін.

Набір основних функціональних реєстрів указаних CISC-мікропроцесорів показаний на рис. 10.27.

Вісім 32-розрядних реєстрів загального призначення *EAX*, *EBX*, *ECX*, *EDX*, *ESP*, *EBP*, *ESI*, *EDI* використовують для зберігання даних і адрес; їх молодші 16 розрядів доступні окремо з іменами *AX*, *BX*, *CX*, *DX*, *SP*, *BP*, *SI*, *DI*. При операціях з байтами реєстри *AX*, *BX*, *CX*, *DX* розділяються на молодші байти (*AL*, *BL*, *CL*, *DL*) і старші байти (*AH*, *BH*, *CH*, *DH*). Доступ до окремих байтів забезпечує додаткову гнучкість при операціях з даними.

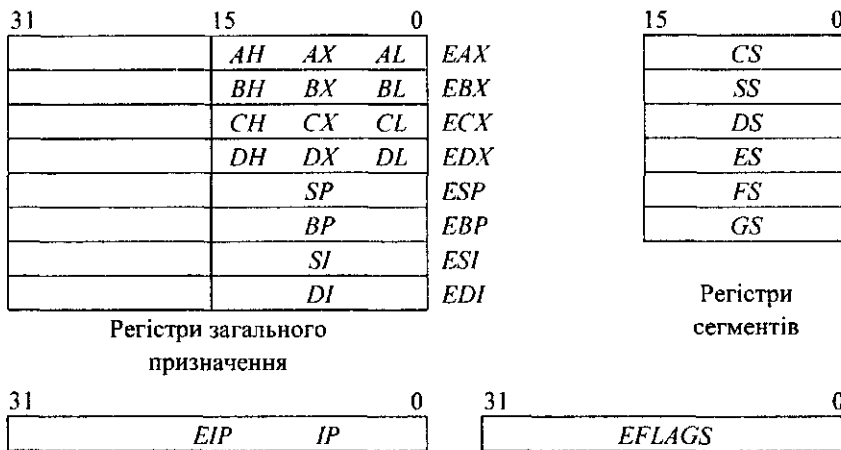


Рис. 10.27. Основні функціональні регістри CISC-мікропроцесорів

Шість 16-розрядних сегментів регістрів *CS*, *SS*, *DS*, *ES*, *FS* і *GS* містять значення селекторів, які адресують поточні сегменти пам'яті. Селектор в *CS* забезпечує звернення до поточного сегмента команд, селектор в *SS* — до поточного сегмента стека, селектор в *DS*, *ES*, *FS* і *GS* — до поточних сегментів даних.

До покажчика команд *EIP* входить зміщення наступної команди, яка виконується відносно бази сегмента коду. При 16-розрядній адресації використовуються *IP* (молodші 16 біт). Регістр прапорців *EFLAGS* містить ознаки результату виконання команди, керує обробкою виключень і маскованих переривань та ін.

Розряди, визначені для процесорів *VM80*, *VM85* і *VM86*, мають однакові позначення і функції.

Процесори можуть працювати в одному з двох режимів:

- реальної адресації (або просто — реальний режим — *Real Mode*) повністю сумісний з *8086* і може адресувати до 1 Мбайт фізичної пам'яті;
- захищений режим віртуальної адресації (або просто — захищений режим — *Protected Mode*). У цьому режимі адресується до 4 Гбайт пам'яті, через які при використанні механізму сторінкової адресації може відображатися до 4 Тбайт віртуальної пам'яті для кожної задачі.

Суттєвим доповненням способів адресації є режим віртуальної адресації процесора *8086 VM (Virtual 8086 Mode)*. Він є особливим станом захищеного режиму, в якому процесор функціонує як *8086*, але може адресувати до 4 Мбайт фізичної адресації; устанавлюється бітом *VM* в регістрі прапорців.

10.5.4. Типи даних

Процесори безпосередньо підтримують (використовують як операнди) знакові й беззнакові цілі числа, рядки бітів, байтів, цифр і символів, покажчики і числа з плаваючою комою. Розглянемо ці типи даних детальніше:

- біт (*Bit*) — одиниця інформації; задається адресою слова і своїм номером у слові;
- бітове поле (*Bit Field*) — група до 32 суміжних бітів, які розміщуються не більше, ніж у чотирьох байтах;

- бітовий рядок (*Bit String*) — набір суміжних бітів довжиною до 4 Гбіт,
- байт (*Byte*) — 8 біт,
- числа без знака та зі знаком
- байт (*Byte*) — 8 біт, слово (*Word*) — 16 біт, подвійне слово (*Double Word*) — 32 біт, учетверенне слово (*Quadre Word*) — 64 біт

Від'ємні числа зберігаються у пам'яті в доповняльному коді, одиничне значення старшого біта визначає від'ємне число

Дійсні числа у форматі з плаваючою комою обробляються співпроцесором *FPU*. До них відносяться такі дані

- одинарна точність — 32 біт (23 біт відведено для мантиси, 8 біт — для порядку)
- подвійна точність — 64 біт (52 біт — мантиса, 11 біт — порядок)
- підвищена точність — 80 біт (64 біт — мантиса, 15 біт — порядок)

Двійково-десяткові *BCD* числа

- 8-бітні упаковані містять дві десяткові цифри в одному байті
- 8-бітні неупаковані, містять одну десяткову цифру в байті,
- 80-бітні упаковані

Процесори використовують покажчики для звернень, наприклад, до підпрограм. Близький (внутрішньо сегментний) покажчик *NEAR* — це 32-бітне зміщення від бази поточного сегмента. Далекий (міжсегментний) 48-бітний покажчик *FAR* (16-бітний селектор і 32-бітне зміщення) застосовується при передачах керування в другий сегмент.

В процесорах x86, як правило, слова записуються у двох суміжних комірках пам'яті, починаючи з молодшого. Адресою слова є адреса його молодшого байта. Подвійні слова записуються в чотирьох суміжних байтах, його адреса визначається адресою молодшого байта. Такий порядок адресації називається *Little-Endian Memory Format*. В інших сім'ях процесорів застосовують обернений порядок *Big-Endian Memory Format*, в якому адресою слова чи подвійного слова є адреса його старшого байта, а молодші байти розміщуються в наступних адресах. Для взаємного перетворення форматів слова є інструкція *XCHG*, а для подвійного слова — *BSWAP*.

10.5.5. Формати команд

Узагальнений формат команд показаний на рис. 10.28

КОП	<i>MOD R/M</i>	<i>SIB</i>	Зміщення	Операнд
1 або 2 байт	0 чи 1 байт	0 чи 1 байт	0, 1, 2 або 4 байт	0, 1, 2 чи 4 байт

Рис. 10.28 Узагальнений формат команд

Команди містять від одного до 11 байт. В середньому довжина команди становить 4–5 байт. Розглянемо призначення основних полів формату команд.

Код операції КОП займає 1 чи 2 байт. Перед КОП в окремих випадках записується один чи декілька префіксних байтів, які модифікують операцію, що виконується. У багатьох командах арифметико-логічних операцій і пересилок значення біта *W* в першому байті КОП визначає розрядність операндів: *W* = 0 — операції з байтами, *W* = 1 — операції з 16- чи 32-розрядними операндами.

Розрядність слів (16 або 32 біт) визначається режимом роботи мікропроцесора і встановлюється бітом *D* в дескрипторі сегмента коду. При виконанні окремих команд розрядність операндів змінюється відповідним префіксом.

У ряді команд перший байт містить поле *REG* (три біти) чи поле *SREG* (два або три біти), які визначають відповідно регістр операндів (табл. 10.13) або сегментний регістр (табл. 10.14).

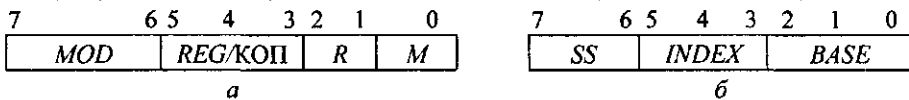
Таблиця 10.13

Поле <i>REG</i>	Розрядність операндів		
	8	16	32
000	<i>AL</i>	<i>AX</i>	<i>EAX</i>
001	<i>CL</i>	<i>CX</i>	<i>ECX</i>
010	<i>DL</i>	<i>DX</i>	<i>EDX</i>
011	<i>BL</i>	<i>BX</i>	<i>EBX</i>
100	<i>AH</i>	<i>SP</i>	<i>ESP</i>
101	<i>CH</i>	<i>BP</i>	<i>EBP</i>
111	<i>BH</i>	<i>DI</i>	<i>EDI</i>

Таблиця 10.14

<i>SREG</i> 2 біта	<i>SREG</i> 3 біта	Регістр сегмента
00	000	<i>ES</i>
01	001	<i>CS</i>
10	010	<i>SS</i>
11	011	<i>DS</i>
	100	<i>FS</i>
	101	<i>GS</i>

Байт адресації *MOD R/M* містить три поля (рис. 10.29, а). Поля *MOD* і *R/M* задають адресу одного з операндів, який зберігається в регістрі чи в комірці пам'яті.

Рис. 10.29. Формати байтів: а — *MOD R/M*; б — *SIB*

Байт *SIB* містить поля *INDEX*, *BASE* та *SS*. Вони визначають відповідно регістри, які використовуються як базові та індексні, а також масштабний коефіцієнт для модифікації значення індекса (рис. 10.29, б).

У полі зміщення задають значення зміщення довжиною 8, 16, або 32 біт. При виконанні операцій з безпосередніми даними один з операндів записують в останніх байтах команди.

10.5.6. Захищений режим. Дескриптори

Захищений режим забезпечує захист апаратних і програмних ресурсів однієї задачі від можливого впливу іншої. Основним ресурсом, що захищається, є пам'ять, в якій зберігаються команди, дані та системні таблиці.

Захист пам'яті оснований на **сегментації**. Це — механізм, за допомогою якого пам'ять розділяється на окремі області адресного простору визначеного призначення. Така область з множиною комірок пам'яті із суміжними адресами називається **сегментом**.

Сегменти пам'яті задають слова-селектори, які операційна система завантажує у сегменті регістри. У режимі реальної адресації прикладна програма може помилково записати в сегментний регістр селектор чужого сегмента. У захищеному режимі така ситуація виключена — доступ до свого сегмента забезпечується як селектором, так і попередньо сформованими таблицями дескрипторів сегментів (їхні значення прикладна програма не може змінити).

Дескриптор. — це восьмибайтова структура даних, які використовуються для визначення властивостей сегментів: базова адреса сегмента *BASE* і його розмір *LIMIT*, його призначення *TYPE* (тип), характеристики захисту і додаткова інформація. Узагальнений формат дескриптора сегмента показаний на рис. 10.30.

Байт 4				Байт 3				Байт 2		Байт 1	
31	24	23	16	15	8	7			0		
<i>BASE</i>						<i>LIMIT</i>					
15–0						15–0					

Байт 8			Байт 7				Байт 6 – <i>AR</i>				Байт 5			
63	56	55	52	51	48	47	44		40					
<i>BASE</i>			<i>G</i>	<i>D</i>	<i>X</i>	<i>U</i>	<i>LIMIT</i>		<i>P</i>	<i>DPL</i>	<i>S</i>	<i>TYPE</i>	<i>A</i>	<i>BASE</i>
31–24							19–16							23–16

Рис. 10.30. Формат дескриптора

Розглянемо призначення окремих полів дескриптора сегмента.

Поле базової адреси *BASE* довжиною 32 біт займає байти 3, 4, 5 і 8 формату і визначає початкову адресу сегмента в лінійному адресному просторі 4 Гбайт.

Поле границі *LIMIT* довжиною 20 біт займає байти 1, 2 і молодшу тетраду байта 7. Границя сегмента задає розмір сегмента. Якщо в байті 7 біт гранулярності (дробовості) $G = 0$, то максимальний розмір сегмента становить 1 Мбайт; при $G = 1$ максимальний розмір сегмента дорівнює 4 Мбайт.

В байті керування доступом *AR* (*Access Rights*) поля мають такі призначення:

- A — біт звернення; при $A = 0$ — до сегмента не було звернення; якщо $A = 1$, то селектор даного сегмента завантажений у регістр сегмента;
- *TYPE* — трибітне поле типу, яке визначає цільове використання сегмента та операції, які в ньому можуть виконуватися (наприклад, сегменти коду, стека, даних);
- *DPL* — двобітне поле атрибутів привілеїв сегмента;
- *S* — системний біт;
- P — біт присутності в пам'яті; при $P = 1$ сегмент відображений у фізичній пам'яті; при $P = 0$ сегмента у пам'яті немає, тому поле базової адреси і ліміту не використовується.

У старшій тетраді байта 7 біти мають таке призначення:

- U — біт користувача (системного програміста);
- X — резерв, містить лог. 0;
- D — визначає розрядність адрес і операндів: по замовчуванню: $D = 0$ — 16 біт; $D = 1$ — 32 біт.

Процесори мають спеціальні засоби, які в захищеному режимі не допускають несанкціонованого доступу до пам'яті. Двобітне поле *DPL* устанавлює чотири рівні привілеїв з номерами від нуля (максимальний привілей) до трьох (мінімальний привілей). Найбільш захищеним є ядро операційної системи (ініціалізація роботи, керування доступом до пам'яті та ін.), а найменш захищені — прикладні програми.

Визначена в адресному просторі пам'яті множина дескрипторів називається **таблицею дескрипторів**. Кожний дескриптор займає в пам'яті 8 байт. Кількість де-

скрипторів у таблиці — довільна, але не може перевищувати максимальних значень 8192 при цьому необхідний об'єм пам'яті для розміщення таблиці дорівнює $8192 \cdot 8 = 64$ Кбайт. Порядок розміщення дескрипторів у таблиці ролі не відіграє.

Дескриптори розміщують у загальносистемних таблицях дескрипторів трьох типів

- *GDT* — глобальна таблиця дескрипторів,
- *LDT* — локальна таблиця дескрипторів,
- *IDT* — таблиця дескрипторів переривань

Глобальна таблиця дескрипторів використовується спільно всіма задачами, які виконуються в даному комп'ютері.

Локальна таблиця *LDT* визначає сегменти, доступні тільки конкретній задачі. Таблиці *LDT* створюються при необхідності по одній для кожної задачі. Для однозадачних систем таблиця *LDT* може бути відсутня (вся інформація міститься у глобальній таблиці).

Таблиця переривань *IDT* містить дескриптори спеціальних об'єктів, які визначають точки входження процедур оброблення переривань і особливих випадків.

Доступ до таблиці дескрипторів забезпечують спеціальні реєстри (рис. 10.31)

- *GDTR* — реєстр глобальної таблиці дескрипторів,
- *LDTR* — реєстр локальної таблиці дескрипторів,
- *IDTR* — реєстр таблиці дескрипторів переривань

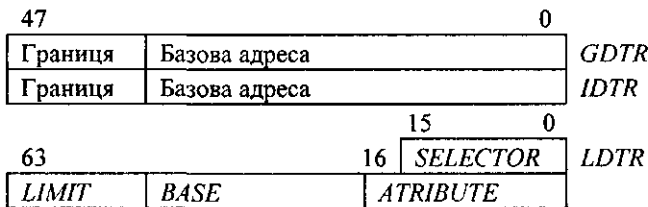


Рис. 10.31. Формати реєстрів дескрипторів

Реєстри *GDTR* і *IDTR* мають однаковий формат і функціональний зміст полів. Реєстр *LDTR* має специфічну структуру, яка вміщує

- 16-бітне поле *SELECTOR*, зміст якого виконує функцію селектора сегментного реєстра,
- 64-бітне поле власне локального дескриптора з полями бази *BASE*, границі *LIMIT* та атрибутів *ATTRIBUTE* (тобто формат звичайного дескриптора)

Для доступу до глобальної таблиці *GDI* процесор завантажує селектор в один із сегментних реєстрів *CS*, *SS*, *DS* та інші. Формат селектора показаний на рис. 10.32.

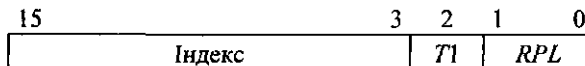


Рис. 10.32. Формат селектора

У двобітному полі *RPL* записується номер привілею, який вимагає задача. Однобітне поле *T1* задає тип таблиці дескрипторів: при $T1 = 0$ — глобальна, при $T1 = 1$ — локальна. Поле індексу довжиною 13 біт задає зміщення в адресі таблиці дескриптора. Процесор автоматично множить значення індексу на 8 (три нулі справа),

складає з базовою адресою, яку містить реєстр *GDTR*, і зчитує необхідну таблицю. Одночасно виконується перевірка, щоб значення помноженого індексу не перевищувало границі, заданої в реєстрі *GDTR*. Якщо ця умова не виконується, то програма переривається.

Дескриптор локальної таблиці *LDT* розміщений у глобальній таблиці *GDT*. Для доступу до локального дескриптора процесор попередньо завантажує поле *SELECTOR* реєстра *LDTR*. Поле *SELECTOR* визначає зміщення адреси локального дескриптора в таблиці *GDT*. Значення цього поля складається з базовою адресою глобального реєстра і створює адресу локального дескриптора. Після перевірки допустимих границь 8 байт локального дескриптора зчитується і записується в основній частині реєстра *LDTR* (він розташований у процесорі). Таким чином, доступ до локального дескриптора виконується швидше, ніж до глобального.

Зчитаний дескриптор автоматично засилається у відповідний 'тіньовий' реєстр швидкодіючої кеш-пам'яті (рис. 10.33). Кожний сегментний реєстр (*CS*, *SS* та ін.) має функціонально пов'язані тіньові реєстри дескрипторів, що прискорює доступ до даних.

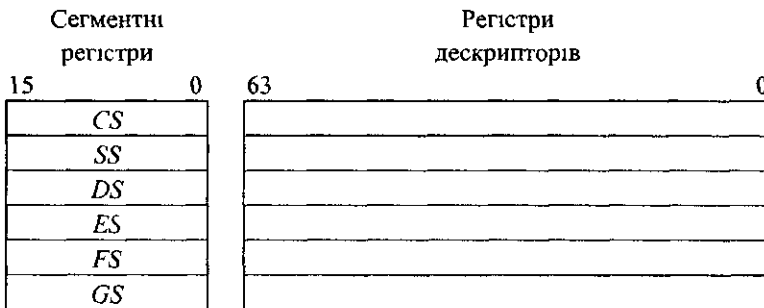


Рис. 10.33 Тіньові реєстри дескрипторів сегментів

Команди завантаження селекторів використовують прикладні програми. Команди завантаження глобальних і локальних реєстрів доступні тільки сегментним програмам.

Захист пам'яті за допомогою сегментації не дозволяє

- використовувати сегменти не за призначенням,
- порушувати права доступу (наприклад, записування в сегмент, який призначений тільки для читання),
- адресувати до сегментів, які виходять за границю сегмента,
- змінювати зміст таблиці дескрипторів без достатніх привілеїв

10.5.7. Способи адресації в процесорах з CISC-архітектурою

При сегментній організації пам'яті 32-розрядна фізична адреса комірки пам'яті формується додаванням базової адреси сегмента і відносної адреси комірки всередині сегмента.

Ефективна адреса операнда *EA* формується залежно від полів *MOD*, *R/M* і *SAB*.

Значення БА створюється арифметичним додаванням трьох компонентів:

- змісту базового реєстра *EBP* чи *EBX*;
- змісту індексного реєстра *ESI* або *EDI*;
- 8-, 16- чи 32-розрядного зміщення *d8*, *d16* або *d32*, які задані в команді.

У 32-розрядних процесорах способи адресації є розвитком і модифікацією методів адресації процесора 8086. Перечислимо основні способи адресації.

- Безпосередня — операнди задаються в байтах команди.
- Регістрова — операнд вибирається з реєстра, який визначається полем *R/M*.
- Непряма регістрова — адреса операнда знаходиться в реєстрах загального призначення.
- Пряма адресація — відносна адреса задається байтами команди.
- Базова адресація — відносна адреса формується додаванням змісту базового реєстра *BX* або *BP* і зміщенням в команді.
- Базово-індексна адресація — відносна адреса створюється додаванням змісту базового і сегментного реєстрів.
- Базово-індексна адресація із зміщенням — це варіант базово-індексної адресації з додаванням зміщення.
- Індексна адресація з масштабуванням — відносна адреса створюється додаванням масштабованого індексу (множення на 1, 2, 4 і 8) і 32-розрядного зміщення.
- Базово-індексна адресація з масштабуванням — відносна адреса створюється додаванням масштабованого індексу і бази.
- Базово-індексна адресація зі зміщенням і масштабуванням — це варіант попереднього способу з додатковим додаванням зміщення.
- Відносна адресація — адреса формується як сума змісту реєстра *IR*, яка відповідає поточному моменту і зміщенню.

Структура формування 32-розрядної фізичної адреси в пам'яті показана на рис. 10.34.

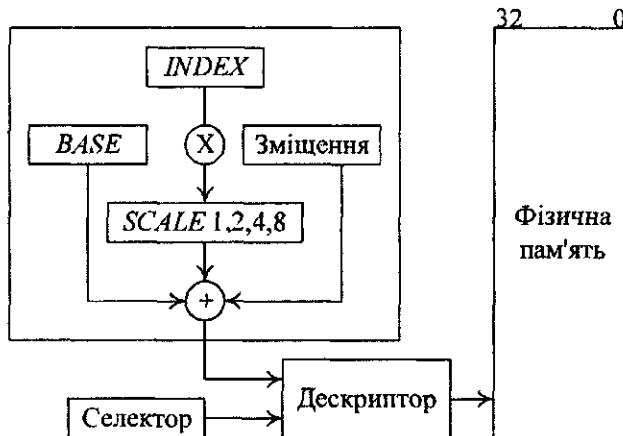


Рис. 10.34. Формування адреси пам'яті 32-розрядного процесора в захищеному режимі

10.5.8. Команди суперскалярних CISC-мікропроцесорів

Система суперскалярних CISC-мікропроцесорів порівняно з *Intel 80486* суттєво розширена і модифікована. Збільшена розрядність адрес і операндів, створена більш гнучка система адресації, використовуються нові типи команд і даних. Систему команд розділяють на функціональні групи.

- Команди пересилки даних (всього біля 40) дозволяють передавати константи або змінні між регістрами і пам'яттю, а також портами введення — виведення.
- Команди двійкової арифметики (біля 15) виконують усі арифметичні дії з байтами, словами і подвійними словами.
- Команди десяткової арифметики (біля 10) дозволяють оперувати з неупакованими і упакованими двійково-десятковими даними. Операції з цими числами вимагають застосування команд корекції форматів.
- Команди логічних операцій (4) реалізують усі функції булевої алгебри з байтами, словами і подвійними словами.
- Логічні та циклічні зсуви (всього 10) виконують над змістом регістрів чи операндів у пам'яті. Число позицій, на яке проводиться зсув, можна задавати операндом та іншими способами.
- Команди оброблення бітів і байтів (26) дозволяють перевіряти та установлювати значення вказаного операнда чи біта.
- Команди передачі керування (біля 30) здійснюються за допомогою команд безумовних та умовних переходів. Команда виклику процедур *CALL* передає керування в точку переходу, при цьому адреса наступної за нею команди зберігається в стеку.
- Рядкові операції (24) виконуються з операндами в пам'яті.
- Операції з прапорцями (всього 13) дозволяють змінювати значення окремих прапорців, а також зберігати їхні значення в стеку та відновлювати ці значення.
- Команди завантаження покажчиків (5) забезпечують завантаження покажчиків з пам'яті в регістр загального призначення і відповідний сегментний регістр.
- Команди математичного співпроцесора *FPU* (92) виконують операції з дійсними, цілими і двійково-десятковими числами, які представлені у форматі з плаваючою комою.
- Системні команди (32) — це операції керування захистом, завантаженням і зберіганням регістрів дескрипторів, перевірки привілеїв, тестування, керування кешуванням пам'яті, зупинки процесора та ін.

