

688.31(075)
ПЗ1

Анатолій Петух, Дем'ян Обідник

ЕОМ І МІКРОПРОЦЕСОРНІ СИСТЕМИ

Міністерство освіти і науки України
Вінницький державний технічний університет

Анатолій Петух, Дем'ян Обідник

ЕОМ І МІКРОПРОЦЕСОРНІ СИСТЕМИ

Затверджено Ученою радою Вінницького державного технічного університету як навчальний посібник для студентів спеціальностей "Програмне забезпечення автоматизованих систем" та "Інтелектуальні системи прийняття рішень" денної та заочної форм навчання. Протокол № 5 від 27.12.2000 р.

Вінниця ВДТУ 2001

Рецензенти:

О.Д.Азаров, доктор технічних наук, професор

В.Є.Качуровський, кандидат технічних наук, доцент

О.М.Роїк, кандидат технічних наук, доцент

Рекомендовано до видання Ученою радою Вінницького державного технічного університету Міністерства освіти і науки України

А.М.Петух, Д.Т.Обідник

ПЗ1 ЕОМ і мікропроцесорні системи.

Навчальний посібник.- Вінниця: ВДТУ, 2001. - 124 с.

В навчальному посібнику розглянуто зміст дисципліни “ЕОМ і мікропроцесорні системи”, викладені основні положення, наведені приклади і рекомендації щодо їх засвоєння, приведені рекомендації розширеного опанування. Приведені контрольні запитання, задачі і приклади, рекомендації щодо лабораторного практикуму і виконання курсової роботи.

Навчальний посібник призначений для студентів спеціальностей “Програмне забезпечення автоматизованих систем” та “Інтелектуальні системи прийняття рішень” денної та заочної форм навчання. Він також може бути використаний студентами спеціальності “Комп’ютерні системи та мережі” для опанування дисциплін “Цифрові ЕОМ” та “Архітектура обчислювальних машин”.

ВСТУП

Дисципліна "ЕОМ і мікропроцесорні системи" передбачена навчальним планом для студентів бакалаврського напрямку "Комп'ютерні науки". Вона базується на таких дисциплінах, як

- організація і функціонування ЕОМ та систем,
- основи програмування та алгоритмічні мови,
- прикладна теорія цифрових автоматів,
- схемотехніка ЕОМ,
- системне програмування і операційні системи.

Забезпечує дисципліни

- методи та засоби комп'ютерних інформаційних технологій,
- комп'ютерні мережі,
- мікропроцесори,
- основи автоматизованого проектування складних об'єктів і систем,
- САПР засобів обчислювальної техніки.

Метою дисципліни є викладення студентам систематизованої інформації про архітектуру і принципи обробки інформації в сучасних і перспективних обчислювальних системах, принципи побудови і функціонування основних блоків сучасних ЕОМ і мікропроцесорних систем.

В результаті вивчення дисципліни набуваються знання: особливостей архітектури сучасних ЕОМ; типів, основних характеристик, принципів побудови та галузей застосування запам'ятовувальних пристроїв; структур та принципів роботи арифметико-логічних пристроїв; структур, особливостей побудови, принципів роботи керуючих пристроїв; основних характеристик і особливостей систем переривання сучасних ЕОМ; характеристик та особливостей інтерфейсу введення-виведення; принципів побудови та функціонування паралельних обчислювальних систем; організації сучасних мікропроцесорів та мікропроцесорних систем.

В навчальному посібнику поєднані основні принципи побудови і функціонування блоків ЕОМ, які стали класичними, та особливості їх реалізації в сучасних засобах обчислювальної техніки.

Поєднання в одному посібнику викладення теоретичного матеріалу дисципліни, рекомендацій, контрольних запитань, задач, прикладів, лабораторного практикуму і рекомендацій до виконання курсової роботи особливо актуальне для студентів заочної форми навчання, де високий відсоток самостійної роботи при вивченні дисциплін, передбачених навчальним планом.

Автори висловлюють подяку студентам гр. 1-ІТП-95 Сенюк А.С. та 2-ІТП-95 Беркович І.А., які в дипломних роботах відпрацювали і дослідили питання, пов'язані з постановкою лабораторного практикуму та виконанням курсової роботи з дисципліни "ЕОМ і мікропроцесорні системи".