Кількість команд у кожній групі вказана тільки для базових мнемонік.

10.5.9. Узагальнена характеристика суперскалярних CISC-мікропроцесорів

Покоління суперскалярних CISC-мікропроцесорів почалося з процесора *Pentium* і характеризується:

- великою кількістю складних машинних команд різної розрядності й переважно двооперандного формату (у команд *RISC* — триоперандний формат);

- наявністю передвибірки, конвеєра команд і динамічного прогнозування умовних переходів;
- великим числом способів адресації;
- використанням всього восьми регістрів загального призначення (у *RISC* — як мінімум 32);
- широким використанням мікрокоманд;
- роздільними кешами команд і даних та їхнім заповненням у пакетному режимі;
- вбудованим співпроцесором *FPU*.

Архітектура *Pentium* є оптимізованою, оскільки вона має властивості сучасних *RISC*-мікропроцесорів:

- апаратну реалізацію простих команд, які виконуються за один машинний такт;
- суперскалярність і 64-розрядну зовнішню шину даних;
- приблизно однакову продуктивність конвеєрного *FPU* для операцій з одинарною, подвійною і розширеною точністю;
- прогнозування переходів, наявність 64- і 256-розрядних шин на кристалі;
- ефективні засоби забезпечення багатопроцесорності та ін.

Таким чином, *Pentium* відноситься до *CISC*-архітектур, але має багато властивостей *RISC*-мікропроцесорів.

Наступне покоління *CISC*-мікропроцесорів відраховується від випуску *Pentium Pro*, *Pentium II*, *Pentium III* та *Celeron*. Вони мають такі основні архітектурні удосконалення:

- можливість виконання команд не в порядку їхнього розміщення в програмі;
- вбудований вторинний кеш об'ємом до 2 Мбайт;
- появу двох незалежних внутрішніх шин: одну — для зв'язку з пам'яттю, другу — для обміну з вторинним кешем;
- керування енергоспоживанням;
- наявність вбудованих засобів контролю операцій за модулем два;
- розширення можливостей побудови багатопроцесорних МПС з симетричною архітектурою.

Розвиток сучасних мікропроцесорів визначається конкуренцією між фірмами, які розвивають напрямки *CISC*- та *RISC*-архітектур.

10.6. Суперскалярні мікропроцесори з *RISC*-архітектурою

10.6.1. Основні положення

RISC-архітектура є основою сучасних робочих станцій і серверів. Елементи архітектури із спрощеною системою команд застосували в суперкомп'ютерах *Cray*. В 1980 р. були розроблені машини *RISC-I* і *RISC-II*; їх головними ідеями було відокремлення повільної пам'яті від швидкодійчих регістрів і використання регістрових вікон. Роком пізніше був опублікований опис машини *MIPS*, основним аспектом якої була

ефективна реалізація конвеєрної обробки за допомогою детального планування компілятором його завантаження.

Остаточно поняття архітектури *RISC* у сучасному виді сформувалося на базі проєктів трьох комп'ютерів — *RISC*, 801 фірми *IBM* та *MIPS*.

Загалом *RISC*-архітектура має такі риси:

- порівняно невеликий (скорочений) набір команд, які найбільше застосовуються у прикладних програмах;
- триоперандний формат команд з довжиною операндів 32 або 64 розряди;
- мало способів адресації;
- великий регістровий файл (16, 32, 64 і більше регістрів);
- апаратна реалізація простих команд за один машинний такт;
- звернення до пам'яті за допомогою спеціальних команд завантаження-запам'ятовування, які обробляють групу регістрів;
- широке використання регістрових вікон при переключеннях задач;
- суперскалярність — за один машинний такт виконуються дві і більше команд;
- передбачення переходів;
- наявність засобів підтримки багатопроцесорності.

В цілому структура *RISC*-процесорів містить:

- два і більше виконавців-пристроїв, в тому числі і з плаваючою комою;
- роздільні кеші команд і даних;
- блок прогнозування переходів;
- інтерфейс з 64-розрядною зовнішньою шиною даних.

Розробкою і промисловим випуском *RISC*-процесорів займається ряд відомих фірм. Вони присвоюють своїм виробам власні назви архітектур, наприклад:

- *POWER* — фірма *IBM*, *Power PC* — фірми *Motorola*, *Apple* та *IBM*;
- *Alpha* — фірма *DEC*;
- *MIPS* — фірма *MIPS Technology* та ін.

Розглянемо особливості побудови і функціонування процесорів з архітектурою *POWER*.

10.6.2. Архітектура *POWER*

Архітектура *POWER* — це розвиток ідей, які були закладені в основу процесора *IBM 801* в напрямку реалізації суперскалярної обробки, скорочення довжини конвеєра і часу виконання команд та пріоритетної орієнтації на ефективне виконання операцій з плаваючою комою.

Архітектура *POWER* в багатьох відношеннях є традиційною *RISC*-архітектурою, яка має фіксовану довжину команд, прості способи адресації, операції регістр-регістр і триоперандний формат команд. Але архітектура *POWER* має додаткові властивості, які відрізняють її від інших архітектур:

- Набір команд оснований на ідеї суперскалярної обробки. В процесорі команди розподіляються по трьох незалежних виконавчих пристроях: переходів, з фіксованою і плаваючою комою. Команди одночасно поступають і закінчуються у цих трьох пристроях.
- Архітектура *POWER* розширена кількома "змішаними" командами: групового

завантажування і запам'ятовування реєстрів, автоінкрементної адресації, операцій з полями бітів та ін

- Архітектура переходів в *POWER* організована за принципом “попереднього перегляду умовних переходів”
- Архітектура *POWER* визначає розширені властивості реєстра умов (прапорців) Для цього вводиться спеціальний біт в коді операції в кожній команді для модифікацій реєстра умов, а також використовуються вісім реєстрів прапорців Це розширює кількість бітів умов

Після першої реалізації архітектури *POWER* (1990 р) з'явилися процесори *POWER2* і *POWER2+*

Багатокристалний набір *POWER2* складається з восьми напівказаних мікросхем

- блок кеш-пам'яті команд — 32 Кбайт,
- пристрій цілочислової арифметики АЛП — містить два конвеєри і два блоки реєстрів У кожному блоці є 32 реєстри довжиною 32 біт Виконує цілочислові та логічні операції та всі звернення до пам'яті,
- пристрій операції з плаваючою комою *FPU* — містить два конвеєри для виконання операцій з подвійною точністю, а також 54 реєстри довжиною 64 біт,
- чотири блоки кеш-пам'яті даних — всього 256 Кбайт,
- блок керування пам'яттю

Цей набір кристалів містить 23 млн транзисторів на площі 12 17 см², виконаний за КМОП-схемотехнікою з проектними нормами 0,45 мкм, на частоті 66,5 МГц споживає потужність 65 Вт

Система *Power PC* використовує однокристалну реалізацію архітектури *POWER* для побудови малих машин На основі архітектури *Power PC* послідовно були розроблені кристали 601 (для настільних робочих станцій), 603 (для портативних машин), 604 (для високопродуктивних настільних систем), 620 (для серверних конфігурацій з 64-бітовою організацією) У цих процесорах були зроблені декілька змін у напрямках

- спрощення архітектури з метою реалізації на однокристалних процесорах
- виключення команд, які перешкоджають підвищенню тактової частоти,
- введення безчергового виконання команд,
- розширення засобів підтримки симетричного багатопроесорного оброблення

Процесор *Power PC* 603 містить (рис 10 35)

- пристрій оброблення переходів і вибірки команд,
- цілочисловий пристрій АЛП,
- пристрій для операцій з плаваючою комою *FPU*,
- кеш-пам'ять для даних і команд ємністю по 8 Кбайт кожен,
- набір із 32 цілочислових реєстрів РЗП,
- набір із 32 реєстрів РПТ для чисел з плаваючою комою,
- пристрій черги і розподілу команд,
- буфер завершення команд,
- буфери записів і завантаження та інтерфейс із зовнішніми ША і ШД

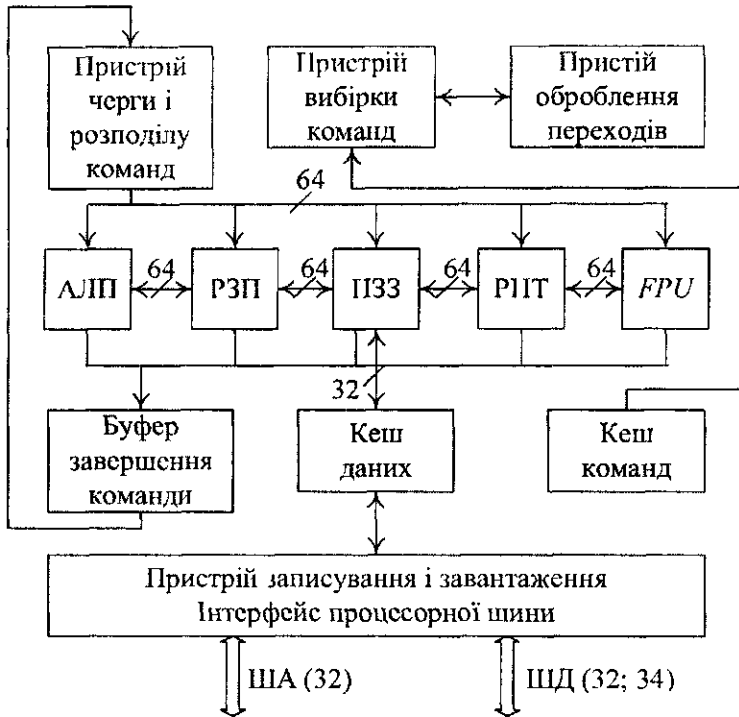


Рис. 10.35. Структура процесора Power PC 603

Оскільки *Power PC 603* є суперскалярним мікропроцесором, то він може видавати у виконавчі пристрої до трьох команд і завершувати їхнє виконання в одному машинному такті. Для збільшення продуктивності процесор допускає позачергове виконання команд. Крім цього, він забезпечує програмовні режими зниження споживаної потужності.

При обробленні даних команди розподіляються по п'яти виконавчих пристроях у заданому програмою порядку. Якщо залежності між командами відсутні, то вони виконуються миттєво.

Цілочисловий АЛП виконує більшість команд за один такт. Пристрій *FPU* має конвеєрну організацію і виконує операції з плаваючою комою як з одинарною, так і з подвійною точністю.

Команди умовних переходів обробляються пристроєм переходів. Якщо умови переходу доступні, то рішення про напрямок переходу приймається зразу; інакше виконання наступних команд продовжується з допущенням. Пересилки даних між кеш-пам'яттю даних і регістром РПТ обробляються пристроями завантаження і за-пам'ятовування.

Після закінчення виконання команди у виконавчому пристрої її результати направляються в буфер завершення команд. Після цього дані послідовно записуються у відповідний регістровий файл.

Контрольні запитання

1. Що таке процесор?
2. Охарактеризуйте архітектуру процесора.
3. Нарисуйте і поясніть структуру процесора.
4. Що таке мікропроцесор?
5. Назвіть характеристики мікропроцесора.
6. Що відноситься до статичних і динамічних параметрів мікропроцесора?
7. Як розділяють мікропроцесори за конструкцією?
8. Що входить в мікропроцесорні комплекти?
9. Назвіть типи мікропроцесорів.
10. Перечисліть техніко-економічні переваги мікропроцесорів.
11. Назвіть види архітектур мікропроцесора.
12. Охарактеризуйте програмно керований обмін інформацією.
13. Нарисуйте і поясніть структуру однокристалного восьмирозрядного мікропроцесора.
14. Охарактеризуйте список команд восьмирозрядного мікропроцесора.
15. Нарисуйте і поясніть структуру 16-розрядного мікропроцесора.
16. Поясніть особливості суперскалярних мікропроцесорів.
17. Охарактеризуйте *RISC*-архітектуру.
18. Охарактеризуйте *CISC*-архітектуру.
19. Нарисуйте і поясніть структуру мікропроцесора *Pentium*.
20. Нарисуйте і поясніть структуру процесора *Power PC 603*.

Розділ 11

Інтерфейси мікропроцесорних систем

11.1. Загальна характеристика мікропроцесорних систем

Мікропроцесорною системою називають сукупність інформаційно-обчислювальних засобів, куди входить один або декілька мікропроцесорів.

Для побудови МПС потрібен обмежений набір мікроелектронних ресурсів, оскільки реалізація більшості функцій покладається на програмне забезпечення. Таким чином, МПС — це цифрові системи оброблення інформації і керування, функціональні можливості яких визначаються програмним забезпеченням, а взаємозв'язок із зовнішнім середовищем здійснюється ПП. На основі МПС реалізують мікрокомп'ютери, персональні комп'ютери, інформаційні мережі, системи штучного інтелекту та ін.

Як і комп'ютер, МПС містить такі пристрої:

- процесор, який виконує основні функції керування і оброблення інформації;
- запам'ятовуючі пристрої на напівпровідникових мікросхемах, призначені для зберігання програм і даних;
- пристрої введення-виведення, які забезпечують взаємозв'язок із зовнішніми пристроями.

Пристрої МПС об'єднані між собою внутрішньо-системним інтерфейсом і взаємодіють за адресним принципом (рис. 11.1).

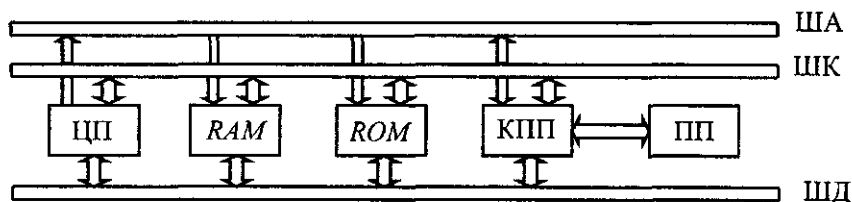


Рис. 11.1. Структура магістральної МПС

Конкретний склад і структура МПС визначається її призначенням. В однопроцесорних системах всі функції оброблення інформації і керування виконує один процесор. У багатопроцесорних системах є ряд рівноправних процесорів або один центральний і декілька підпорядкованих для розв'язання спеціалізованих задач (арифметики з плаваючою комою, операцій введення-виведення). У багатопроцесорних МПС може бути спільна системна пам'ять і локальна (резидентна) пам'ять окремих процесорів.

Мікропроцесор в МПС обробляє інформацію трьох типів — даних, адрес і команд програми. Для виконання цих функцій мікропроцесор містить: засоби оброб-

лення даних і адрес та їхнього зберігання в регістрах; схеми мікропрограмного керування, синхронізації та забезпечення режимів роботи.

Внутрішньосистемний інтерфейс найчастіше реалізують на основі спільної системної шини, по якій передаються адреси, дані й команди. Системна шина може мати окремі шини для адрес і даних або суміщені шини адрес/даних, які передаються мультіплексним способом.

У однопроцесорних МПС системна шина керується одним процесором (шина одного користувача). У багатопроцесорних МПС системна шина використовується в режимі розподілу часу. У них може застосовуватися багатощинна структура: спільна системна шина і декілька резидентних для окремих процесорів.

Мікропроцесорні системи поділяються на універсальні, спеціалізовані та багатопроцесорні (рис. 11.2).

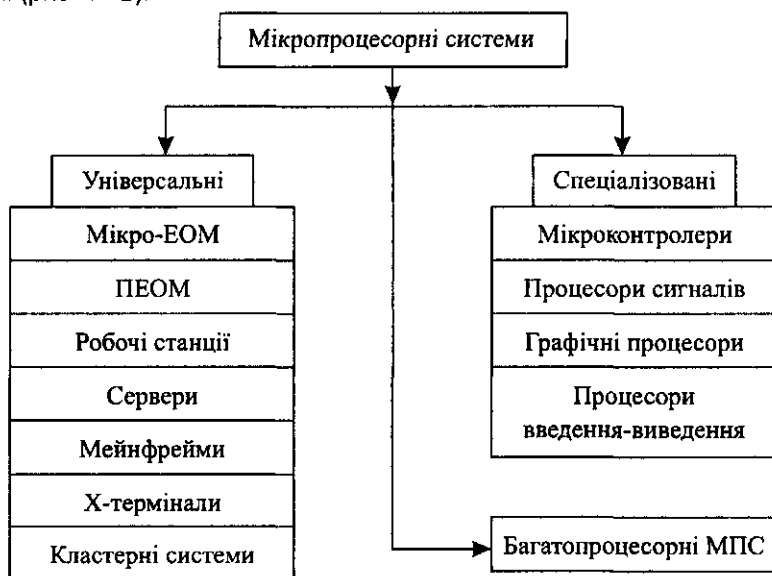


Рис. 11.2. Склад мікропроцесорних систем

Універсальні МПС орієнтовані на розв'язання широкого класу задач. Найбільш поширеними є мікро-ЕОМ — комп'ютер малих розмірів на основі мікропроцесора, напівпровідникової пам'яті та засобів зв'язку із зовнішнім середовищем. При настільному виконанні мікро-ЕОМ має єдину несучу конструкцію, пульт керування і джерела живлення. Ряд одно- або двоплатних мікро-ЕОМ вбудовують у технічні системи, в яких вони виконують функції керування, контролю та діагностики. На основі модифікацій мікро-ЕОМ створені такі комп'ютери:

- персональні (ПЕОМ), розраховані на масового користувача;
- робочі станції — автоматизовані робочі місця для конструкторів;
- сервери — виконують частину функцій з обслуговування групи користувачів (розподіл ресурсів пам'яті, принтерів, баз даних);
- мейнфрейми (мейнфрейм — синонім поняття "великий універсальний комп'ютер");
- X-термінали — комбінації бездискових робочих станцій і стандартних терміналів;

- кластерні системи — об'єднання машин з єдиним механізмом керування і програмного забезпечення. Кластерні МПС дуже надійні в роботі — при відмові одного процесора його функції перерозподіляються між іншими процесорами. Загалом кластерні системи забезпечують розподіл ресурсів, високу готовність, зручність розширення конфігурації.

Багатопроцесорні МПС призначені для розпаралелювання обчислювального процесу, що забезпечує високу продуктивність системи.

Спеціалізовані МПС орієнтовані на вирішення спеціалізованих задач керування чи оброблення інформації в складі технічних систем (мікрокомп'ютери), цифрового оброблення сигналів (процесори сигналів), оброблення графіки (графічні процесори). Їхні апаратні засоби реалізують у виді одноплатних конструкцій або наборів плат, які об'єднані внутрішньосистемним інтерфейсом.

Універсальні МПС орієнтовані на розв'язання широкого кола обчислювальних задач. Працюють вони в автономному режимі, містять широкий набір ПП.

Багатопроцесорні системи забезпечують функціонування багатьох процесорів під спільним керуванням.

Основними технічними характеристиками МПС є: розрядність, ємність пам'яті, продуктивність, число зовнішніх пристроїв та їхня пропускна здатність, функції системи і склад програмного забезпечення.

11.1.1. Програмне забезпечення мікропроцесорних систем

Програмне забезпечення МПС сприяє ефективному функціонуванню і використанню системи в різних областях науки і техніки. Основними складовими програмного забезпечення є операційні системи (ОС), пакети прикладних задач (ППП) і комплекси програм технічного обслуговування (КПТО).

Основними компонентами ОС є (рис. 11.3):

- керуюча програма, яка планує ресурси МПС, забезпечує взаємодію із зовнішнім середовищем, виявляє несправності технічних засобів;
- програма обслуговування, яка виконує допоміжні функції (копювання даних, підготовка носіїв даних та ін.);
- засоби генерацій, які настроюють МПС на конкретний склад і конфігурацію технічних засобів, та програми забезпечення необхідних режимів роботи.

При описуванні роботи керуючої програми важливим є поняття процесу оброблення даних (задач), що є сукупністю дій однієї чи декількох програм над розподіленими ресурсами процесорів, пам'яті, периферії.

Основними режимами обробки даних є режими розподілу часу, реального часу (в темпі порівняно зі швидкістю процесу), інтерактивний, діалоговий та пакетної обробки. Основними поняттями системи програмування є мова програмування, транслятор і компілятор програм, програмний модуль.

Мова програмування — це формалізована мова представлення програм. Важливими мовами програмування є машинна, асемблер, проблемно орієнтована. Машинна мова представляє програми у формі, готовій до виконання. Мова асемблера (автокод чи мнемокод) машинно орієнтована: її оператори аналогічні командам конкретної машини. Проблемно-орієнтовані мови призначені для вирішення визначеного класу задач (мови Паскаль, Турбо-Паскаль, Си, Пролог та ін).

Трансляція програм — це перетворення програм з однієї мови програмування в рівноцінну на іншій мові. **Компіляцією програм** називають трансляцію її на мову, близьку до машинної. Частковим видом компіляції є асемблювання — компіляція програм з мови асемблера. Форма представлення програми після компіляції називається об'єктним кодом.



Рис. 11.3. Склад програмного забезпечення МПС

Засоби, переважно програмні, які виконують трансляцію чи компіляцію, називаються відповідно трансляторами чи компіляторами. Основними видами компіляторів є крос-компілятор (транлює з автокоду однієї МПС в об'єктний код другої), препроцесор (перетворює початкову програму перед наступною компіляцією), налагоджувальний компілятор (засоби специфікації різних процедур налагоджування).

Пакети прикладних програм — це функціонально завершений комплекс програмних засобів, орієнтований на розв'язання визначеного логічно цілісного класу задач. Вони розвивають ОС в прикладному напрямку.

Пакети прикладних програм класифікують за ознаками:

- областю використання і класом задач, які вирішуються. Це ППП, які розширюють можливості ОС (забезпечують нестандартні режими експлуатації МПС), загального призначення (програми для розв'язання прикладних науково-технічних і економічних задач);
- орієнтацією на визначений метод чи процедуру обробки (методорієнтовані, технологічно чи проблемно орієнтовані);

- способом реалізації і принципами функціонування (бібліотеки підпрограм, спеціальні мови, програмні системи).

Методо-орієнтовані пакети програм призначені для розв'язання задач числового аналізу, статистичних задач. До проблемно орієнтованих відносяться пакети для розв'язання задач планування, оперативного керування, матеріально-технічного постачання та ін.

Бібліотека прикладних програм — це набір окремих програм, призначених для розв'язання деякого класу задач; кожна програма з бібліотеки має самостійне значення і застосовується для розв'язання нескладних задач.

Програмна система призначена для розв'язання порівняно великої за об'ємом типової задачі.

Комплекс програм технічного обслуговування забезпечує процедуру контролю роботи і діагностики МПС.

11.1.2. Поняття архітектури мікропроцесорної системи

Архітектура МПС в широкому розумінні включає поняття багаторівневої організації системи, в тому числі процесора, пам'яті, системної магістралі і засобів введення-виведення даних. Стосовно МПС багаторівнева архітектура визначається як розподіл функцій системи, точніше — окреслення границь між рівнями.

Архітектура першого рівня характеризує функції, які виконують МПС та засоби її взаємодії із зовнішнім середовищем: мова оператора, графічний інтерфейс, керування завданнями і програмним забезпеченням.

У програмному забезпеченні виділяють рівні логічного керування ресурсами (базою даних, файлами, віртуальною пам'яттю, мережною телеобробкою) та фізичного (зовнішньою і оперативною пам'яттю та процесами, що здійснюються в МПС). Другий рівень відображає межу між програмним забезпеченням і апаратурою, а потім — розподіл функцій між окремими частинами фізичної системи, наприклад:

- центральним процесором і каналами введення-виведення;
- каналами і контролерами зовнішніх пристроїв;
- контролерами і зовнішніми пристроями (терміналами, модемами, НГМД та ін.).

Архітектура таких рівнів часто називається архітектурою фізичного введення-виведення. У вузькому розумінні архітектура МПС — це архітектура набору команд. Вона служить межею між апаратурою та програмним забезпеченням і являє собою частину системи, яку бачить програміст чи розробник компіляторів.

На сучасному етапі розвитку МПС використовують дві основні архітектури наборів команд:

- з повним набором команд *CISC* (*Complete Instruction Set Computer*);
- із скороченим набором команд *RISC* (*Reduced Instruction Set Computer*).

Архітектура *CISC* є стандартом для світового ринку мікрокомп'ютерів, виготовлених фірмою Intel.

Для *CISC*-мікропроцесорів характерні:

- велика кількість машинних команд, деякі з них завантажені семантично аналогічно операторам високорівневих мов програмування. Команди виконуються за багато тактів;

- велика кількість методів адресації й форматів команд та даних різної розрядності. Переважно використовуються двоадресні формати команд,
- порівняно мала кількість регістрів загального призначення (від восьми до 16),
- широке використання мікрокоманд, конвеєризація команд і прогнозування переходів. Наявність команд обробки даних типу регістр-пам'ять,
- відсутність жорсткого обмеження на доступ операндів у пам'ять,
- роздільне використання кеша команд і даних та заповнення кеш-пам'яті в пакетному режимі.

Архітектура *RISC* є основою сучасних робочих станцій і процесорів. Вона застосована також при створенні серії суперкомп'ютерів фірми *Cray Research*.

Для архітектур *RISC*, *RISC I*, *RISC II* характерні

- невелике число простих команд, які апаратно реалізуються за один машинний такт,
- команди мають фіксовану довжину та формат, що спрощує логіку їхнього декодування,
- достатньо великий регістровий файл (32 і більше регістрів). Це забезпечує зберігання значного об'єму даних в регістрах, що спрощує роботу компілятора по розподілу регістрів під дані,
- як правило, використовується трьоадресний формат.

В останніх розробках фірми *Intel* (*Pentium*, *Pentium II*) використовують ідеї *RISC*-мікропроцесорів, тому відмінності між *RISC*- і *CISC*-архітектурами поступово стираються.

11.2. Загальна характеристика інтерфейсів

Сучасні МПС мають магістрально-модульну організацію, яка основана на принципах агрегування і уніфікації. **Модульність (агрегування)** — це розділення МПС на прості функціонально і конструктивно закінчені блоки, які називаються модулями, наприклад модуль мікропроцесора, модуль пам'яті та ін.

Уніфікація полягає в оптимізації складу модулів, зв'язків між ними та в їхніх конструктивних оформленнях.

Зв'язок пристроїв МПС один з одним здійснюють за допомогою сполучень, які називаються інтерфейсами. **Інтерфейс** — це сукупність інформаційно-логічних і конструктивних засобів і вимог (правил), які забезпечують оптимальний алгоритм взаємодії всіх модулів МПС.

Інформаційно-логічні вимоги визначають структуру і склад ліній і сигналів, способи кодування та формати даних, адрес і команд, протоколів обміну для різних режимів і фаз роботи. Вони безпосередньо впливають на пропускну здатність, надійність обміну та апаратні витрати.

Електричні вимоги задають необхідні статичні та динамічні параметри сигналів на сигнальних лініях інтерфейсу: рівні напруг, тривалість фронтів, навантажувальна здатність, завадостійкість та ін.

Конструктивні вимоги вказують на тип з'єднувальних елементів та розподіл ліній по їхніх контактах, геометричні розміри плат, каркаса та інші ознаки.

Сигнальна лінія — це провідник (електричний ланцюг), який фізично з'єднує джерело і приймач інформації. Сукупність сигнальних ліній, по яких передають сигнали однакового функціонального призначення, називають **шиною**. Розрізняють шини даних, адреси і керування.

Інтерфейс має забезпечувати:

- побудову машин із змінним складом обладнання (змінною конфігурацією, відкритістю архітектури);
- паралельне у часі виконання програм і процедур введення-виведення;
- збільшення швидкості обміну інформацією;
- спрощення і стандартизацію програмування операцій введення-виведення та їхню незалежність від особливостей периферійних пристроїв;
- автоматичне розпізнавання і реакцію ЦП на різноманітні ситуації в ПП (готовність пристрою, відсутність носія даних, порушення нормальної роботи).

Особливо актуальним є розв'язання цих задач для машин великої продуктивності, які містять сотні різних зовнішніх пристроїв.

Периферійні пристрої з'єднуються з інтерфейсом МПС за допомогою контролерів КПП, які забезпечують виконання кожним ПП своїх специфічних функцій.

Термін "інтерфейс" застосовують як до апаратури МПС, так до її програмного забезпечення.

11.2.1. Класифікація інтерфейсів

Інтерфейси класифікують за такими основними ознаками.

За функціональним призначенням інтерфейси розподіляються на такі типи:

- внутрішні — внутрішньооплатні, міжплатні та системні;
- зовнішні — для периферійних пристроїв, для локальних мереж, для розподілених систем керування (рис. 11.4).



Рис. 11.4. Класифікація інтерфейсів за функціональним призначенням

За напрямком обміну інформацією розрізняють такі інтерфейси:

- симплексні — обмін в одному напрямку (рис. 11.5, а);
- напівдуплексні — почерговий обмін у двох напрямках (рис. 11.5, б);
- дуплексні — одночасний обмін у двох напрямках (рис. 11.5, в);
- мультиплексні — обмін реалізується за допомогою спільної магістралі (шини), в якій в кожний момент часу взаємодіють джерело і приймач інформації (рис. 11.5, г).

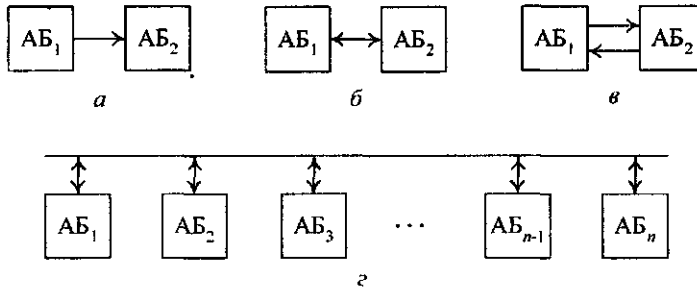


Рис. 11.5. Напрямки обміну інформацією:

a — симплексний; *b* — напівдуплексний; *c* — дуплексний; *d* — мультиплексний

Всі пристрої (модулі), які підключаються до каналу передачі даних, називаються абонентами (АБ).

За структурою зв'язків між абонентами МПС розрізняють інтерфейси з радіальним, кільцевим, каскадним і магістральним підключеннями (рис. 11.6).

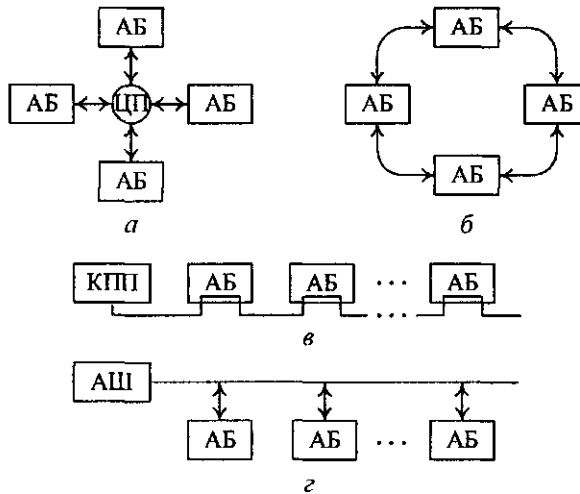


Рис. 11.6. Структура з'єднань абонентів:

a — радіальна; *b* — кільцева; *c* — каскадна; *d* — магістральна

У радіальних інтерфейсах (рис. 11.6, а) до ЦП за допомогою двонаправлених шин і заданого пріоритету підключаються абоненти (робочі станції, віддалені периферійні пристрої, схеми промислової автоматики). Повідомлення між абонентами передаються через ЦП, який виконує функції концентратора та забезпечує незалежність і паралельність роботи абонентів. Радіальний інтерфейс — логічно простий, але вимагає великих апаратних затрат. Крім цього, його живучість залежить від надійності ЦП.

У кільцевому інтерфейсі (рис. 11.6, б) кожний абонент зв'язаний з двома сусідніми. В кільці можуть одночасно циркулювати декілька повідомлень від джерел до приймачів на основі заданих способів адресації і керування. При розширенні МПС додаткові модулі включаються в кільце системи. Недоліком кільцевих інтерфейсів є складність взаємодії абонентів.

Каскадні інтерфейси (рис. 11.6, в) мають ланцюгове підключення абонентів. Абоненти обслуговуються в порядку їхнього підключення до ліній інтерфейсу. Каскадні інтерфейси характеризуються малим числом ліній і обмеженням за швидкістю.

У магістральних інтерфейсах (рис. 11.6, г) використовується колективна шина, яку також називають магістраллю. Інформація, яка передається по магістралі, доступна всім абонентам, які до неї підключені. Звичайно у кожний момент часу тільки один абонент може бути джерелом інформації. Пріоритет абонентів визначається арбітром шини (АШ). Магістральний інтерфейс є гнучким, економічним і використовується в більшості системних інтерфейсів МПС.

За способом передачі інформації в часі розрізняють такі інтерфейси:

- синхронні — з фіксованою тривалістю операцій обміну;
- асинхронні — зі змінною тривалістю операцій обміну;
- синхронно-асинхронні — з комбінованим поєднанням двох способів.

За розрядністю даних, які передаються, розрізняють інтерфейси з паралельним (словами), послідовним (бітами) і паралельно-послідовним обмінами.

До класифікаційних ознак інтерфейсів МПС також відносять:

- організацію переривання (векторне чи послідовне опитування) та прямого доступу до пам'яті;
- спосіб арбітражу доступу модулів до шин (паралельний, послідовний, циклічний);
- число ліній адресації даних, керування, синхронізації;
- довжину і тип ліній зв'язку (провідники, радіоканал, волоконно-оптичний тип);
- максимальне число абонентів, які можуть одночасно підключатися до шини;
- спосіб адресації пристроїв периферії — із власним адресним простором (характерно для виробів фірми *Intel*) чи з відображенням на адресний простір пам'яті (використовується у виробках фірми DEC).

11.2.2. Системні інтерфейси

В МПС широко використовуються паралельні системні стандартні інтерфейси, в яких уніфікуються: формати команд і даних та процедури обміну; алгоритм функціонування; склад і типи ліній зв'язку; швидкодія передачі; конструктивні вимоги; допустимі відстані між модулями системи; можливість розширення і сумісність з попередніми інтерфейсами.

Для забезпечення високих швидкостей обміну інформацією використовують асинхронні мультиплексні інтерфейси з паралельним способом передачі інформації. До них відносяться: восьмирозрядні — *Microbus*; 16-розрядні — *Unibus*; *Q-bus*; *Multibus I*; 32-розрядні — *Vercabus*. Основні технічні характеристики цих інтерфейсів наведені в табл. 11.1.

Таблиця 11.1

Технічні характеристики	<i>Microbus</i>	<i>Z-bus</i>	<i>Unibus</i>	<i>Q-bus</i>	<i>Multibus I</i>	<i>Vercabus</i>
Число ліній:						
загальне	37	—	56	43	86	260
даних	8	8	16	16	16	32
адресних	16	—	18	—	20	35
керування	13		20	—	11	—

Таблиця 11.1. Продовження

Технічні характеристики	Microbus	Z-bus	Unibus	Q-bus	Multibus I	Vercabus
Швидкість передачі, Кбіт/с	10^3	—	$2 \cdot 10^3$	800	10^4	10^5
Довжина ліній, м	—	—	15	15	—	—
Число абонентів	—	—	20	15	—	—

Інтерфейс *Multibus I* (стандарт російський — И41, зарубіжний — IEEE-796) призначений для побудови мікро-ЕОМ ПЕОМ і зосереджених багатопроцесорних МПС Він забезпечує

- чотири операції обміну — записування в пам'ять чи порт, читання пам'яті чи порту,
- прямий доступ до пам'яті і оброблення переривань програми,
- використання двох незалежних адресних просторів для адресації комірок пам'яті і зовнішніх пристроїв (їхніх портів),
- функціонування на основі принципу — ведучий (здатчик) і ведений (виконавець)

Сигнали на лініях інтерфейса активні за низьких рівнів напруг, призначення ліній і сигналів на них збігаються

Магістраль містить такі основні функціональні групи ліній і сигналів на них (цифри дані у шістнадцятковій системі числення)

- передачі адреси $\overline{ADRE} - \overline{ADRO}$ з можливістю розширення до $\overline{ADRI7} - \overline{ADRI4}$ на додатковому роз'єднувачі Сигнал \overline{BHEN} дозволяє зчитування старшого байта даних, а сигнал \overline{ALE} дозволяє записування адреси у зовнішній реєстр-фіксатор,
- передачі даних $\overline{DATE} - \overline{DAT0}$ забезпечує двонаправлену передачу слів чи байтів даних Інформація на ШД може видаватися тільки задатчиком Лінії $\overline{PAR1}$ і $\overline{PAR2}$ призначені для передачі сигналів парності молодшого і старшого байтів даних На лінії $\overline{INX1}$, $\overline{INX2}$ подають сигнали заборони звернення до ОЗП чи ПЗП відповідно,
- керування передачею даних \overline{MWTC} , \overline{IOWC} — записування в пам'ять чи порт, \overline{MRDC} , \overline{IORC} — читання пам'яті чи порта, \overline{XACK} — підтвердження передачі,
- синхронізації й арбітражу пріоритетів \overline{BCLK} , \overline{CCLK} — синхронізація шини і системних модулів, \overline{BUSY} — шина зайнята, \overline{BREQ} , \overline{CBRQ} — запит і спільний запит шини, \overline{BPRN} , $\overline{BCR0}$ — вхід і вихід дозволу пріоритету,
- керування станами системи, \overline{INIT} — початкове установлення, \overline{HALT} — зупинка, $\overline{AUX RESET}$ — додаткове скидання, \overline{WAIT} — очікування,
- керування електроживленням і його контроль \overline{ACLO} — зниження напруги в мережі чи її несправність, \overline{PFIN} — переривання при несправності електроживлення, \overline{MPRQ} — захист пам'яті

Електроживлення: плюс 5 В — дев'ять ліній живлення; плюс 12 В — три лінії додаткового живлення; плюс 12 В, мінус 12 В — по дві лінії додаткового живлення; *GND* — 23 лінії заземлення.

До інтерфейсів *Multibus I* відносять системну магістраль персонального комп'ютера *IBM PC/AT-bus*.

Інтерфейс *Multibus II* використовують у багатопроцесорних 16- і 32-розрядних МПС. Він містить три основні і три додаткові магістралі.

Основна магістраль системи — паралельна система *iPSB (Parallel System Bus)* призначена для організації обміну даними з продуктивністю до 40 Мбайт/с. Магістраль має п'ять груп ліній, можливий обмін восьми-, 16-, 24- і 32-розрядними даними. Адреси передаються по 32-розрядній мультиплексній ШАД.

Магістраль *iLBX* забезпечує швидкодіючий локальний обмін даними з пам'яттю. Асинхронна немультимплексна шина *iLBX* призначена для розширення ШВВ. Асинхронна мультимплексна магістраль *MDMA* забезпечує прямий доступ до віддаленої пам'яті.

Магістраль *iSSB* призначена для створення послідовної системної шини. Локальна шина зв'язку з модулями промислової телекерованої системи використовує магістраль *BitBus*.

У наш час більшість стандартних системних інтерфейсів МПС оптимізовані для конкретних типів мікропроцесорів.

11.3. Інтерфейсні мікросхеми

У МПС широко використовують інтерфейсні мікросхеми, які реалізують типові функції, а саме: генерацію тактових імпульсів, запам'ятовування адрес, забезпечення двонаправленого обміну даними між модулями системи, вироблення сигналів керування записуванням — зчитуванням пам'яті чи в портах, арбітражу, пріоритетного доступу до системної шини багатьох мікропроцесорів та ін.

У складі МПК КР580 і К1810 є значний набір інтерфейсних мікросхем (аналогів виробів фірми *Intel*); розглянемо деякі з них.

11.3.1. Генератор тактових імпульсів

Генератор тактових імпульсів забезпечує синхронізацію роботи МПС. У складі МПК К1810 є мікросхема К1810ГФ84 (далі ГФ84), яка призначена для формування серій синхроімпульсів, сигналів скидання та готовності системи на основі мікропроцесора ВМ86.

Генератор ГФ84 характеризується: схемотехнологією — ТТЛШ; частотою синхронізації — до 80 МГц; напругою живлення — плюс 5 В і споживаною потужністю — 0,7 Вт; амплітудою імпульсів — 4,5 В (рис. 11.7).

Мікросхема ГФ84 містить схеми задаючого генератора *G*, дільники частоти на три *DIV3* і два *DIV2*, які разом з логічними елементами виробляють прямокутні імпульси *CLK*, *PCLK* і синусоїдальний сигнал *OSC*. Вони призначені відповідно для тактування роботи мікропроцесора ВМ86, портів периферії чи зовнішніх пристроїв системи. Сигнали синхронізації можуть формуватися з коливань основної частоти кварцового резонатора, підключеного до входів *X1* і *X2* (рис. 11.8).

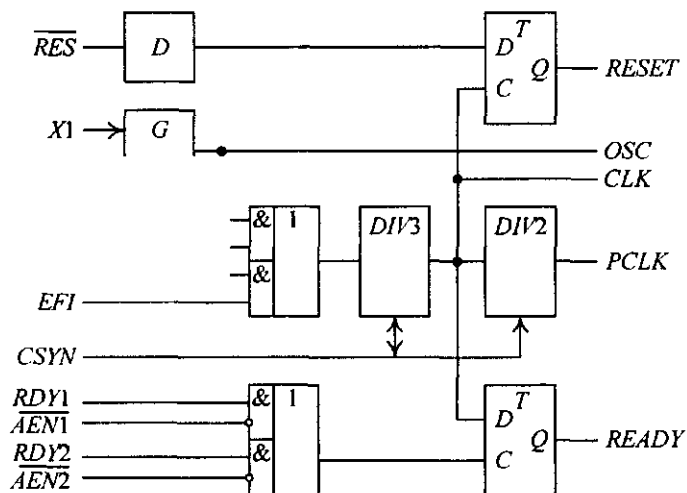


Рис. 11.7. Структура ГТІ К1810ГФ84

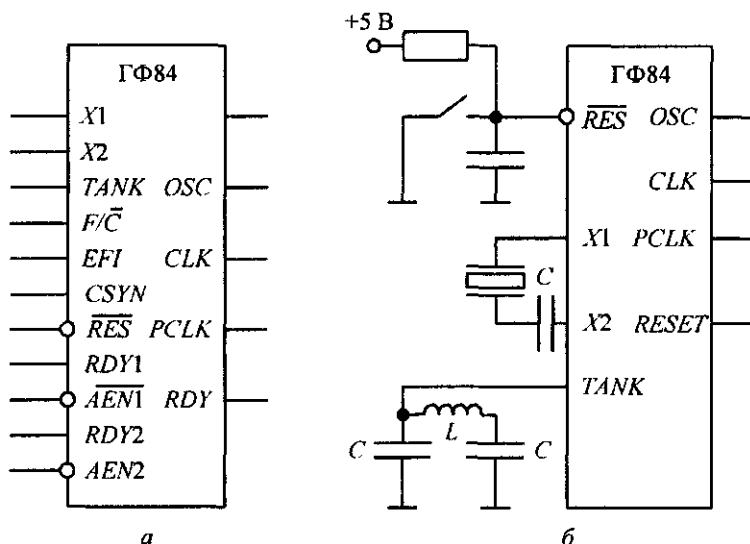


Рис. 11.8. Генератор тактових імпульсів ГФ84:

а — умовне графічне позначення; б — схема підключення кварцового резонатора

За допомогою LC-фільтра, з'єднаного із входом *TANK*, можна змінити частоту коливань резонатора.

Генератор може синхронізуватися і від зовнішнього джерела сигналів, яке підключається до входу *EFI*. Режим синхронізації задають рівнем сигналу на вході *F/C*: при високому — від зовнішнього генератора, при низькому — від кварцового генератора. В обох режимах синхронізації частота сигналів *CLK* вдвічі більша частоти *PCLK*, але в три рази менша частоти *OSC* чи зовнішнього генератора (рис. 11.9, а).

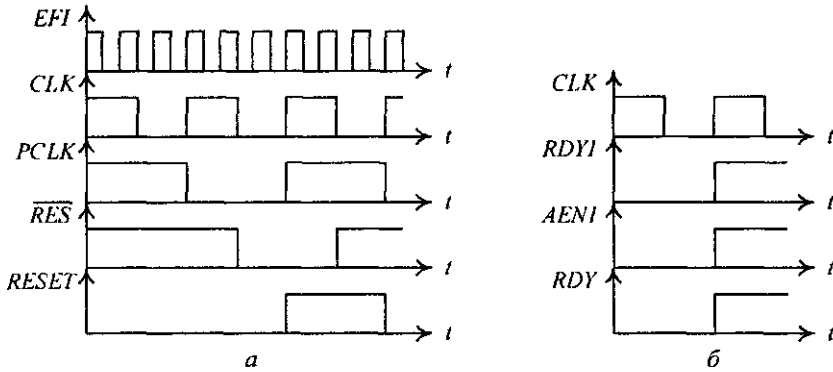


Рис. 11.9. Часові діаграми сигналів: а — синхронізації; б — готовності

Вхід *CSYN* служить для синхронізації декількох ГТІ системи: якщо *CSYN* = 1, то на виходах *CLK* і *PCLK* устанавлюються високі рівні; при *CSYN* = 0 на виходи *CLK* і *PCLK* поступають синхросерії.

Схема формування сигналу скидання *RESET* має на вході тригер Шмітта, а на виході — *D*-тригер, який формує фронт сигналу *RESET* за спадом *CLK*. Звичайно до входу *RES* підключається *RC*-ланцюг, який забезпечує автоматичне формування сигналу скидання при включенні джерела напруги.

Схема формування сигналу готовності до обміну даними *RDY* побудована з урахування шини *Multibus*. Вона має дві пари однакових сигналів *RDY1*, *AEN1* та *RDY2*, *AEN2*, об'єднаних схемою І ЧИ:

$$RDY = RDY1 \cdot AEN1 \vee RDY2 \cdot AEN2.$$

D-тригер формує фронт сигналу *RDY* за спадом *CLK* і подає його на вхід готовності мікропроцесора.

11.3.2. Буферні регістри та двонаправлені шинні формувачі

Буферні регістри та двонаправлені шинні формувачі забезпечують в МПС відповідно тимчасове зберігання (фіксацію) адреси з виходів ЦП та двонаправлений обмін даними між локальною і системною шинами. У складі МПК КР580 для виконання цих функцій є регістр КР580ІР82 (далі ІР82) та шинний формувач КР580ВА86 (далі ВА86).

Восьмирозрядний буферний регістр ІР82

Восьмирозрядний буферний регістр ІР82 з трьома станами характеризують такі параметри: схемотехнологія — ТТЛШ; число транзисторів на кристалі — 524; напруга живлення — плюс 5В і споживана потужність 1Вт.

Регістр ІР82 має вісім тригерів, стробоване записування даних у які відбувається по входах *D17–D10* при значенні сигналу *STB* = 1 (рис. 11.10).

Прямі виходи *D07–D00* регістра підключені до буферних схем *SW* з трьома станами.

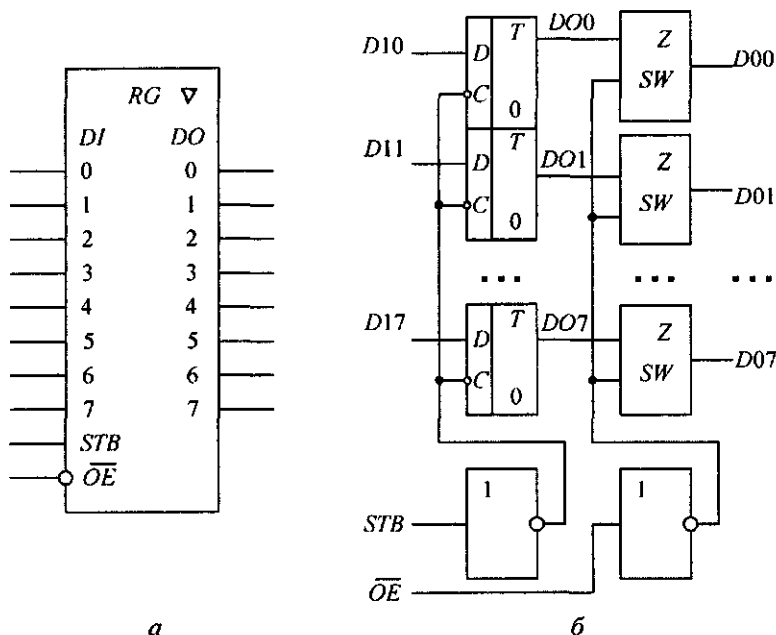


Рис. 11.10. Буферний регістр IP82: а — умовне позначення; б — функціональна схема

При значенні сигналу $\overline{OE} = 0$ буфери відкриваються і дані передаються на вихід. Якщо $\overline{OE} = 1$, то буфери устанавлюються в Z-стан. Сигнал \overline{OE} не впливає на стан тригерів та функцію записування інформації.

Шинний формувач ВА86 забезпечує двонаправлений обмін даними між локальною та системною шинами, підсилення сигналів та відключення від шини у визначені моменти часу. Мікросхема ВА86 характеризується такими параметрами: ТТЛШ схемотехнологією, числом транзисторів на кристалі — 567; напругою живлення — плюс 5 В і споживаною потужністю — 1 Вт.

Шинний формувач (ШФ) має двонаправлені входи — виходи А7–А0 та В7–В0, вхід Т для керування напрямком обміну і вхід \overline{OE} для зняття Z-стану визначеного напрямку переходу (рис. 11.11).

Кожний розряд формувача містить дві схеми SW з трьома станами кожна. При $\overline{OE} = 1$ всі SW-схеми знаходяться в Z-стані, при $\overline{OE} = 0$ і $T = 0$ обмін даними відбувається в напрямку від А до В. При $\overline{OE} = 0$ і $T = 1$ обмін даними відбувається від В до А.

Формувач споживає від ЦП струм біля одного міліампера, а може віддавати в системну шину десятки міліампер.

11.3.3. Контролер системної шини

Для керування обміном даними в МПС використовується системний контролер. У складі МПК К1810 є мікросхема К1810ВГ88 (далі ВГ88), яка реалізує функції системного контролера шини (КШ). Мікросхема ВГ88 має такі технічні характеристики: схемотехнологія — ТТЛШ; частота синхронізації — 8 МГц; напруга живлення — плюс 5 В; споживана потужність — 0,85 Вт.

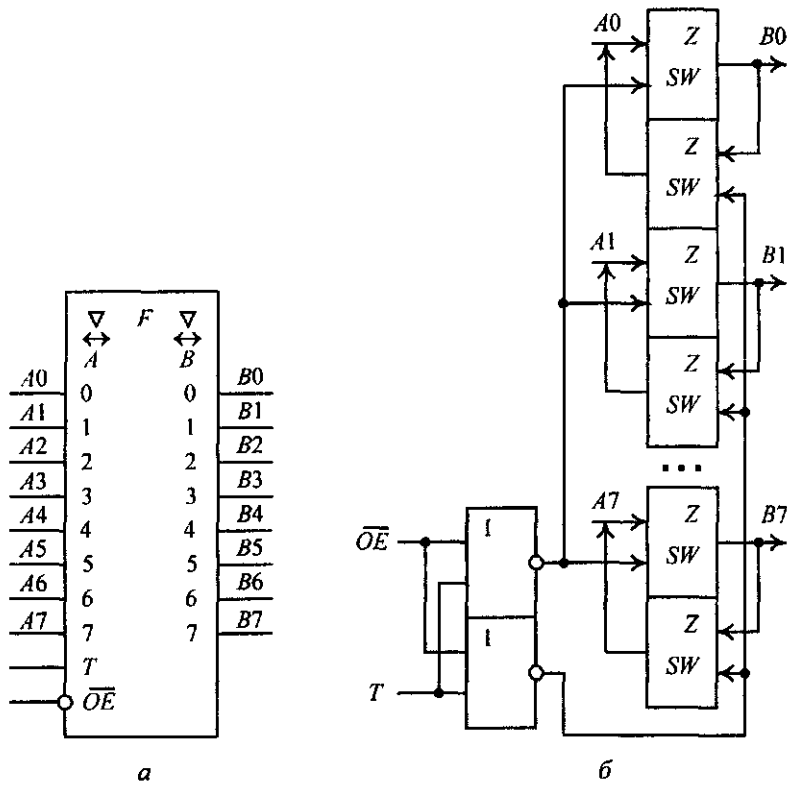


Рис. 11.11. Шинний формувач ВА86: а — умовне позначення; б — функціональна схема

Контролер шини реалізує в МПС такі функції: дешифрацію станів мікропроцесора, формування керуючих і командних сигналів для локальних і системних шин; вироблення сигналів керування ШФ, буферними регістрами і контролером переривання. До складу КШ входять (рис.11.12): дешифратор станів ДшС, схема керування СхК, формувач командних Ф1 і керуючих Ф2 сигналів.

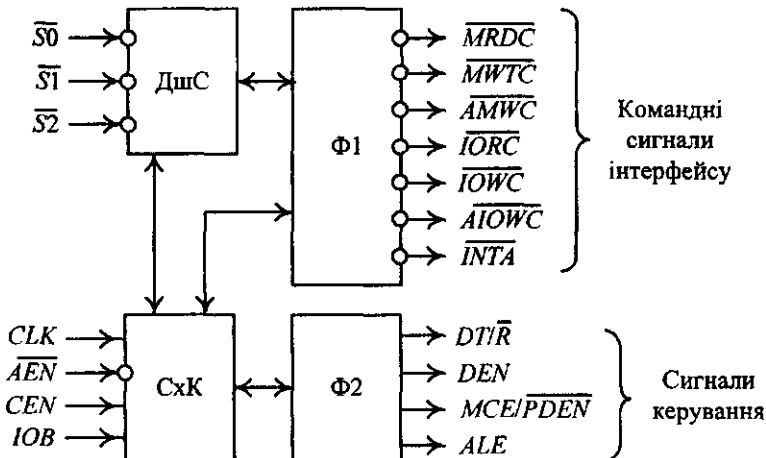


Рис. 11.12. Структура контролера шини ВГ88

Функціональне призначення вхідних сигналів КШ: $\overline{S2}$, $\overline{S1}$, $\overline{S0}$ — стани мікропроцесора; CLK — синхронізація від ГТІ; \overline{AEN} — дозвіл видачі сигналів за командними виводами; CEN — дозвіл видачі командних сигналів і сигналів керування \overline{DEN} , \overline{PDEN} ; IOB — керування режимом роботи.

Вихідні сигнали КШ за функціональними ознаками розподілені на командні сигнали інтерфейсу і сигнали керування.

До командних сигналів інтерфейсу відносяться: \overline{MRDC} — читання пам'яті; \overline{MWTC} — записування в пам'ять; \overline{IORC} — введення з порта; \overline{IOWC} — виведення в порт; \overline{AMWC} , \overline{AIOWC} — попереджуючі підготовчі сигнали записування в пам'ять чи виведення в порт; \overline{INTA} — підтвердження переривання.

До вихідних керуючих сигналів АШ відносяться: DT/\overline{R} — вибір напрямку передачі даних через ШФ; \overline{DEN} — дозвіл видачі даних; \overline{ALE} — строб записування адреси в буферний регістр; $\overline{MCE}/\overline{PDEN}$ — в режимі роботи із системною шиною використовується як строб читання номера веденого контролера переривань. При роботі з ШВВ використовується для керування станом "включено" ШФ.

Основною інформацією для роботи КШ є код стану $\overline{S2}$, $\overline{S1}$, $\overline{S0}$, який декодується дешифратором (табл. 11.2).

Таблиця 11.2

Код стану			Стан	Керуючі сигнали
$\overline{S0}$	$\overline{S1}$	$\overline{S2}$		
0	0	0	Підтвердження переривання	\overline{INTA}
0	0	1	Читання ПБВ	\overline{IORC}
0	1	0	Записування ПБВ	\overline{IOWC} , \overline{AIOWC}
0	1	1	Зупинка	—
1	0	0	Вибірка команди	\overline{MRDC}
1	0	1	Читання пам'яті	\overline{MRDC}
1	1	0	Записування в пам'ять	\overline{MWTC} , \overline{AMWC}
1	1	1	Холостий стан	—

Вхідні сигнали IOB , CEN , \overline{AEN} визначають два режими роботи контролера — із СШ і ШВВ. Режим роботи із СШ устанавлюється при $IOB = 0$, при цьому КШ формує командні сигнали і сигнали керування \overline{ALE} , \overline{DEN} , DT/\overline{R} фіксаторами адреси і ШФ.

Типова схема включення КШ в однопроцесорних системах на основі ЦП ВМ86 в максимальному режимі показана на рис. 11.13.

На входах формується постійні значення $\overline{AEN} = 0$ і $CEN = 1$, які дозволяють видачу командних сигналів і сигналів керування. На виході $\overline{MCE}/\overline{PDEN}$ за такого включення формується сигнал \overline{MCE} . Він використовується в МПС з каскадуванням контролерів переривань для визначення моменту передачі номера веденого кон-

тропера, який запитує переривання. Сигнал ALE визначає момент фіксації адреси в буферних регістрах IP82. Вихідні сигнали DT/\bar{R} і DEN використовуються для керування роботою ШФ. Сигнал DEN має високий рівень, тому він має інвертуватися перед поданням на вхід \overline{OE} мікросхеми BA86.

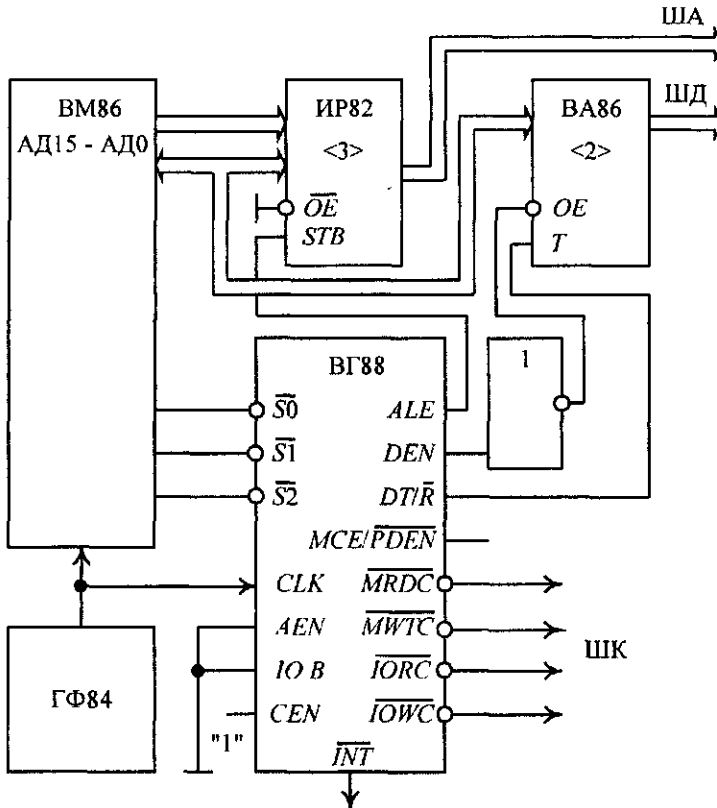


Рис. 11.13. Типова схема підключення ВГ88 до ЦП ВМ86

Системна шина в цьому випадку працює тільки з пам'яттю і ЦП одержує доступ до неї за сигналом AEN від арбітра шин, а командні сигнали для портів не використовуються.

11.3.4. Арбітр шин

У багатопроцесорній МПС пріоритетний доступ до СШ забезпечує АШ. У складі МПК К1810 є мікросхема К1810ВБ89 (далі ВБ89), яка реалізує функції арбітра.

В структуру АШ входять (рис. 11.14):

- схема арбітра АРБ і дешифратор станів ДшС;
- схема керування арбітражем СхК і блок шинного інтерфейсу БіФ.

В АШ входи і сигнали на них мають таке функціональне призначення:

- $\overline{S2}$, $\overline{S1}$, $\overline{S0}$ — сигнали станів мікропроцесора;
- CLK — вхід синхронізації;
- \overline{BCLK} — вхід системної синхронізації;

- \overline{INIT} — початкове устанавлення (скидання) АШ;
- \overline{LOCK} — блокування СШ;
- \overline{RESB} — вибір резидентної шини;
- \overline{IOB} — вибір режиму роботи з ШВВ;
- \overline{CRCK} — сигнал спільного блокування СШ;
- $AQST$ — будь-який запит;
- $\overline{SB/RB}$ — вибір системної чи локальної шини.

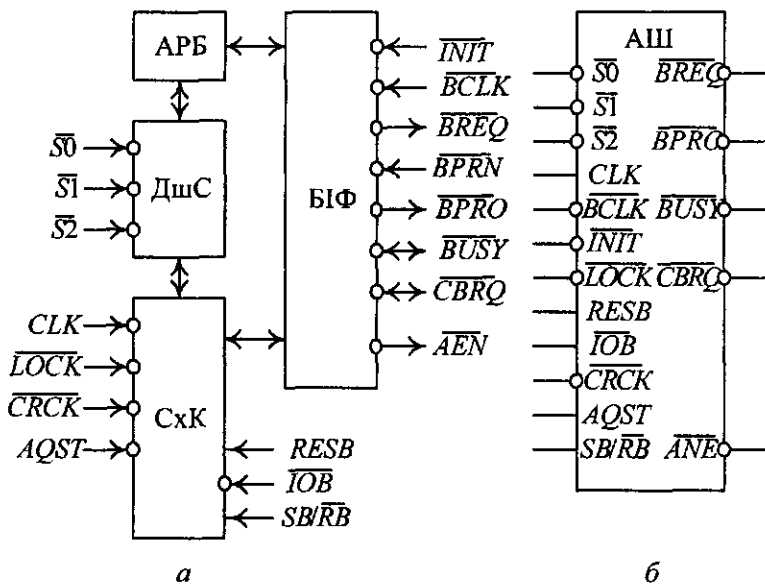


Рис. 11.14. Арбітр шин V589: а — структура; б — умовне графічне позначення

Функціональне призначення виходів і сигналів на них:

- $\overline{BREQ}, \overline{BPRO}, \overline{BUSY}, \overline{CBRQ}$ — використовуються для організації схеми визначення пріоритетів;
- \overline{AEN} — дозвіл СШ.

У багатопроцесорній МПС за допомогою АШ організують такі схеми оброблення пріоритетів: послідовні; паралельні з фіксованим пріоритетом; паралельні з циклічно змінюваними пріоритетами. У багатопроцесорній МПС кожний ЦП має свій АШ.

У схемі послідовного арбітражу пріоритетний вихід $\overline{BPQ0}$ арбітра з вищим пріоритетом підключається до входу \overline{BPRN} арбітра з нижчим пріоритетом (рис. 11.15)

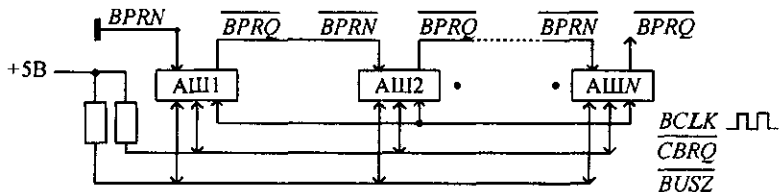


Рис. 11.15. Схема послідовного арбітра

Вхід \overline{BPRN} арбітра з найвищим пріоритетом підключається до землі. Якщо декільком АШ необхідний доступ до СШ, то сигнал $\overline{BPRQ} = 0$ у арбітра з найбільшим пріоритетом поступає на вхід \overline{BPRN} сусіднього арбітра і забороняє йому захват шини.

Схема послідовного визначення пріоритету найпростіша і не вимагає додаткових мікросхем. Але при такому з'єднанні пріоритет до останнього АШ приходиться із затримкою. Оскільки доступ до СШ має здійснюватися за один період сигналу \overline{BCLK} , то при частоті 10 МГц можна послідовно з'єднати не більше трьох АШ.

Схема паралельного визначення пріоритетів дозволяє підключитися до 16 арбітрів, для їхньої організації додатково використовуються пріоритетний шифратор і дешифратор (рис. 11.16).

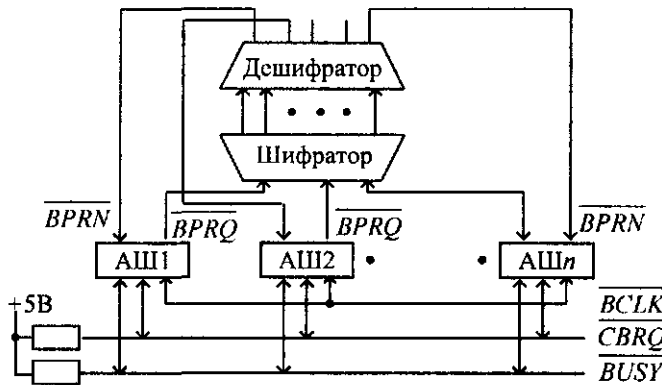


Рис. 11.16. Схема паралельного арбітражу

Сигнал з виходу запиту шини \overline{BPRQ} кожного АШ подається на вхід пріоритетного шифратора, який формує двійковий код номера запиту з найбільшим пріоритетом. Цей код декодується дешифратором і поступає на вхід \overline{BPRN} вибраного АШ і дозволяє йому захват шини.

Схема циклічного визначення пріоритетів аналогічна схемі паралельного арбітражу. Але вона має складніші додаткові мікросхеми, за допомогою яких пріоритети обробляються за чергою.

У всіх схемах арбітражу є лінія \overline{BUSY} , на яку виставляється низький рівень арбітром, який запитує СШ. Якщо СШ вільна, то арбітр, який одержав доступ до шини, формує сигнал $\overline{AEN} = 0$. Після цього для вибраного КШ дозволяється видавати на СШ командні сигнали з КШ, а також адресу і дані відповідно з фіксаторів і формувачів.

Залежно від конфігурації МПС і сигналів на входах \overline{IOB} , \overline{RESB} можна задавати чотири режими роботи з АШ. Сигнал на вході $\overline{SYSB}/\overline{RESB}$ виконує допоміжну роль — визначення умов доступу до СШ.

Режим роботи із СШ задають сигнали $\overline{IOB} = 1$, $\overline{RESB} = 0$, вхід $\overline{SB}/\overline{RB}$ ігнорується. При цій комбінації сигналів ЦП може підключитися тільки до однієї СШ (рис. 11.17).

При виконанні циклу звернень до СШ арбітр спочатку одержує пріоритетний доступ, потім формує сигнал $\overline{AEN} = 0$ і подає його на схему шинного інтерфейсу СШ1

(вона містить КШ, фіксатор адреси і формувач даних). Арбітр утримує СШ, використовує адресний простір пам'яті (тобто керується сигналами \overline{MRDC} і \overline{MWTC}).

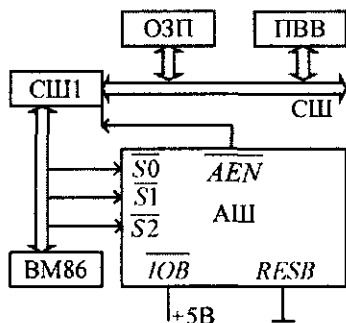


Рис. 11.17. Схема підключення АШ до системної шини

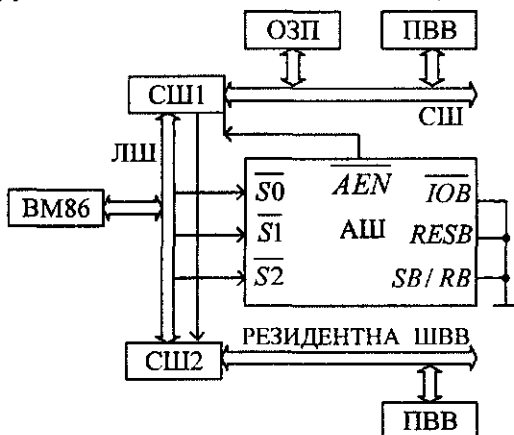


Рис. 11.18. Схема підключення АШ до СШ і резидентної ШВВ

Режим роботи із СШ пам'яті й резидентною ШВВ задають сигнали $\overline{IOB} = \overline{RESB} = 0$ і $\overline{SB/ \overline{RB}} = 0$. При цій комбінації сигналів ЦП підключається до СШ, яка обслуговує тільки пам'ять, і до резидентної ШВВ, що здійснює обмін даними з периферією (рис. 11.18)

У цьому режимі сигнал $\overline{AEN} = 0$ виробляється арбітром тільки при необхідності ЦП звертатися до пам'яті і наявності пріоритетного дозволу. Доступ до резидентної ШВВ надається процесору завжди при виконанні команд введення \overline{IN} і виведення \overline{OUT} сигналом $\overline{PDEN} = 0$. Схема СШ2, яка зв'язує локальну шину з резидентною ШВВ, не містить системного контролера ВГ88. Його функції виконує КШ в блоці СШ1, установлений в режим $\overline{IOB} = 1$. При цьому командні сигнали керування пам'яттю (\overline{MRDC} , \overline{MWTC}) поступають на СШ, а керування введенням-виведенням (\overline{IORC} , \overline{IOWC}), а також \overline{INTA} — на резидентну ШВВ.

Детальна схема розширеного ЦП в режимі обслуговування СШ і резидентної ШВВ показана на рис. 11.19.

Режим роботи із системною і резидентною шинами задають сигналами $\overline{IOB} = \overline{RESB} = 1$. Він використовується в МПС з системною і резидентною шинами і в кожній з них забезпечується доступ до пам'яті і ПВБ (рис. 11.20).

Сигнал \overline{AEN} , який забезпечує доступ до СШ, виробляється при $\overline{SB/ \overline{RB}} = 1$ (за умови, що АШ одержав дозвіл на доступ). Для формування сигналів доступу до системної $\overline{SB/ \overline{RB}} = 1$ або до резидентної $\overline{SB/ \overline{RB}} = 0$ шин використовують дешифратор адреси ДША, підключений до СШ.

Режим роботи із СШ пам'яті, резидентною шиною і резидентною ШВВ задають сигнали $\overline{IOB} = 0$ і $\overline{RESB} = 1$. Цю комбінацію сигналів використовують в МПС, де поряд із СШ, яка забезпечує доступ тільки до пам'яті, є дві резидентні шини. Одна з них здійснює доступ як до пам'яті, так і до ПВБ, друга — тільки до ПВБ.

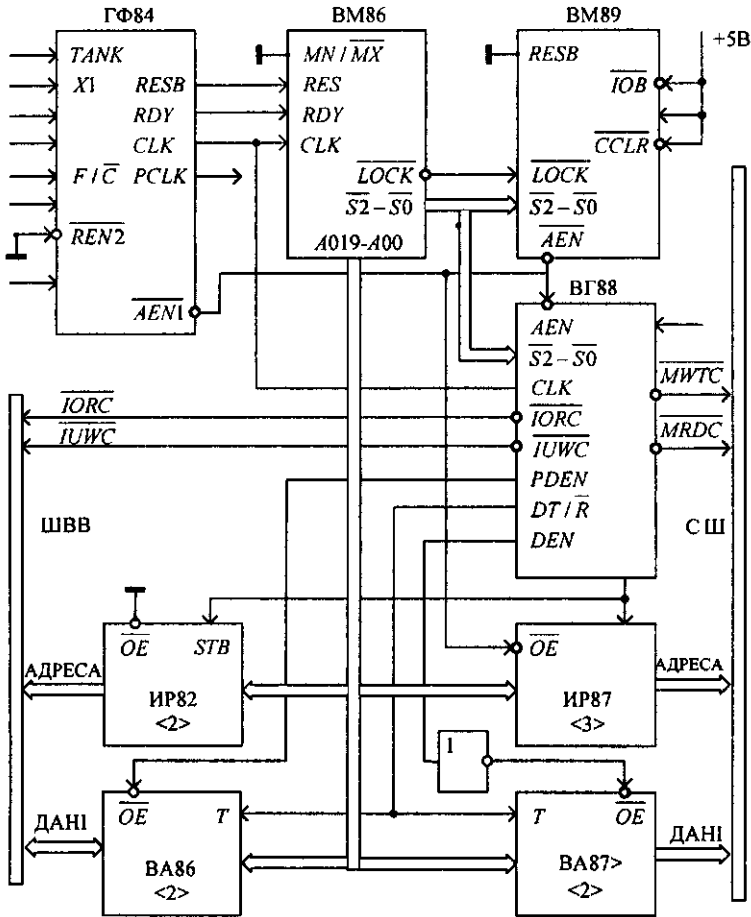


Рис. 11.19. Схема ЦП в режимі обслуговування СШ і резидентної ШВВ

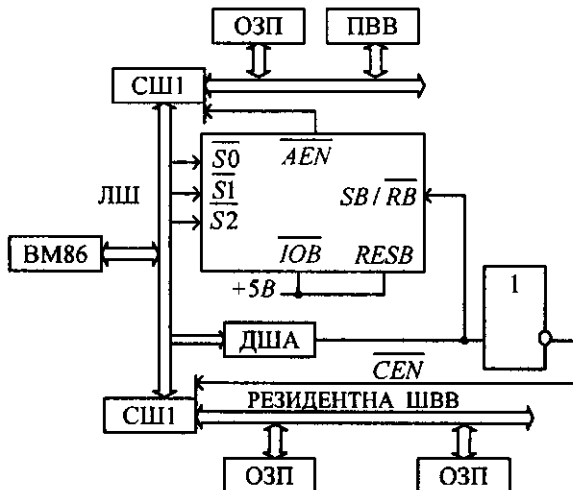


Рис. 11.20. Схема використання АШ із системою і резидентною шинами

11.4. Програмовні інтерфейсні контролери

11.4.1. Загальна характеристика програмовних інтерфейсних контролерів.

Периферійні пристрої підключаються до комп'ютера з допомогою спеціалізованих адаптерів або контролерів. **Адаптер** — це засіб сполучення деякого зовнішнього пристрою з визначеною шиною МПС. Контролер також виконує подібні функції, але він здатний на самостійні дії після одержання команд від машини. Складний контролер може мати і власний процесор. Сьогодні часто терміни "адаптер" і "контролер" вважають майже синонімами. Усі зовнішні інтерфейси мають у своєму складі адаптер або контролер.

Мікропроцесорні комплекти КР580 і К1810 мають такі програмовні інтерфейсні контролери (адаптери):

- контролер переривань — К1810ВН59 (ВН59);
- контролер ПДП — КР580ВТ57 (ВТ57);
- програмовний таймер — КР580ВІ53 (ВІ53);
- паралельний інтерфейс — КР580ВВ55 (ВВ55);
- послідовний інтерфейс — КР580ВВ51 (ВВ51).

11.4.2. Програмовний контролер переривань

Під час виконання поточної (фонової) програми в комп'ютері виникають заздалегідь невідомі події, які вимагають миттєвої реакції на них (аварія в системі, готовність ПП до передачі даних, ділення на нуль та ін.). Реакція полягає в перериванні фонової програми, переході до підпрограми обслуговування даної події та поверненні програми в попередній стан. Кожна подія, яка вимагає переривання, супроводжується сигналом — запитом переривання. Переривання — важлива архітектурна особливість машини, яка дозволяє ефективно організовувати роботу процесора за наявності декількох паралельних процесів.

Для технічної реалізації переривань використовують програмовні контролери переривань (ПКП). До ПКП відноситься мікросхема ВН59 (аналог модуля 8259) з такими характеристиками: схемотехнологія — n -МОН, напруга живлення — плюс 5 В, споживана потужність — 1 Вт. Структура ПКП ВН59 показана на рис. 11.21.

До складу ПКП ВН59 входять :

- двонаправлений восьмирозрядний буфер даних БД;
- блок керування (входи \overline{CS} , \overline{WR} , \overline{RD} , AD , \overline{INTA} , INT);
- регістри запитів переривань IRR (входи $IR7$ – $IR0$), керуючих слів RKC , стану ISR , маскування IMR ;
- блок каскадування (вхід SP , двонаправлені виходи $CAS3$ – $CAS0$);
- блок оброблення пріоритетів (БОП).

Програмовний контролер переривань сприймає запити на переривання від восьми ПП; за рахунок каскадування за допомогою виходів $CAS3$ – $CAS0$ число входів для запитів може збільшитися до 64.

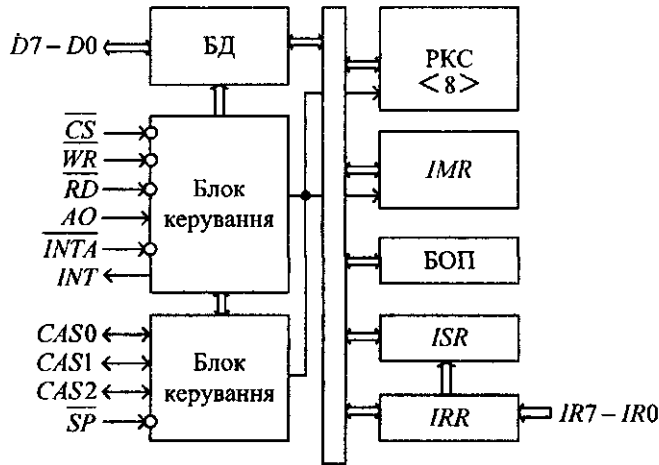


Рис. 11.21. Структура ПКП ВН59

Запити, які поступають на входи $IR7-IR0$, фіксуються в регістрі запитів переривання IRR . Блок оброблення пріоритетів визначає найбільший пріоритетний запит. Блок керування формує сигнал запити переривання INT , який поступає на такий же вхід мікропроцесора. Якщо в мікропроцесорі переривання дозволено, то він відповідає контролеру сигналом $INTA$, який поступає на такий же вхід ПКП. В результаті на ШД від КПК видається спочатку перший байт — код команди $CALL$ (перехід на підпрограму оброблення переривання), а потім другий і третій байти, які визначають адресу підпрограми. Після цього розряд регістра запитів IRR , на який поступив запит з вищим пріоритетом, обнуляється і встановлюється одиниця у відповідному розряді регістра станів ISR .

За допомогою завантаження керуючих слів ініціалізації ICW в режимі налаштування і операційних слів OCW при обслуговуванні запитів ПКП може програмуватися на такі режими обслуговування переривань:

- пріоритети зовнішніх пристроїв фіксовані;
- циклічний зсув пріоритетів;
- спеціального маскуванню;
- послідовного опитування ПКП.

У режимі з фіксованими (незмінними) пріоритетами IRO має найвищий пріоритет, а вхід $IR7$ — найнижчий. Програмовний контролер переривань завжди обслуговує запити з вищим пріоритетом. Якщо під час обслуговування одного із запитів приходять запит вищого пріоритету, то ПКП переключається на його обслуговування.

При циклічному зсуві пріоритети можуть динамічно змінюватися: входу після обслуговування присвоюється нижчий пріоритет (рис. 11.22).

У даному режимі після кожного зсуву необхідно завантажувати керуюче слово. Режим циклічного зсуву пріоритетів дозволяє усувати недоліки, коли при частих зверненнях до входів з вищими пріоритетами ігноруються запити з нижчими пріоритетами. Застосовування цього способу доцільне при обслуговуванні зовнішніх пристроїв з однаковими пріоритетами.

Запис одиниць у розряди регістра маски $IMR7$ – $IMR0$ забороняє приймання запитів на відомих входах $IRR7$ – $IRR0$.

У режимі спеціального маскуванню дозволяється переривання на входах з меншими пріоритетами, ніж запит, який обслуговується у даний час.

У режимі послідовного опитування мікропроцесор за допомогою ПКП послідовно опитує джерела запитів. Контролер виставляє на ШД інформацію, в якій в молодших розрядах байта вказується номер ПП з вищим пріоритетом, а одиниця в старшому розряді байта вказує на наявність запиту. У мікропроцесорі ця інформація обробляється програмою.

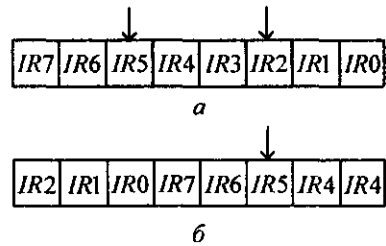


Рис. 11.22. Ілюстрація циклічного зсуву пріоритету: а — запити на входах $IR2$ та $IR5$; б — після обслуговування

11.4.3. Контролер прямого доступу до пам'яті

Режим прямого доступу до пам'яті забезпечує безпосередній обмін інформацією між ОП МПС і зовнішніми швидкодіючими пристроями, наприклад, НГМД, реалізація такого обміну досягається за допомогою ВІС програмовних КПДП.

Функції КПДП виконує мікросхема ВТ57 (аналог модуля 8257), спрощена структура якої показана на рис. 11.23. Технічні характеристики мікросхеми: схемотехнологія — n -МОН; напруга живлення — плюс 5 В і споживана потужність — 0,7 Вт.

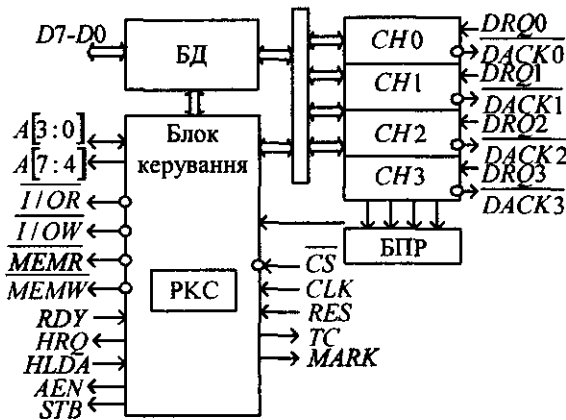


Рис. 11.23. Структура мікросхеми контролера ВТ57

Мікросхема контролера ВТ57 містить:

- двонаправлений БД;
- чотири незалежних канали (*channel*) $CH3$ – $CH0$;
- блок задання пріоритетів БПР;
- блок керування процесом програмування контролера та обміном між системою пам'яттю і зовнішніми пристроями.

Двонаправлений восьмирозрядний БД забезпечує обмін інформацією між мікропроцесором і КПДП в режимах програмування та перевірки.

Канали здійснюють безпосередній обмін даними між системною пам'яттю і зовнішніми пристроями. Кожний з каналів містить:

- 16-розрядний реєстр адреси *PAK*, за допомогою якого задають адресний простір до 64 Кбайт;
- 16-розрядний реєстр числа циклів (*РЦЦ*), причому старші два розряди визначають тип обміну: 00 — перевірка; 01 — записування в ОП даних від ПП; 10 — читання даних з ОП і пересилка їх до ПП; 11 — заборонена комбінація; реєстри числа циклів (без двох старших розрядів) забезпечують обмін даними блоками об'ємом до 16 Кбайт;
- схему запитів підтвердження, на входи якої поступають сигнали запиту захвату циклу *DRQ*, а на виході *DACK* формується сигнал підтвердження захвату.

БПР обслуговує канали з фіксованим пріоритетом (канал *CH0* має найвищий пріоритет, *CH3* — найнижчий пріоритет) або з циклічним — після обслуговування каналу йому присвоюється найнижчий пріоритет.

Блок керування контролера містить:

- схему приймання сигналів вибору мікросхеми \overline{CS} , синхронізації *CLK*, скидання *RES*, готовності від периферії *RDY* та підтвердження захвату циклу *HLDA* від мікропроцесора;
- буфери адреси *A7–A4* та *A3–A0* (двонаправлені);
- реєстр керуючого слова (РКС) і реєстр стану;
- схему вироблення вихідного сигналу захвату циклу *HRQ* для мікропроцесора, сигналів записування-читання пам'яті *MEMW*, *MEMR* та портів $\overline{I/O}$, $\overline{I/O}$ W;
- формувач адресних стробів *AEN*, *STB*, *ADS*;
- генератор сигналу кінця обміну "TC" та сигналу *MARK* — "до закінчення блоку, що передається, залишилося число циклів, кратне 128".

У режимі програмування інформацію в канали записують у такій послідовності: молодший, а потім старший байт адреси; молодший, а потім старший байт числа циклів (зменшено на одиницю) і типу обміну. Адресу реєстрів задають кодом *A3–A0*, як це показано в табл. 11.3. Записування двох байтів в реєстрах виконується командою *OUT* за однією адресою за допомогою спеціального тригера, який визначає спочатку молодший байт, а потім старший.

Таблиця 11.3

<i>A3</i>	<i>A2</i>	<i>A1</i>	<i>A0</i>	Реєстр	<i>A3</i>	<i>A2</i>	<i>A1</i>	<i>A0</i>	Реєстр
0	0	0	0	PA0	0	1	0	1	РЦ2
0	0	0	1	РЦ0	0	1	1	0	РА3
0	0	1	0	РА1	0	1	1	0	РЦ3
0	0	1	1	РЦ0	1	0	0	0	РСС
0	1	0	0	РА2					

Формат керуючого слова, показаний на рис. 11.24, записується в РКС.

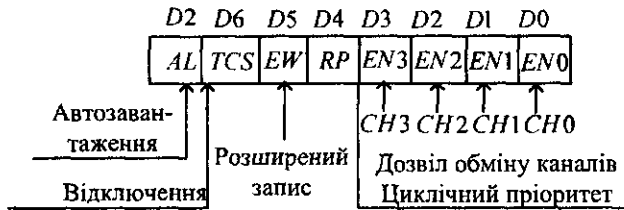


Рис. 11.24. Формат керуючого слова

Сигнали $EN3$ – $EN0$ при одиничних значеннях дозволяють обмін з відповідним каналом, а при нульових — забороняють. Сигнал установлює послідовність обслуговування каналів: $RP = 0$ — фіксований пріоритет; $RP = 1$ — циклічний пріоритет. Сигнал $EW = 1$ збільшує тривалість сигналів $MEMW$ і I/OW , які генерує КПДП; це узгоджує роботу системної пам'яті й периферії, що мають різні цикли звернень.

Коли значення сигналу $TSC = 1$, поява повідомлення TC в одному з каналів $CH0$ – $CH3$ скидає його сигнал дозволу $EN0$ – $EN3$ на обмін і цей канал відключається.

У режимі автозавантаження може працювати тільки канал $CH2$. Після появи сигналу кінця обміну "TC" в каналі $CH2$ він автоматично завантажує в свої регістри адреси і числа з регістрів каналу $CH3$. Це дозволяє організовувати пересилки БД з однаковими параметрами або з'єднувати декілька блоків з різними параметрами.

Схема підключення в МПС контролера ПДП, пам'яті й зовнішнього пристрою показана на рис. 11.25.

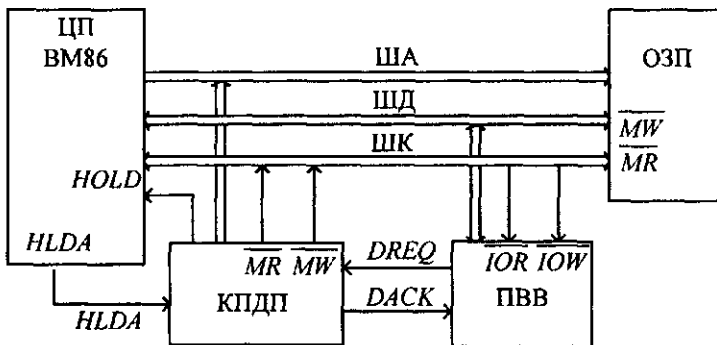


Рис. 11.25. Структурна схема МПС з контролером ПДП

Контролер ПДП за запитом бере на себе керування системною шиною і виконує цикли читання-записування до одержання нульового змісту його лічильника циклів.

11.4.4. Програмовний таймер

Програмовний таймер (ПТ) призначений для організації роботи МПС в режимі реального часу і дозволяє формувати сигнали з різними часовими інтервалами і частотними характеристиками.

В МПС широко використовують ПТ ВІ53 (аналог модуля 8253). Технічні параметри мікросхеми: схемотехнологія — n -МОП, число транзисторів — 1500; тактова частота — 2 МГц; напруга живлення — плюс 5 В і споживана потужність — 1 Вт.

Структура ПТ ВІ53 показана на рис. 11.26.

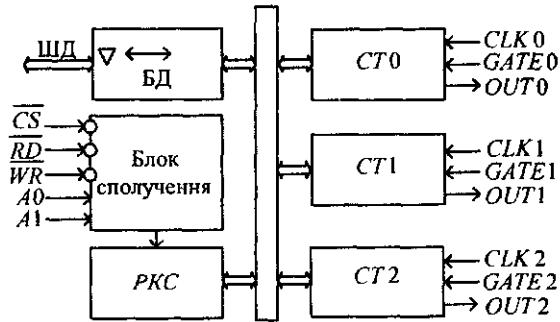


Рис. 11.26. Структура ПТ ВИ53

До складу ПТ ВИ53 входять: буфер БД для зв'язку з мікропроцесором; блок сполучення, який забезпечує виконання операцій введення-виведення інформації; регістр керуючого слова РКС; лічильники $CT0$ – $CT3$.

Призначення сигналів на входах і виходах ПТ: \overline{CS} — вибір мікросхеми; \overline{RD} , \overline{WR} — читання та записування; $A1$, $A0$ — адресні входи для вибору каналу ПТ чи РКС; $CLK0$ – $CLK1$ — входи синхроімпульсів, кожен з яких зменшує зміст відповідного лічильника на одиницю, $GATE0$ – $GATE2$ — входи керування лічильниками; $OUT0$ – $OUT2$ — вихідні сигнали лічильників.

Операції обміну інформацією між мікропроцесором і ПТ, які задають сигнали адреси і керування, наведені в табл. 11.4

Таблиця 11.4

Операція	Сигнали керування				
	\overline{CS}	\overline{RD}	\overline{WR}	$A1$	$A0$
ШД→РКС	0	1	0	1	1
ШД→ $CT0$	0	1	0	0	0
ШД→ $CT1$	0	1	0	0	1
ШД→ $CT2$	0	1	0	1	0
$CT0$ →ШД	0	0	1	0	0
$CT1$ →ШД	0	0	1	0	1
$CT2$ →ШД	0	0	1	1	0
Z-стан	1	X	X	X	Z

Режим роботи кожного каналу програмується записуванням керуючого слова в РКС і початкового значення змісту лічильників. Формат керуючого слова і призначення його окремих розрядів показані на рис. 11.27.



Рис. 11.27. Формат керуючого слова

Під час роботи ПТ на входи *GATE* поступають сигнали дозволу або заборони лічби. Рівень вихідного сигналу *OUT* змінюється залежно від записаних в лічильник даних, заданого режиму і сигналу на вході *GATE*. Кожний канал ПТ можна програмувати на роботу в одному із шести режимів:

- "0" — програмовна затримка (рис. 11.28, а);
- "1" — програмовний чекаючий мультивібратор (рис. 11.28, б);
- "2" — програмовний генератор тактових імпульсів (рис. 11.28, в);
- "3" — генератор прямокутних сигналів (рис. 11.28, г);
- "4" — програмовно-керований строб;
- "5" — апаратно-керований строб.

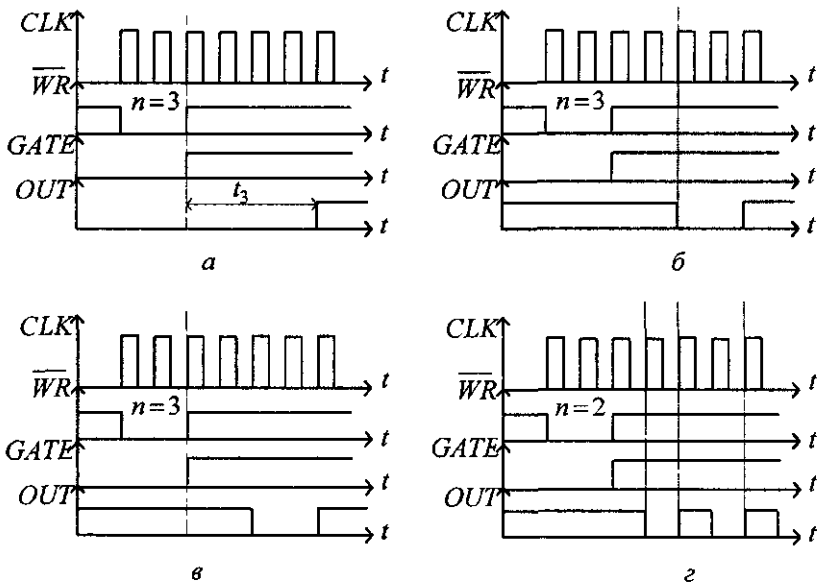


Рис. 11.28. Часові діаграми роботи ПТ:

а — режим "0"; б — режим "1"; в — режим "2"; г — режим "3"

Схема підключення ПТ до магістралі МПС показана на рис. 11.29.

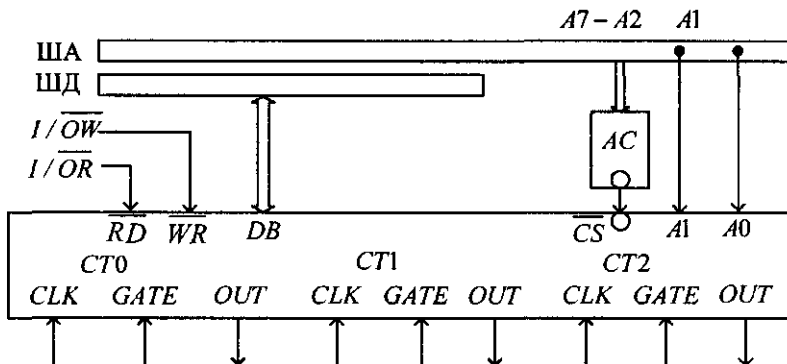


Рис. 11.29. Схема підключення ПТ до магістралі МПС

Тривалість керуючих сигналів на входах *GATE* має бути не менше 150 нс.

11.4.5. Програмовний паралельний інтерфейс.

Програмовний паралельний адаптер (ППА) типу ВВ55 призначений для організації паралельного обміну інформацією між ядром МПС і периферією (клавіатурою, індикаторами та ін.). Технічні характеристики: схемотехнологія — n -МОН, число транзисторів — 1600; напруга живлення — плюс 5 В і споживана потужність — 0,3 Вт. До складу ППА входять (рис. 11.30):

- двонаправлений буфер даних БД, який підключається до системної шини МПС;
- блок сполучення і керування;
- три регістри PA , PB і PC , які разом із серійними двонаправленими буферами створюють порти введення — виведення A , B і C . Порти зв'язані з периферією каналами KA , KB і KC ;
- регістр керуючого слова РКС.

Усі регістри, буфери і канали — восьмирозрядні. При цьому порт C розбивається на дві незалежні частини. Порт A і старші розряди порту C створюють групу A ; порт B і молодші розряди порту C створюють групу B .

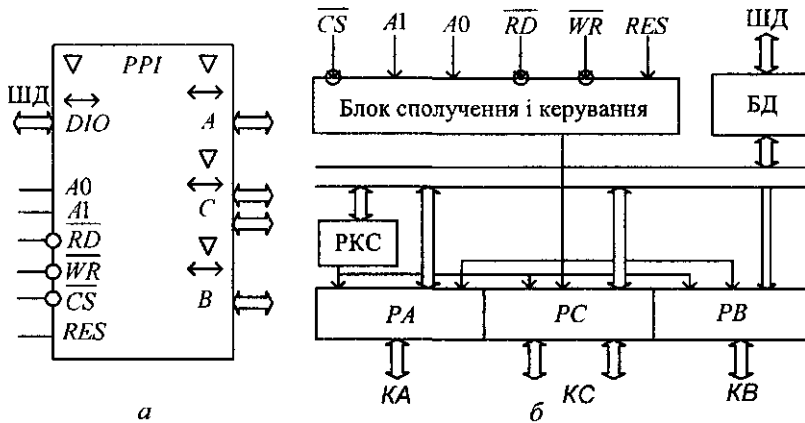


Рис. 11.30. Програмовний паралельний адаптер ВВ55:

а — умовне позначення; *б* — функціональна схема

На блок сполучення і керування поступають сигнали вибору мікросхеми \overline{CS} , читання \overline{RD} , записування \overline{WR} . Ці сигнали визначають напрямок передачі: введення (команда $IN\ ADR$) і виведення (команда $OUT\ ADR$). Значення розрядів адрес $A1$, $A0$ адресують конкретний порт в ППА: $A1 \cdot A0 = 00$ — A , 01 — B , 10 — C , 11 — РКС. Старші розряди адреси $A15$ – $A3$ поступають на адресний селектор, який формує сигнал вибору всієї мікросхеми \overline{CS} (табл. 11.5)

Таблиця 11.5

Операція	\overline{CS}	\overline{RD}	\overline{WR}	$A1$	$A0$	Напрямок обміну
IN (введення)	0	0	1	0	0	ШД ← A
	0	0	1	0	1	ШД ← B
	0	0	1	1	0	ШД ← C

Таблиця 11 5. Продовження

Операція	\overline{CS}	\overline{RD}	\overline{WR}	$A1$	$A0$	Напрямок обміну
OUT (виведення)	0	1	0	0	0	ШД→А
	0	1	0	0	1	ШД→В
	0	1	0	1	0	ШД→С
	0	1	0	1	1	ШД→РКС
Z-стан	1	x	x	x	x	

Група *A* може налаштуватися на один з трьох режимів обміну 0 — напівдуплексний синхронний, 1 — напівдуплексний асинхронний, 2 — дуплексний асинхронний
Група *B* може налаштуватися тільки на режим 0 або 1

Керуючі слова (рис 11 31) задають режими роботи портів (при $D7 = 1$) або побітно (при $D7 = 0$) установлюють розряди порту *C*, при цьому значення біта записують у розряд $D0$ слова керування

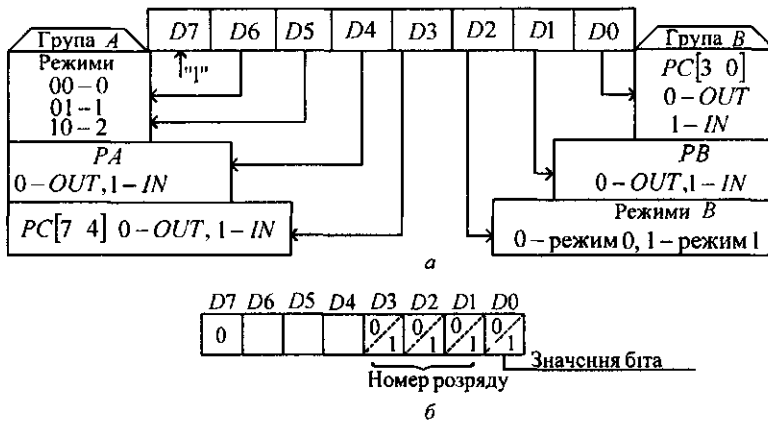


Рис. 11 31 Структура керуючого слова ППА

а — для задання режимів, б — для побітного установлення

Роздільне установлення використовують для послідовного обміну даними та організації обміну за перериванням В портах *A* і *B* є тригери дозволу переривання відповідно T_A і T_B Установлення в режим 1 цих тригерів дозволяє обмін за перериванням

В режимі 0 порти *A* і *B* та дві рівні частини порту *C* незалежно налаштовуються на введення або виведення інформації без супроводжуючих сигналів керування Дані, які виводяться з мікропроцесора, завжди попередньо завантажуються в акумулятор За командою *OUT* зміст акумулятора пересилається в адресований порт, де він запам'ятовується Дані, які вводяться в мікропроцесор за командою *IN*, поступають в акумулятор, після цього прочитаний порт обнуляється

Приклад 11.1

Налаштувати порт *A* на виведення, порти *B* і *C* — на введення даних в режимі 0
Адреса РКС ППА дорівнює 1103H Команди програмування
MVI A, 10001011B,
OUT 1103H

В режимі 1 порти *A* і *B* використовують для асинхронної передачі даних, а лінії порту *C* — для вироблення та сприймання керуючих сигналів. Функціональне призначення керуючих сигналів порту *C* при виведенні даних:

- \overline{STB} — вхідний сигнал стробування записування даних в порт;
- \overline{IBF} — вихідний сигнал повідомлення для периферії, що вхідний буфер заповнений;
- \overline{INT} — вихідний сигнал запиту на переривання.

Керуюче слово для введення в режим, конфігурація порту *C* і часові діаграми роботи показані на рис. 11.32.

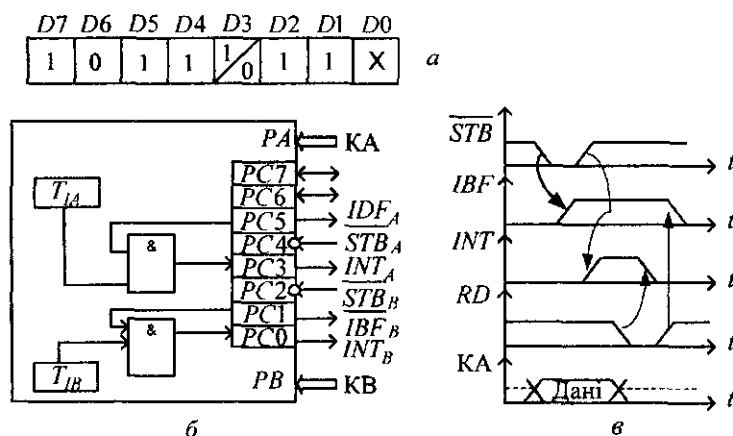


Рис. 11.32. Робота ППА при введенні в режимі 1:

а — керуюче слово; б — конфігурація порту *C*; в — часові діаграми

Тригери дозволу переривання в цьому режимі попередньо встановлюються в режим 1 за розрядами *PC4* — для порту *A* і *PC2* — для порту *B*. Вільні лінії *PC5* і *PC4* використовують для послідовного обміну даними.

Функціональне призначення керуючих сигналів порту *C* при виведенні даних в режимі 1:

- \overline{OBF} — сигнал для периферії повідомляє, що вихідний буфер повний;
- \overline{ACK} — вхідний сигнал від периферії, який підтверджує приймання даних;
- \overline{INT} — вихідний сигнал запиту переривання мікропроцесора.

Керуюче слово при виведенні в режимі 1, конфігурація порту *C* і часові діаграми роботи показані на рис. 11.33.

Тригери дозволу переривання при виведенні в режимі 1 попередньо встановлюються в положення 1 за розрядами *PC6* — для порту *A* і *PC2* — для порту *B*.

Режим 2 забезпечує дуплексний обмін між портом *A* і периферією та супроводжується п'ятьма керуючими сигналами *PC7–PC3*. Останні 11 інтерфейсних ліній можуть настроюватися на режими 0 або 1. Керуюче слово для режиму 2, конфігурація порту *C* і часові діаграми роботи ППА показані на рис. 11.34. Функції керуючих сигналів аналогічні режиму 1. Установлення в положення 1 тригерів дозволу переривання при введенні здійснюється за розрядами *PC4*, а при виведенні — *PC6*.

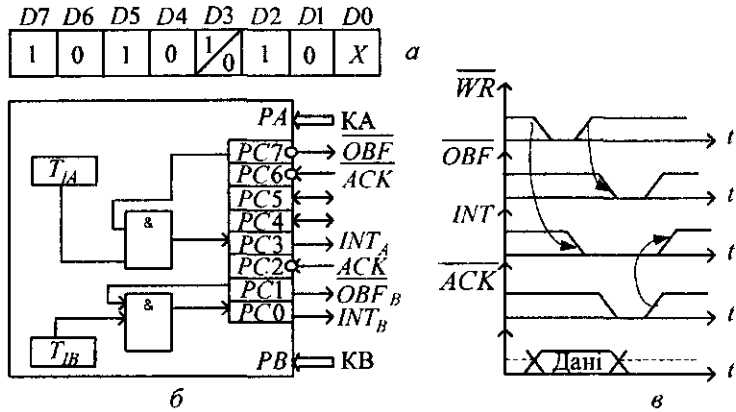


Рис. 11.33. Робота ППА при виведенні в режимі 1:
 а — керуюче слово; б — конфігурація порта С; в — часові діаграми

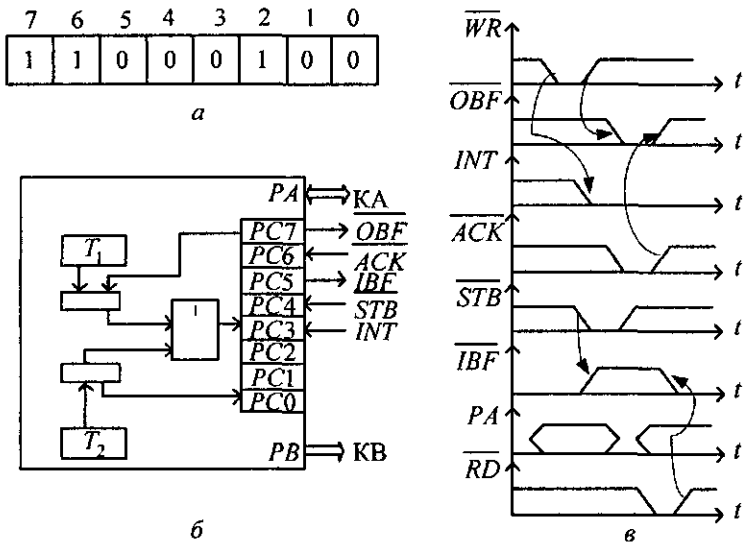


Рис. 11.34. Робота ППА в режимі 2:
 а — керуюче слово; б — конфігурація порта С; в — часові діаграми

Схема підключення ППА до СШ МПС, до восьмирозрядних індикаторів та клавіатури показана на рис. 11.35.

Приклад 11.2

Запрограмувати ППА для обслуговування клавіатури (введення даних) та індикаторів (виведення даних) в режимі 1. Програмування виконують такими командами:

```
MVI A, 10110100B;
OUT 1103H.
```

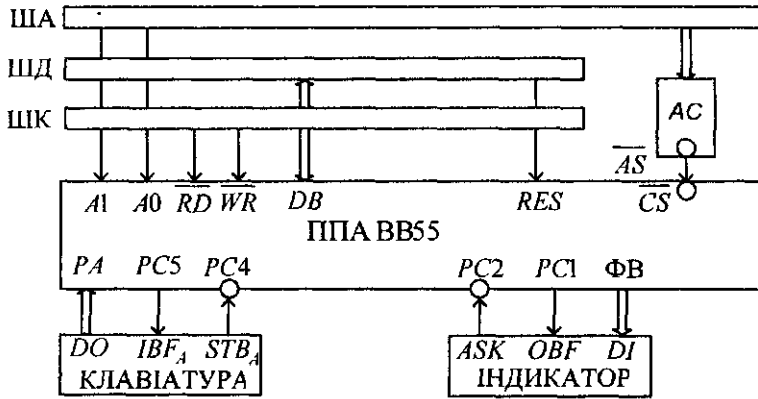


Рис. 11.35. Підключення ППА до системних шин, клавіатури та індикатора: АС — адресний селектор

11.4.6. Програмовний послідовний інтерфейс

В МПС для послідовного обміну інформацією між мікропроцесором і терміналом (телетайпом, дисплеєм та ін.) використовують ВІС, яку називають програмовним послідовним інтерфейсом (ППІ). Узагальнена структура МПС з ППІ показана на рис. 11.36.

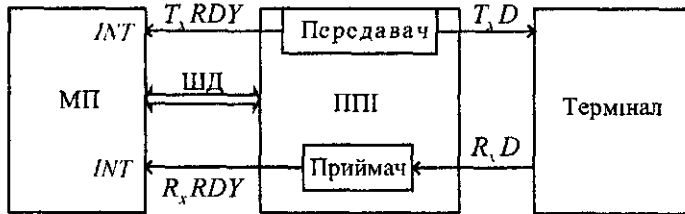


Рис. 11.36. Структура МПС з ППІ

Програмовний послідовний інтерфейс містить передавач (*transmitter*) і приймач (*receiver*). Передавач одержує від мікропроцесора по ШД дані (*data*) в паралельному коді і передає їх послідовно по лінії T_xD в термінал. Приймач одержує від терміналу по лінії R_xD дані в послідовному коді, перетворює їх у паралельні слова і пересилає по ШД в мікропроцесор. Система обміну може бути асинхронною чи синхронною, симплексною, напівдуплексною чи дуплексною.

Асинхронний режим використовують для послідовного обміну даними у вигляді одиничних символів (букв, цифр, знаків), які програмують на довжину 5–10 біт (рис. 11.37).

Передавач автоматично обрамляє кожний символ в асинхронному режимі додатковими службовими бітами: нульовим старт-бітом (початок передачі); інформаційними бітами, починаючи з молодшого розряду D_0 ; бітом контролю на парність чи непарність; одиничним стоп-бітом (кінець передачі) тривалістю 1, 1,5 або 2 біт. В такому ж форматі в асинхронному режимі дані поступають від терміналу, а приймач ППІ автоматично виключає службові біти. Часовий інтервал між символами при асинхронному обміні – фіксований.

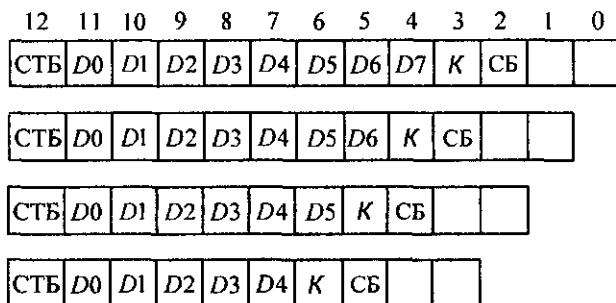


Рис. 11.37. Формат символів в асинхронному режимі:

СТБ — старт-біт; D7 (6,5,4),..., D0 — інформаційні біти; K — біт контролю; СБ — стоп-біт

При синхронному обміні стартові й стопові біти виключаються, а синхронізація здійснюється за допомогою одного або двох синхросигналів, які попередньо записуються в ППІ (рис. 11.38.)

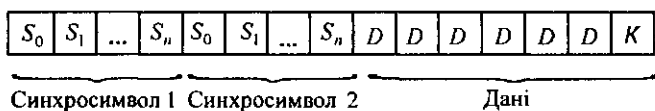


Рис. 11.38. Формат даних при синхронному обміні

При синхронному прийманні приймач виконує побітне порівняння вхідного потоку даних із записаними синхросимволами, доки не виявить їхнього проходження. Після цього він інтерпретує кожну групу довжиною від 5 до 8 біт (залежно від програмування) як символ. Символи ідуть один за одним безперервно. При неготовності даних передавач продовжує роботу, посылаючи для підтримки синхронізації спеціальні коди.

Синхронний режим обміну використовують для пересилок масивів даних; він підвищує швидкість обміну, яка вимірюється числом бітів, що пересилаються за секунду. Одиницю біт за секунду (біт/с) називають **бодом**.

В МПС широко застосовують мікросхему ВВ51 (аналог модуля 8251), яка є універсальним асинхронно-синхронним ППІ. Технічні характеристики мікросхеми: схемотехнологія — n -МОН, число транзисторів — 3000; тактова частота — 2 МГц; напруга живлення — плюс 5 В і споживана потужність — 0,4 Вт; швидкість обміну в асинхронному режимі — $0...64 \cdot 10^3$ бод, а асинхронному — $0...19,2 \cdot 10^3$ бод.

Мікросхема ВВ51 містить буфер БД, блок сполучення з керуючими сигналами, магістралі, передавач, приймач та блок зв'язку з терміналом (рис. 11.39).

Буфер БД забезпечує двонаправлений обмін паралельними восьмирозрядними даними між мікропроцесором та ППІ. На входи блока сполучень поступають сигнали, які ініціюють напрямки обміну між мікропроцесором і ППІ, та види слів: дані, програмування та стану ППІ (табл. 11.6).

Призначення вхідних сигналів блока сполучення: CLK — синхронізація; \overline{CS} — вибір мікросхеми; \overline{RD} — дозвіл читання інформації з ППІ на ШД; \overline{WR} — дозвіл на записування інформації із ШД в ППІ; C/\overline{D} (control data) — тип слів: $C/\overline{D} = 1$ — записування керуючого слова; $C/\overline{D} = 0$ — передача даних.

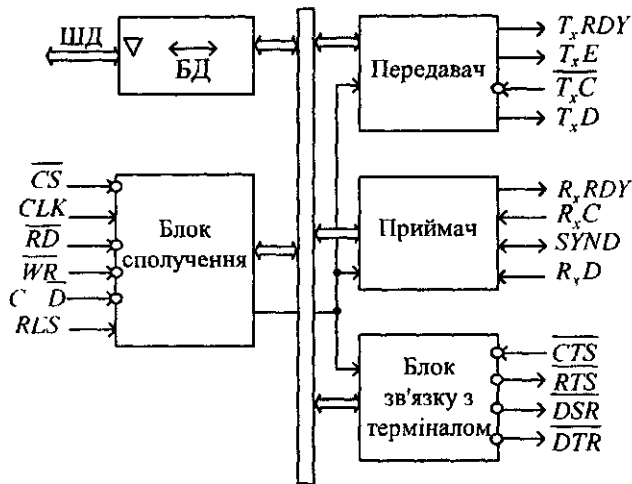


Рис. 11.39. Функціональна схема VIC BB51

Таблиця 11.6

Операція	\overline{CS}	C/\overline{D}	\overline{RD}	\overline{WR}
ППІ → ШД; читання даних	0	0	0	1
ШД → ППІ; записування даних	0	0	1	0
ШД → ППІ; записування керуючих слів	0	1	1	0
ППІ → ШД; читання слова стану	0	1	0	1
Z-стан	1	x	x	x

На входах і виходах передавача діють такі сигнали:

- T_xD — послідовна передача даних;
- $\overline{T_xC}$ — синхронізація передачі даних;
- T_xRDY — готовність передавача до приймання даних із ШД;
- T_xE — регістр даних передавача пустий;

На входах і виходах приймача діють такі сигнали:

- R_xD — послідовні дані з терміналу;
- $\overline{R_xC}$ — синхронізація приймання даних;
- $\overline{R_xRDY}$ — дані готові до записування в мікропроцесор;
- \overline{SYND} — вхід/вихід; його призначення задають програмно в синхронному режимі;

На входах і виходах блока зв'язку з терміналом сигнали мають таке призначення:

- \overline{CTS} — готовність приймача терміналу;
- \overline{RTS} — готовність ППІ передавати дані;
- \overline{DSR} — готовність передавача терміналу;
- \overline{DTR} — готовність ППІ прийняти дані.

В асинхронному режимі швидкість обміну кратна частоті сигналів синхронізації по входу $\overline{T_xC}$ для передачі і $\overline{R_xC}$ — для приймання. Коефіцієнт кратності встанов-

люється програмно і дорівнює 1 16 та 64. Наприклад, для швидкості обміну 100 бод частота сигналів синхронізації на входах T_xC або R_xC має бути 100 Гц при кратності 1 1,6 КГц — при кратності 16 та 6,4 КГц — при кратності 64

У режимі синхронного обміну з внутрішньою синхронізацією швидкість обміну дорівнює частоті сигналів на вході T_xC для передавача і R_xC — для приймача

Режим роботи і функції ВІС ВВ51 задає програма за допомогою слів ініціалізації та слів команд. Слово ініціалізації визначає асинхронний чи синхронний режим роботи, кратність коефіцієнта ділення частоти, довжину даних в бітах, наявність контролю на парність чи непарність, тривалість стоп-біта, кількість синхросимволів та вид синхронізації — зовнішню чи внутрішню. Формат слова ініціалізації ППІ для асинхронного режиму і призначення розрядів показані на рис 11 40

D7	D6	D5	D4	D3	D2	D1	D0
Стоп-біти		Контроль		Довжина,біт		Кратність	
01-1		01-за парністю		00-5		01-1	
10-1 5		11-за непарністю		01-6		10-16	
11- 2				10-7		11-64	
				11-8			

Рис 11 40 Формат слова ініціалізації для асинхронного режиму

У форматі слова ініціалізації для синхронного режиму ($D1 D0 = 00$) розряд $D6$ визначає спосіб синхронізації: нуль — зовнішня, одиниця — внутрішня, $D7$ — задає кількість синхросимволів: нуль — 1, одиниця — 2

За допомогою слова команди (рис 11 41) робота ППІ програмується на виконання таких функцій (при "1" в розрядах) $D0$ — передача, $D1$ — запит на передачу, $D2$ — приймання, $D3$ — кінець передачі, $D4$ — скидання прапорців помилок, $D5$ — запит на приймання, $D6$ — програмне скидання ППІ, $D7$ — пошук синхросимволів у синхронному режимі

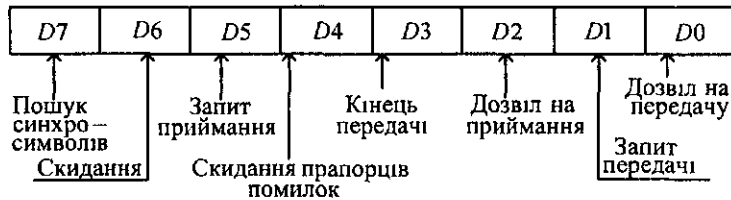


Рис 11 41 Формат слів команд

Керуючі слова мають формуватися після сигналу "RES" у такій послідовності: слово ініціалізації (режиму), синхросимвол "1", синхросимвол "2" в синхронних режимах, слова команди, дані, слова команди, дані і т д

Для контролю стану ППІ в процесі обміну даними мікропроцесор командою введення зчитує слово стану із спеціального реєстра стану. Формат слова стану і призначення розрядів показані на рис 11 42

У розрядах слова стану формується інформація (при "1" в розрядах) $D0$ — дозвіл для передавача, $D1$ — дозвіл для приймача, $D2$ — реєстр передавача пустий, $D3$ — помилка парності PE, $D4$ — помилка переповнення (мікропроцесор не прочи-

тав символа); $D5$ — не виявлений стоп-біт (в асинхронному режимі); $D6$ — відсутність синхронізації; $D7$ — готовність терміналу до видачі даних.

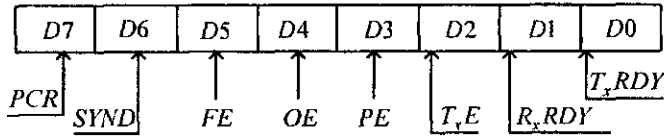


Рис. 11.42. Формат слова стану ППІ

При синхронній передачі дані поступають на вихід сигналу T_xD . Якщо наступні дані не поступили в ППІ від мікропроцесора, то вихідний сигнал $T_xE = 1$. Після закінчення передачі символу вихідний сигнал $T_xRDY = 1$. Записування в ППІ наступного символу від мікропроцесора може здійснюватися двома способами: перший — читання слова стану; другий — використання сигналу T_xRDY як запит на переривання.

Асинхронне приймання даних починається з пошуку стоп-біта, який встановлюється на виході сигналу R_xD низьким рівнем напруги (лог. 0). Наявність цього рівня повторно перевіряється стробуванням його середини внутрішнім імпульсом. Якщо старт-біт знайдений, то запускається внутрішній лічильник, який визначає початок і кінець бітів даних, біт контролю і стоп-біт.

Після цього на виході готовності приймання R_xRDY встановлюється лог. 1. Передача прийнятого символу в мікропроцесор може здійснюватись двома способами: перший — читання слова стану; другий — використання сигналу R_xRDY як запиту на переривання.

Під час синхронної передачі на виході T_xD з частотою синхронізації на вході T_xC формується послідовність бітів, яка починається із синхросимволів. Потім передаються дані, які попередньо поступають в ППІ з мікропроцесора. Якщо черговий символ не завантажений до моменту передачі, то ППІ вставляє в послідовність синхросимволи. Кінець передачі визначається записуванням у слові команди $D3 = 1$.

Під час синхронного приймання з внутрішньою синхронізацією ППІ починає роботу з пошуку у вхідній послідовності синхросимволів і порівнює символи, що приймаються, із записаними синхросимволами під час програмування. Після виявлення синхросимволів встановлюється значення $SYND = 1$ і починається приймання даних; цей сигнал автоматично скидається при читанні слова стану.

При синхронному прийманні із зовнішньою синхронізацією на вхід $SYND$ поступає сигнал (строб) від зовнішнього пристрою, який дозволяє приймання даних на вході R_xD із швидкістю синхроімпульсів на вході R_xC . У цьому режимі пошук синхросимволів не відбувається, тому немає потреби подавати їх на початку кодової послідовності.

Контрольні запитання

1. Що таке мікропроцесорна система?
2. Назвіть основні складові частини мікропроцесорної системи
3. Поясніть структуру магистральної МПС
4. Назвіть основні складові частини програмного забезпечення МПС
5. Охарактеризуйте поняття архітектури МПС
6. Що таке інтерфейс?
7. Які функції в МПС має забезпечувати інтерфейс?
8. Поясніть класифікацію інтерфейсів
9. Які технічні характеристики уніфікуються в паралельних системних стандартних інтерфейсах?
10. Охарактеризуйте інтерфейсні мікросхеми генератор тактових імпульсів, буферні регістри, двонаправлені шинні формувачі
11. Поясніть призначення контролера системної шини
12. Поясніть функції арбітра шин ВВ89
13. Що таке системна і резидентна шини?
14. Дайте характеристику програмних інтерфейсних контролерів
15. Назвіть режими роботи контролера переривань ВН59
16. Поясніть функції КПДП ВТ57

Висновки

Успішне вирішення задач інформатики неможливе без використання методів і засобів комп'ютерної схемотехніки, яка вивчає теоретичні методи проектування та способи технічної реалізації різноманітних апаратних засобів комп'ютерів, мікропроцесорних систем і обчислювальних мереж на сучасній елементній базі

Теоретичною основою комп'ютерної схемотехніки є теорія інформації, двійкова арифметика та алгебра логіки. Теорія інформації й обчислювальна техніка складають ядро **інформатики** — фундаментальної дисципліни, яка вивчає процеси передачі, оброблення, зберігання і використання наукової інформації

Дуже важливим параметром мікросхем є рівень інтеграції, який визначається числом транзисторів розміщених на кристалі (чипі). За цією ознакою мікросхеми пройшли з часу створення в 1959 р. шлях від малих (до 100 транзисторів на кристалі) до ультра-BIC (більше одного мільйона транзисторів на кристалі). Згідно з законом Мура рівень інтеграції зростає вдвічі за кожні 2–3 роки. За прогнозами експертів у 2012 р. на чипі буде розміщено 1,4 млрд транзисторів. На ультра-BIC побудовані суперскалярні мікропроцесори з *CISC*- та *RISC*-архітектурами.

Елементна база представлена потенціальною, імпульсною та потенціально-імпульсною системами елементів. Головна увага приділена потенціальним елементам, властивості яких (відсутність конденсаторів, індуктивності) найбільше відповідають вимогам сучасної інтегральної технології. Потенціальні логічні елементи реалізують операції ЧИ, І, НЕ-ЧИ, НЕ-І, НЕ-І-ЧИ та ін.

Для побудови мікросхем з великим рівнем інтеграції широко застосовуються структури КМОН з дуже малим енергоспоживанням.

З ростом рівня інтеграції зменшуються розміри транзисторів, що призводить до зміни їхніх фізичних характеристик. Тому ведуться інтенсивні пошуки нових фізичних принципів побудови двійкових елементів — оптоелектронних, акустоелектронних, молекулярних, біологічних та ін.

Основою побудови цифрових автоматів з пам'яттю є тригери — елементи з двома стійкими станами, кожен з яких відображає значення логічного нуля та логічної одиниці. В посібнику розглянуті одно- і двоступеневі асинхронні та синхронні *RS*-, *JK*-, *T*- і *D*-тригери.

Типові функціональні вузли розподіляються на накопичувальні (автомати з пам'яттю) та комбінаційні (автомати без пам'яті). В накопичувальних вузлах (реєстри, лічильники, розподільвачі тактів) значення вихідних сигналів залежить як від вхідних сигналів, так і стану пам'яті. В комбінаційних вузлах вихідні сигнали визначаються тільки комбінаціями вхідних сигналів у даний момент часу. Власне різні способи оброблення інформації відбуваються у типових комбінаційних вузлах, до яких відносяться дешифратори, шифратори, мультиплексори, демультіплексори, компаратори, схеми контролю за модулем два, кодоперетворювачі, двійкові та двійково-десяткові суматори.

В керуючих мікропроцесорних системах широко застосовується фізичне перетворення сигналів з аналогової форми у цифрову і, навпаки, без зміни суті інформації. Ці функції виконують АЦП і ЦАП, загальна характеристика яких, способи побудови і параметри серійних виробів докладно розглянуті у посібнику.

Мікросхеми пам'яті розвиваються дуже інтенсивно, при цьому збільшується їхня ємність і швидкодія, зменшується споживана потужність. На даний час є мікросхеми пам'яті з ємністю до 512 Мбіт. За прогнозами експертів в 2012 р. мікросхеми пам'яті будуть зберігати до 275 млрд. біт інформації.

Виконання операцій реалізується під дією послідовності керуючих сигналів, які вироблюються пристроями керування. Розглянуті два основні типи пристроїв керування — зі схемною та програмованою логікою. На основі мікропрограм основних арифметичних операцій як приклад синтезовано конкретні схеми мікропрограмних автоматів.

Розробка нової ВІС або НВІС коштує десятки мільйонів доларів. Але з ростом складності мікросхем зменшилась їхня універсальність і відповідно ринок збуту. Кризова ситуація в мікроелектроніці була подолана у 1971 р., коли фірма *Intel* випустила перший мікропроцесор. Після цього з'явилася можливість використання однієї і тієї ж схеми для розв'язання різних задач створенням відповідної програми і розміщенням її в напівпровідниковій пам'яті мікрокомп'ютера. В результаті розвиток комп'ютерної техніки та інформатики вийшов на якісно новий етап.

Сучасні МПК мають у своєму складі велике число допоміжних мікросхем, за допомогою яких з'єднуються модулі мікропроцесорної системи. До них відносяться інтерфейсні схеми, буферні регістри, шинні двонаправлені формувачі, генератори тактових імпульсів, арбітри шин і системні контролери. Розглянуті також програмовані інтерфейсні контролери послідовного та паралельного обміну інформацією, переривань та прямого доступу до пам'яті і таймери.

Матеріал, поданий у навчальному посібнику є результатом викладання авторами навчальних дисциплін "Елементи і вузли ЕОЦМ", "Схемотехніка ЕОМ", "Комп'ютерна схемотехніка", "Мікропроцесори і мікро-ЕОМ", "Обчислювальні комплекси, системи та мережі" на кафедрах "Електронні цифрові обчислювальні машини", "Обчислювальні машини, системи та мережі", "Обчислювальна техніка".

Словник загальних термінів

Автомат без пам'яті	Автомат з одним внутрішнім станом
Автомат Мілі	Автомат, в якому вихідні сигнали є функцією вхідних сигналів і стану пам'яті
Автомат Мура	Автомат, в якому вихідні сигнали визначаються тільки станом пам'яті
Агрегативання	Розділення мікропроцесорної системи на прості функціонально і конструктивно закінчені блоки — модулі
Адаптер	Засіб сполучення деякого зовнішнього пристрою з визначеною шиною мікропроцесорної системи
Активний вихід (дешифратора)	Вихід, на якому з'являється керуючий сигнал
Аналого-цифровий перетворювач	Пристрій, призначений для перетворення аналогової інформації (звичайно у вигляді напруги) у цифровий код

Ансамбль	Множина з відомим розподілом елементів
Арифметико-логічний пристрій	Пристрій, призначений для виконання арифметичних і логічних операцій, передбачених системою команд комп'ютера
Архітектура комп'ютера	Найбільш загальні принципи побудови комп'ютера, що охоплюють коло питань, суттєвих в першу чергу для користувача: функціональний склад технічних та програмних засобів та їхню взаємодію в процесі обробки інформації; систему команд, їхні формати і способи кодування; методи адресації команд і даних
Асемблер	Машинно-орієнтована мова програмування
Асинхронний суматор	Суматор, в якому інтервал часу для додавання визначається моментом фактичного закінчення операції
Асинхронний тригер	Тригер без тактового входу
Байт	8 біт
Біт	Найменша кількість інформації, двійкова цифра
Бод	Одиниця біт за секунду (біт/с)
Булева алгебра	Наука, яка використовує математичні методи для розв'язання логічних задач
Булева змінна	Змінна із двома значеннями (станами)
Введення-виведення	Обмін інформацією між ядром машини і периферійними пристроями
Великий комп'ютер	Машина загального призначення з продуктивністю до 400 тис. оп./с, ємністю пам'яті — декілька мегабайт, розрядністю — 32 або 64 біт
Вентиль	Схема, яка здійснює елементарну логічну операцію
Внутрішня пам'ять	Оперативна і постійна пам'ять, а також реєстри арифметико-логічного пристрою комп'ютера
Гігабайт	1024 мегабайт
Глобальна таблиця дескрипторів	Таблиця дескрипторів, яка використовується спільно всіма задачами, які виконуються в даному комп'ютері
Дані	Інформація, підготовлена для опрацювання на комп'ютерах
Двійковий елемент	Елемент з двома станами
Демультіплексор	Функціональний вузол комп'ютера, призначений для комутації (перемикання) сигналу з одного інформаційного входу на один з інформаційних виходів
Демультіплексування шин	Почергове перемикання груп ліній від одного джерела інформації до багатьох приймачів

Дескриптор	Восьмибайтова структура даних, які використовуються для визначення властивостей сегментів: базова адреса сегмента і його розмір, його призначення (тип), характеристики захисту і додаткова інформація
Дешифратор	Функціональний вузол комп'ютера, призначений для перетворення кожної комбінації вхідного двійкового коду в керуючий сигнал лише на одному із своїх виходів
Диз'юнкція	Булева операція ЧИ, результатом якої є значення нуль тоді і тільки тоді, коли обидва операнди мають значення нуль
Динамічна пам'ять	Пам'ять, в якій інформація зберігається у вигляді зарядів на дуже малій ємності, створеній між стоком і підкладкою МОН-транзистора
Дискретизація сигналів у часі	Принцип подачі інформації на входи елементів і вузлів у тактові моменти
Дискретизація у часі	Перетворення функції $y(t)$ неперервного аргументу t в функцію $y(t_i)$ дискретного аргументу t_i
Діапазон представлення	Інтервал числової осі, замкнутий симетрично між максимальним і мінімальним значенням числа
Діт	3,32 біта
Еквівалентність	Булева операція, результатом якої є одиниця тоді і тільки тоді, коли операнди набувають однакових значень
Експлуатаційна продуктивність комп'ютера	Оцінюється даними про реальне робоче навантаження в основних областях застосування; при цьому враховують необхідну площу розміщення машини, механічні та кліматичні умови експлуатації, споживану потужність та ін.
Електронно-обчислювальна машина	Комплекс електронного устаткування, яке призначене для автоматизації процесів обробки дискретної інформації і має спільне керування
Елемент	Найменша неподільна мікроелектронна схема (вироб), призначена для виконання логічних операцій або зберігання біта інформації
Елементарна булева функція	Булева функція одного або двох аргументів
Елементарний регістр	Регістр, призначений тільки для приймання (записування), зберігання і передачі інформації
Ентропія	Кількість інформації, що припадає на один елемент повідомлення

Заборона	Булева операція, результатом якої є значення одиниця тоді і тільки тоді, коли значення одного операнда дорівнює одиниці, а іншого — нулю
Закодований граф	Граф мікропрограми, в якому мікрооперації подані ідентифікаторами керуючих сигналів, а логічні умови — булевими аргументами
Заперечення	Булева операція НЕ, результатом якої є значення, протилежне значенню операнда
Затвор	Металевий електрод МОН-транзистора, на який надходить керуюча напруга
Захищений режим	Забезпечує захист апаратних і програмних ресурсів однієї задачі від можливого впливу іншої
Змістовний граф	Граф мікропрограми, в якому операторні й умовні вершини містять повний опис мікрооперацій і логічних умов
Зовнішня пам'ять комп'ютера	Призначена для тривалого і енергонезалежного зберігання великих об'ємів інформації
Зсув	Однотактне просторове переміщення двійкового слова в розрядній сітці із збереженням порядку слідування нулів і одиниць
Зсувовий регістр	Регістр, в якому зберігання даних поєднується з мікроопераціями зсуву
Імплікація	Булева операція, результатом якої є значення нуль тоді і тільки тоді, коли значення одного з операндів дорівнює нулю, а іншого — одиниці
Імпульсний сигнал	Сигнал, що наростає в тактовий момент, а спадає в границях даного такту
Інверсія	Див. "Заперечення"
Ініціальний автомат	Автомат, який має початковий стан
Інтегральна мікросхема	Мікроелектронний виріб з високою щільністю упакування електрорадіоелементів (резисторів, діодів, транзисторів) і з'єднань між ними
Інтерфейс	Сукупності шин, сигналів, допоміжних мікросхем та алгоритмів, призначених для обміну інформацією між пристроями комп'ютера
Інформативні параметри	Змінювані параметри фізичної величини за законом переданого повідомлення
Інформатика	Фундаментальна дисципліна, яка вивчає процеси передачі, оброблення, зберігання і використання наукової інформації
Інформаційна ємність	Максимальний об'єм даних, який може одночасно зберігатися в пам'яті

Канал введення-виведення	Спеціальний пристрій універсального комп'ютера, який забезпечує обмін інформацією між оперативною пам'яттю і периферійним пристроєм
Канал (МОН-транзистора)	Триповерхневий провідний прошарок між витоком і стоком МОН-транзистора, у якому значення струму визначається за допомогою електричного поля
Канал передачі (зв'язку)	Сукупність пристроїв, що мають один вхід і один вихід, та призначені для передачі інформації на відстані
Квантування за рівнем	Перетворення неперервної функції $y(t)$ в дискретну множину значень y_i
Кеш-пам'ять	Служить для зберігання копій інформації, що використовуються в поточних операціях обміну
Кілобайт	1024 байт
Кластерна система	Об'єднання машин з єдиним механізмом керування і програмного забезпечення
Код Грея	Код, утворений послідовністю двійкових чисел, в яких два будь-яких сусідніх числа відрізняються тільки одним розрядом
Код числа	Запис числа у деякій системі числення
Командний цикл	Час виконання мікропрограми команди
Комбінаційна схема	Схема, в якій логічний стан виходів елементів залежить тільки від комбінації вхідних сигналів у даний момент часу
Комірка пам'яті	Сукупність однобітних елементів пам'яті, які утворюють n -розрядне слово
Компаратор	Функціональний вузол комп'ютера, призначений для вироблення ознак відношень між двійковими словами
Компіляція програми	Трансляція програми на мову, близьку до машинної
Комп'ютер	Програмно керована фізична система, призначена для алгоритмічного опрацювання інформації, поданої сигналами
Комп'ютерна техніка	Сукупність засобів для автоматизації процесів обробки інформації, а також галузь техніки, яка займається розробленням, виготовленням та експлуатацією комп'ютерів
Компонент	Електрорадіоелемент інтегральної мікросхеми
Контролер	Спеціалізований процесор, призначений для керування зовнішніми пристроями комп'ютера
Контроль парності	Спосіб контролю, заснований на допущенні, що в двійковому числі найчастіше виникають одиничні помилки — втрата або поява зайвої одиниці

Кон'юнкція	Булева операція I, результатом якої є значення одиниця тоді і тільки тоді, коли значення кожного операнда дорівнює одиниці
Кріотрон	Найпростіший кріоелектронний логічний елемент з двома входами та одним виходом
Лічильник	Типовий функціональний вузол комп'ютера, призначений для лічби вхідних імпульсів
Логічна схема	Сукупність взаємозалежних логічних елементів з формальними методами опису
Логічний елемент	Див. "Вентиль"
Локальна таблиця дескрипторів	Таблиця дескрипторів, яка визначає сегменти пам'яті, доступні тільки конкретній задачі
Локальна шина	Сукупність ліній, які зв'язують мікропроцесор з буферними регістрами адреси, формувачами даних та контролерами
Магістраль	Сукупність сигналів з виходів буферних регістрів адреси, формувачів даних та контролерів
Макстерм	Функція n змінних, яка дорівнює нулю тільки на одному наборі
Маркуючий код	Одиниця, по розміщенню якої можна розрізнити стан лічильника при одиничному кодуванні
Машинний такт	Часовий інтервал між двома сусідніми тактовими імпульсами
Машинні зображення	Числа, представлені в прямому, оберненому і доповняльному кодах
Мегабайт	1024 кілобайт
Мегафлопс	Мільйон операцій за секунду над операндами з плаваючою комою
Мейн фрейм	Див. "Великий комп'ютер"
Мікрокоманда	Сукупність мікрооперацій, які виконуються в одному такті
Мікрооперація	Елементарна дія, яка виконується в комп'ютерах в одному машинному такті
Мікропрограмний автомат	Керуючий автомат, функціонування якого задають за допомогою мікропрограм
Мікропроцесор	Процесор, складові частини якого мініатюризовані та розміщені в одній або декількох мікросхемах
Мікропроцесорний комплект	Сукупність мікропроцесорних та інших мікросхем, які сумісні за конструктивно-технологічним виконанням і призначені для спільного використання
Мікропроцесорна система	Сукупність обчислювальних засобів, куди входять один або декілька мікропроцесорів та напівпровідникова пам'ять і засоби інтерфейса

Міні-комп'ютер	Комп'ютер, побудований на мікропроцесорах, складові частини яких розміщені в одній або декількох інтегральних схемах
Мінімально повний базис	Система функцій, яка перетворюється в неповну в разі видалення з неї будь-якої функції
Мінтерм	Функція n змінних, яка дорівнює одиниці тільки на одному наборі
Мінус-двійкова система числення	Позиційна двійкова система числення з основою $q = -2$ і символами 0, 1
Мова асемблеру	Див. "Асемблер"
Мова мікропрограмування	Мова, яка призначена для описування мікропрограм в термінах операторів
Мова програмування	Формалізована мова представлення програм
Модуляція	Зміна параметрів фізичної величини за законом переданого повідомлення
Модульність	Див. "Агрегатування"
Мультиплексор	Функціональний вузол комп'ютера, призначений для почергової комутації (перемикання) інформації від одного з входів на загальний вихід
Мультиплексування шин	Почергове перемикання шин (груп ліній) від кількох джерел інформації до одного приймача
Напівсуматор	Логічна схема, яка виконує додавання значень розрядів двійкових чисел та реалізує на виході значення результату і перенесення в старший сусідній розряд, не враховуючи перенесення з сусіднього молодшого розряду
Непозиційна система числення	Система числення, в якій значення кожної цифри не залежить від її позиції
Номінальна продуктивність комп'ютера	Середнє число суміші команд із врахуванням їхньої частоти повторення, які виконує ядро комп'ютера у вибраному класі задач
Носій інформації	Будь-яке запам'ятовуюче предметне середовище, призначене для записування і зберігання інформації з метою її безпосереднього введення в комп'ютер
Об'єктна програма	Програма в двійкових машинних кодах
Однорідна позиційна система числення	Позиційна система числення з однаковою основою в кожному розряді
Однорозрядний суматор	Логічна схема, яка виконує додавання значень відповідних розрядів двійкових чисел з урахуванням перенесення з молодшого сусіднього розряду та виробляє на виходах функції результат і перенесення в старший сусідній розряд

Операнд	Число, яке бере участь в операціях
Оперативна пам'ять	Призначена для тимчасового зберігання програм і даних, в ній виконуються операції записування і читання інформації
Операційні ресурси	Множина всіх операцій, що реалізуються в комп'ютері
Операція	Чітко визначена дія над одним або декількома операндами, яка створює новий об'єкт (результат)
Основа позиційної системи числення	Кількість графічних знаків, які використовуються для запису чисел у позиційній системі числення
Пакет прикладних програм	Функціонально завершений комплекс програмних засобів, орієнтований на розв'язання визначеного логічно цілісного класу задач
Пам'ять комп'ютера	Сукупність різних пристроїв, призначених для приймання, зберігання і видачі двійкової інформації
Паралельне передавання коду	Одночасне передавання в такті значення всього слова за допомогою багатоканальної передачі
Перемикальна змінна	Змінна із скінченним числом значень (станів)
Перемикання елемента	Перехід елемента з одного стану в інший
Переривання	Особлива операція, яка призупиняє роботу програми для виконання спеціальних системних дій
Перетворювач коду	Функціональний вузол комп'ютера, призначений для перетворення двійкового коду з однієї форми в іншу
Питома ємність	Відношенням інформаційної ємності запам'ятовуючого пристрою до його фізичного об'єму
Підканал	Засоби каналу введення-виведення, які призначені для обслуговування одного периферійного пристрою
Підкладка	Основна пластина напівпровідника в МОН-транзисторі
Пікова продуктивність комп'ютера	Середнє число коротких операцій типу "регістр-регістр" за секунду без операцій обміну з оперативною пам'яттю
Повідомлення	Упорядкована послідовність символів (букв, цифр, математичних знаків, призначених для передачі інформації), закодована в матеріальній формі
Повний дешифратор	Дешифратор з максимально можливим числом виходів
Позиційна система числення	Система числення, у якій для запису чисел використовують певну кількість графічних знаків (цифр і букв), які відрізняються один від одного

Показчик старшої одиниці	Пріоритетний шифратор, який призначений для пошуку старшої (лівої) одиниці в слові та формування на виході двійкового номера шуканого розряду
Покоління комп'ютерів	Категорія в історичній класифікації комп'ютерів, що базується переважно на технології виробництва їхньої елементної бази
Порогова напруга	Напруга, при якій утворюється канал МОН-транзистора
Послідовне передавання коду	Передавання в такті значення одного розряду слова за допомогою одного каналу передачі
Порт введення-виведення	Регістр на вході або виході зовнішнього пристрою для обміну даними з процесором
Послідовнісна схема	Схема, в якій логічне значення виходів визначають як комбінацією вихідних сигналів, так і станом пам'яті схеми в даний момент часу
Постійна пам'ять	Призначена для збереження програм, констант, табличних функцій та іншої інформації, яка записується заздалегідь і не змінюється в процесі поточної роботи комп'ютера
Потенціальний сигнал	Сигнал, який змінюється тільки в тактові моменти часу
Початкова програма	Програма на мові асемблеру
Прапори	Ознаки результатів операції, які формуються автоматично після виконання кожної команди
Представлення інформації	Спосіб формалізованого описування різноманітних сигналів і повідомлень
Пристрій керування	Функціональна частина комп'ютера, яка призначена для автоматичного керування обчислювальним процесом за допомогою послідовності керуючих і синхронізуючих сигналів
Пріоритетний шифратор	Шифратор, який при одночасному натисканні декількох клавіш виробляє код тільки старшої цифри
Програма	Мовна конструкція, яка є впорядкованою послідовністю описів і команд, призначених для обробки інформації
Процесор	Основна функціональна частина комп'ютера, яка інтерпретує й виконує команди, тобто безпосередньо реалізує програмно-керований процес обробки даних
Регістр	Типовий функціональний вузол комп'ютера, призначений для приймання, тимчасового зберігання, перетворення і видачі n -розрядного двійкового слова

Розкладання Шенона	Формула, яка дозволяє переходити до представлення функції n змінних через функції від $(n - 1)$ змінних
Розряд	Окрема позиція в записі числа
Розрядна сітка комп'ютера	Сукупність запам'ятовуючих елементів для розміщення одного двійкового числа
Розрядність	Число розрядів у записі числа
Сегмент	Область пам'яті з множиною комірок із суміжними адресами
Сегментація пам'яті	Механізм, за допомогою якого пам'ять розділяється на окремі області адресного простору визначеного призначення
Сервер	Комп'ютер, який виконує функції з обслуговування групи користувачів (розподіл ресурсів пам'яті, принтерів, баз даних)
Серія інтегральних мікросхем	Набір цифрових мікросхем із спільними конструкційно-технологічними і схемотехнічними ознаками
Сигнальна лінія	Провідник, який фізично з'єднує джерело і приймач інформації
Символ	Елементарна одиниця повідомлення
Симетрична знакорозрядна система числення	Надлишкова система числення з основою q і кількістю символів більше q
Синхронний суматор	Сумматор з постійним інтервалом часу для додавання
Синхронний тригер	Тригер з тактовим входом
Система числення	Сукупність цифр і правил для записування чисел
Системна продуктивність комп'ютера	Вимірюється за допомогою типових оціночних програм (бенчмарків), реалізованих на мовах високого рівня
Системна шина	Див. "Магістраль"
Скалярний процесор	Процесор з одним конвеєром — може виконувати лише одну команду за один машинний такт
Скінчений автомат	Автомат, в якому перехід з одного стану в будь-який інший завершується за скінченне число тактів
Слово	Група символів
Статичний запам'ятовуючий пристрій	Пристрій, в якому функцію запам'ятовування біта інформації виконують тригери
Стрілка Пірса	Булева операція, результатом якої є значення одиниця тоді і тільки тоді, коли обидва операнди дорівнюють нулю

Сума за модулем два	Двомісна булева операція, результатом якої є значення одиниця тоді і тільки тоді, коли операнди мають різні значення
Суматор	Функційний вузол комп'ютера, що виконує операцію додавання двох чисел
Суперкомп'ютер	Найпотужніша обчислювальна система, яка діє у відповідний історичний період
Суперскалярний процесор	Процесор, який має два і більше конвексів — може виконувати декілька команд за один машинний такт
Схема порівняння	Див. "Компаратор"
Таблиця дескрипторів	Визначена в адресному просторі пам'яті множина дескрипторів
Таблиця переривань	Містить дескриптори спеціальних об'єктів, які визначають точки входження процедур оброблення переривань і особливих випадків
Тактовий момент	Початок кожного тактового імпульсу
Тетрада	Чотири двійкових розряди
Трансляція програми	Перетворення програми з однієї мови програмування в рівноцінну на іншій мові
Трансфлюксор	Феритовий диск з двома неоднаковими отворами, що може знаходитися у двох станах, які характеризуються різним розподілом магнітного поля навколо великого отвору і відображають лог. 1 і лог. 0
Тригер	Запам'ятовуючий елемент з двома стійкими станами, зміна яких відбувається під дією вхідних сигналів
Умонтований канал (МОН-транзистора)	Канал з початковою додатковою концентрацією зарядів
Універсальний комп'ютер	Комп'ютер, операційні ресурси якого забезпечують виконання будь-якого алгоритму обробки інформації
Унітарний код	Двійковий код, який вміщує завжди тільки одну одиницю, а інші — нулі
Фіксатор	Див. "Елементарний регістр"
Флеш-пам'ять	Використовує елементи пам'яті на транзисторах ЛІЗМОН з електричним стиранням і записуванням інформації
Формат	Спосіб розміщення компонентів числа у розрядній сітці
Функціонально повна система функцій	Система функцій, суперпозицією яких може бути представлена будь-яка булева функція

Центральний пристрій керування	Сукупність вузлів і блоків процесора, які забезпечують координацію функціонування всіх пристроїв машини і керування ними для всіх прийнятих режимів роботи
Центральний процесор	Процесор, який виконує в обчислювальній системі основні функції
Цифро-аналоговий перетворювач	Пристрій, призначений для перетворення цифрової інформації в аналогову форму у вигляді напруги або струму
Час записування	Інтервал між моментами появи керуючого сигналу записування і установленням комірки пам'яті в стан, який задають вхідні сигнали
Час зчитування	Інтервал між моментами появи керуючого сигналу читання і даних на виході пам'яті
Шина	Сукупність сигнальних ліній, по яких передають сигнали однакового функціонального призначення
Шифратор	Функціональний вузол комп'ютера, призначений для перетворення вхідного унітарного коду у вихідний двійковий позиційний код
Штрих Шефера	Булева операція, результат якої дорівнює нулю тоді і тільки тоді, коли обидва операнди дорівнюють одиниці
Ядро комп'ютера	Процесор і оперативна пам'ять
D-тригер	Синхронний запам'ятовуючий елемент з двома стійкими станами і одним інформаційним <i>D</i> -входом
JK-тригер	Запам'ятовуючий елемент з двома стійкими станами та інформаційними входами <i>J</i> (аналог <i>S</i>) і <i>K</i> (аналог <i>R</i>), які забезпечують відповідно роздільну установку станів "1" і "0"
RS-тригер	Запам'ятовуючий елемент з роздільними інформаційними входами для установлення його в стан "0" (<i>R</i> -вхід) і в стан "1" (<i>S</i> -вхід)
T-тригер	Запам'ятовуючий елемент з двома стійкими станами та одним інформаційним <i>T</i> -входом. Стан <i>T</i> -тригера змінюється на протилежний після кожного надходження лічильного сигналу на <i>T</i> -вхід

Список літератури

1. *Алексенко А Г, Шагурин И И* Микросхемотехника – М Радио и связь 1990 496с
2. *Бабак В П, Хандецкий В С, Шрюфер Е* Обробка сигналів Підруч 2-е вид , перероб і доп – К Либідь 1999 – 496 с
3. *Бабич Н П, Нагорный Л Я* Элементы и узлы цифровых вычислительных машин Курс лекций – К КИИГА, 1971 – 209 с
4. *Бабич Н П, Кубицкий В И, Ефимец В Н* Импульсные и цифровые устройства Учеб пособ К КИИГА 1975 – 95 с
5. *Бабич Н П, Ефимец В Н* Элементы и узлы ЭЦВМ Учеб пособ – К КИИГА 1983 48 с
6. *Бабич Н П, Жуков И А* Компьютерная схемотехника Метод указания К КМУГА 2000 – 72 с
7. *Балашов Е П, Григорьев В Л, Петров Г А* Микро- и мини-ЭВМ Учеб пособ для вузов – Л Энергоатомиздат, 1984 – 376 с
8. *Башков Е П* Аппаратное и программное обеспечение зарубежных микро-ЭВМ Учеб пособ – К Вища шк , 1990 – 207 с
9. *Белоус А И, Блинков О Е, Силин А В* Биполярные микросхемы для интерфейсов систем автоматического управления – Л Машиностроение, 1990 272 с
10. *Бродин В Б, Шагурин И И* Микропроцессор i486 Архитектура, программирование интерфейс – М ДИАЛОГ-МИФИ, 1993 – 240 с
11. *Букреев И Н, Мансуров Б М* Микроэлектронные схемы цифровых устройств 3-е изд – М Радио и связь, 1990 – 415 с
12. *Вишенчук И М, Черкасский Н В* Алгоритмические операционные устройства и суперЭВМ – К Техника, 1990 – 197 с
13. *Гальперин М В* Практическая схемотехника в промышленной автоматике – М Энергоатомиздат, 1987 – 320 с
14. *Григорьев В Л* Программное обеспечение микропроцессорных систем - М Энергоатомиздат, 1983 – 208 с
15. *Григорьев В Л* Архитектура и программирование арифметического сопроцессора – М Энергоатомиздат, 1991 – 208 с
16. *Григорьев В Л* Микропроцессор i486 Архитектура и программирование (в 4-х книгах) – М Гранал, 1993 – Кн 1 – 346 с , кн 2, 3, 4 – 382 с
17. *Гук М* Аппаратные средства IBM PC Энциклопедия – СПб Питер Ком, 1999 – 816 с
18. *Гук М* Процессоры Pentium II, Pentium Pro и просто Pentium – СПб Питер Ком, 1999 – 288 с
19. *Гусев В Г, Гусев Ю М* Электроника Учеб пособ для приборостроит спец вузов – 2-е изд , перераб и доп – М Высш шк , 1991 – 622 с
20. *Евдокимов В Ф, Стасюк А И* Параллельные вычислительные структуры на основе разрядных методов вычислений – К Наук думка 1987 – 312 с
21. *Жабин В И, Ткаченко В В* Однокристалльные и микропрограммируемые ЭВМ / Под ред В И Корнейчука – К Диалектика, 1995 – 115 с
22. *Жабин В И, Ткаченко В В, Зайцев А А, Антэнов Р Л* Логические основы и схемотехника цифровых ЭВМ Практикум – К ВЕК+, 1999 – 128 с
23. *Жуков И А, Плющ Ю А* Математические модели вычислительных структур функ-

- ционально-ориентированных оперативных блоков. Препринт-95 – 461. – К.: ИМПЭ, НАН Украины, 1995. – 76 с.
24. *Изделия* электронной техники. Цифровые микросхемы. Микросхемы памяти. Микросхемы ЦАП и АЦП: Справ. / О.Н. Лебедев и др.; Под ред. А.И. Ладики и А.И. Сташкевича. – М.: Радио и связь, 1994. – 248 с.
 25. *Интегральные* микросхемы энергонезависимой памяти 28F008SA 28F008SA-L: INTEL CORPORATION: Пер. с англ. – М.: БИНОМ, 1992.–78 с.
 26. *Информатика*: Учеб. пособие для студ. пед. вузов / А.В. Могилев, Н.И. Пак, Е.К. Хеннер; Под ред. Е.К. Хеннера. – М.: Академия, 1999. – 816 с.
 27. *Каган Б.М.* Электронные вычислительные машины и системы: Учеб. пособ. для вузов. – 2-е изд., перераб. и доп. – М.: Энергоатомиздат, 1985. – 552 с.
 28. *Калабеков Б.А.* Цифровые устройства и микропроцессорные системы: Учеб. для техникумов связи. – М.: Горячая линия – Телеком, 2000. – 336 с.
 29. *Компьютерная* схемотехника: Лаб. практикум / Н.П. Бабич, Л.Я. Нагорный, И.А. Жуков. – К.: КМУГА, 1998. – 82 с.
 30. *Кончаловский В.Ю.* Цифровые измерительные устройства: Учеб. пособ. для вузов. – М.: Энергоатомиздат, 1985. – 304 с.
 31. *Корячко В.П.* Микропроцессоры и микро-ЭВМ в радиоэлектронных средствах: Учеб. для вузов. – М.: Высш. шк., 1990. – 407 с.
 32. *Кравец В.А., Шпильберг А.Я.* Зарубежные ЭВМ. Оборудование и программное обеспечение: Учеб. пособ. для иностр. студентов. – Харьков: Основа, 1991. – 216 с.
 33. *Лихтциндер Б.Я., Кузнецов В.Н.* Микропроцессоры и вычислительные устройства в радиотехнике: Учеб. пособ.. – К.: Вища шк., 1988. – 272 с.
 34. *Логические* ИС КР1533, КР1554: Справ.: В 2-х частях /И.И. Петровский, А.В. Прибыльский, А.А. Троян, В.С. Чувелев. – М.: БИНОМ, 1993. – 496 с.
 35. *Луцкий Г.М., Корочкин А.В.* Алгоритмы деления в полуавтономном режиме вычислений // Электронное моделирование. – 1982. – №5. – С. 9–15.
 36. *Майоров С.А., Новиков Г.И.* Принципы организации цифровых машин – Л.: Машиностроение, 1974. – 432 с.
 37. *Микропроцессоры*: В 3-х кн.: Учеб. для вузов / В.Д. Вернер, Н.В. Воробьев, А.В. Горячев и др.; Под ред. Л.Н. Преснухина. – М.: Высш. шк., 1986. – Кн.2 – 383 с.
 38. *Микропроцессоры* и микро-ЭВМ в системах автоматического управления: Справ. / С.Т. Хвощ, Н.Н. Варлинский, Е.А. Попов; Под ред. С.Т. Хвоща. – Л.: Машиностроение, 1987. – 640 с.
 39. *Микропроцессорный* комплект К1810: Структура, программирование, применение: Справ. книга / Ю.М. Казаринов и др.; Под ред. Ю.М. Казаринова. – М.: Высш. шк., 1990. – 269 с.
 40. *Микропроцессорный* комплект БИС серии К1810 для цифровой обработки сигналов: Справ. / А.И. Белоус, О.В. Подрубный, В.М. Журба; Под ред. А.И. Сухопарова. – М.: Радио и связь, 1992. – 256 с.
 41. *Мячев А.А.* Мини- и микроЭВМ систем обработки информации: Справ. – М.: Энергоатомиздат, 1991. – 304 с.
 42. *Мячев А.А., Степанов В.Н.* Персональные ЭВМ и микро-ЭВМ. Основы организации: Справ. /Под ред. А.А. Мячева. – М.: Радио и связь, 1991. – 320 с.
 43. *Нефедов А.В.* Интегральные микросхемы и их зарубежные аналоги: Справ. Т.3. – М.: КУБК-а, 1997. – 544 с.

44. *Прикладная теория цифровых автоматов* / К.Г. Самофалов, А.М. Романкевич, В.Н. Валуйский и др. – К.: Вища шк., 1987. – 375 с.
45. *Применение интегральных микросхем памяти: Справ.* / А.А. Дерюгин, В.В. Цыркин, В.Е. Красовский и др.; Под ред. А.Ю. Гордонова, А.А. Дерюгина. – М.: Радио и связь, 1994. – 232 с.
46. *Проектирование микропроцессорной электронно-вычислительной аппаратуры: Справ.* / В.Г. Артюхов, А.А. Будняк, В.Ю. Лапий и др. – К.: Техніка, 1988. – 263 с.
47. *Рабинович З.Л., Раманаускас В.А.* Типовые операции в вычислительных машинах. – К.: Техніка, 1980. – 264 с.
48. *Савельев А.Я.* Прикладная теория цифровых автоматов: Учеб. для вузов. – М.: Высш. шк., 1987. – 272 с.
49. *Самофалов К.Г., Викторов О.В.* Микропроцессоры. – Б-ка инженера. – 2-е изд., перераб. и доп. – К.: Техніка, 1989. – 312 с.
50. *Самофалов К.Г., Корнейчук В.И., Тарасенко В.П.* Цифровые ЭВМ: Теория и проектирование / Под общ. ред. К.Г. Самофалова. – 3-е изд., перераб. и доп. – К.: Вища шк., 1989. – 424 с.
51. *Сигорский В.П.* Математический аппарат инженера. – К.: Техніка, 1975. – 768 с.
52. *Соломатин Н.М.* Логические элементы ЭВМ: Практ. пособ. для вузов. 2-е изд., перераб. и доп. – М.: Высш. шк., 1990. – 160 с.
53. *Схемотехника ЭВМ: Учеб.* / Под ред. Г.Н. Соловьева. – М.: Высш. шк., 1985. – 391 с.
54. *Схемотехника БИС постоянных запоминающих устройств* / О.А. Петросян, И.Я. Козырь, Л.А. Колядов, Ю.И. Щетинин. – М.: Радио и связь, 1987. – 304 с.
55. *Схемотехника ЭВМ: Лаб. практикум* / Н.П. Бабич, В.И. Андреев, И.А. Жуков. – К.: КМУГА, 1997. – 245 с.
56. *Узрюмов Е.П.* Проектирование элементов и узлов ЭВМ: Учеб. пособ. – М.: Высш. шк., 1987. – 318 с.
57. *Узрюмов Е.П.* Цифровая схемотехника. – СПб.: БХВ – Петербург, 2001. – 528 с.
58. *Хоровиц П., Хилл У.* Искусство схемотехники: В 3-х томах: Т.3. Пер. с англ. – 4-е изд., перераб. и доп. – М.: Мир, 1993. – 367 с.
59. *Цапенко М.П.* Измерительные информационные системы: Учеб. пособ. для вузов. – 2-е изд., перераб. и доп. – М.: Энергоатомиздат, 1985. – 439 с.
60. *Цифровые устройства и микропроцессорные системы. Задачи и упражнения: Учеб. пособ. для вузов.* / Л.М. Гольденберг, В.А. Малев, Г.Б. Малько. – М.: Радио и связь, 1992. – 256 с.
61. *Шагурин И.И., Бродин В.Б., Мозговой Г.П.* 80386: описание и система команд. – М.: Малип, 1992. – 160 с.
62. *Шевкопляс Б.В.* Микропроцессорные структуры. Инженерные решения. – М.: Радио и связь, 1986. – 264 с.
63. *Шило В.Л.* Популярные микросхемы КМОП: Справ.. – М.: Ягуар, 1993. – 64 с.
64. *Электроника СБИС. Проектирование микроструктур: Пер. с англ.* / Под ред. Н. Айнспрука. – М.: Мир, 1989. – 256 с.
65. *Юшин А.М.* Цифровые микросхемы для электронных устройств: Справ. для ПТУ. – М.: Высш. шк., 1993. – 176 с.
66. *Microprocessors.* Intel Corporation. – New-York, 1990. – 1424 с.

1 0 0 1 1 0 1 0 0 0 1 1 0 0 0 0
1 0 0 0 1 0 0 1 0 1 0 0 0 0 0 1
0 1 1 1 1 0 0 0 0 1 0 1 0 0 1 0
0 1 1 0 0 1 1 1 0 1 1 0 0 0 1 1
0 1 0 1 0 1 1 0 0 1 1 1 0 1 0 0



Бабич Микола Павлович - кандидат технічних наук, професор кафедри обчислювальної техніки Інституту інформатики Національного авіаційного університету (40 років педагогічного стажу). Область наукових і педагогічних інтересів: комп'ютерна схемотехніка, мікропроцесорні системи, комп'ютерна електроніка. Автор 45 публікацій, з них три навчальних посібники.



Жуков Ігор Анатолійович - доктор технічних наук, професор, заслужений винахідник України, директор Інституту інформатики Національного авіаційного університету (30 років науково-педагогічного стажу). Область наукових і педагогічних інтересів: обчислювальні машини, системи і мережі та апаратно-програмні засоби підвищення їх продуктивності. Автор 195 публікацій.

Систематизовано і викладено з урахуванням багаторічного педагогічного досвіду роботи в НАУ основи теорії і практичної реалізації цифрової схемотехніки згідно з навчальним планом підготовки студентів у напрямку "Комп'ютерна інженерія".

Розглянуті теоретичні основи комп'ютерної схемотехніки (інформатика, арифметика, логіка), методи аналізу і синтезу логічних і запам'ятовуючих елементів, комбінаційних і послідовнісних функціональних вузлів.

Коротко подана історія розвитку комп'ютерів, їх архітектура та основні структурні пристрої: процесори, мікропроцесори, пам'ять, арифметико-логічні, інтерфейси та програмовані контролери.

Книга написана доступною мовою, багато ілюстрована і може бути корисна широкому колу читачів.