1. АРХІТЕКТУРНІ ОСОБЛИВОСТІ ЦИФРОВИХ ЕОМ

Інформацію, втілену і зафіксовану в деякій матеріальній формі, називають повідомленням. Повідомлення можуть бути безперервними (аналоговими) і дискретними (цифровими). Безперервні повідомлення зображаються деякою фізичною величиною (електричною напругою, струмом та ін.), зміна якої в часі відображає протікання певного процесу. Фізична величина, яка передає безперервне повідомлення, може в визначеному інтервалі приймати будь-які значення і змінюватись в довільні моменти часу. Шляхом квантування безперервних повідомлень за рівнем і часом їх можна замінювати дискретними, коли окремим елементам інформації можуть бути присвоєні числові (цифрові) значення.

Електронні обчислювальні машини (ЕОМ), інакше комп'ютери, є перетворювачами інформації. Вони являють собою комплекс технічних засобів, який в відповідності з програмою реалізує обробку інформації. В відповідності з формою подання інформації ЕОМ поділяються на два класи: безперервної дії (аналогові) та дискретної дії (цифрові). Якраз цифрові ЕОМ і є предметом подальшого розгляду.

Поняття архітектури ЕОМ включає сукупність її властивостей і характеристик з точки зору користувача, воно об'єднує апаратні, мікропрограмні та програмні засоби обчислювальної техніки. Складовою архітектури є структура, яка визначає складові частини системи і взаємозв'язок між ними.

В ЕОМ широко використовується ієрархічний спосіб побудови, який характеризується наявністю певних неподільних одиниць, що будучи об'єднаними в систему певного рівня, можуть розглядатись в якості елементів для систем більш високого рівня. Розрізняють 5 ієрархічних рівнів побудови ЕОМ: рівень електричних схем, рівень логічних схем (логічних елементів), рівень операційних схем (базових вузлів), рівень структурних схем (блоків або пристроїв) та програмний рівень.

До основних принципів побудови ЕОМ відносять принцип програмного керування (Чарльз Беббідж, 1833 р.), принцип зберігання програми в пам'яті (Дж.фон Нейман, 1945 р.), принцип умовного переходу, принцип ієрархічності запам'ятовувальних пристроїв, принцип використання двійкової системи числення.

Основними характеристиками ЕОМ є продуктивність, коефіцієнт ефективності, розрядність, ємність пам'яті, швидкість виконання основних типів команд, максимальна швидкість обміну інформацією між ядром і периферійними пристроями, надійність, вагогабаритні показники, вартість, сумісність апаратних і програмних засобів з іншими ЕОМ.

Продуктивність визначають середнім значенням кількості виконуваних операцій за одиницю часу (оп/сек). В зв'язку з тим, що різні групи операцій за часом виконання суттєво відрізняються, розрізняють такі міри продуктивності: MIPS (Mega Instructions Per Second) - мільйон інструкцій (команд процесора) в секунду; - MFLOPS (Million Floating Point

Operations Per Second) - мільйон операцій з плаваючою точкою за секунду. Коефіцієнт ефективності визначає відношення продуктивності машини до її вартості. Надійність визначають середнім часом роботи на одну відмову або збій та витратами часу на їх усунення.

В залежності від властивостей ЕОМ існують їх різноманітні класифікації: за призначенням розрізняють універсальні та спеціалізовані ЕОМ; за умовами експлуатації – стаціонарні та ті, що транспортуються; за швидкістю – низької швидкодії, середньої, високої та супер-ЕОМ (цей показник постійно змінюється з часом); за способом передачі кодів чисел - послідовної дії, паралельної дії та паралельно-послідовної дії; за організацією структури – одномагістральні і багатемагістральні; за кількістю програм, що одночасно виконуються – однопрограмні та багатопрограмні; за кількістю процесорів – однопроцесорні та багатопроцесорні; за вагогабаритними показниками – стаціонарні, настільні, портативні, кишенькові.

Покоління ЕОМ

Обчислювальна техніка має багату історію. Спочатку її засоби були механічними. Великий вклад в їх створення внесли Леонардо да Вінчі, Вільгельм Шікард, Блез Паскаль, Г. Лейбніц, Е. Якобсон, Фельт, У. Беррроуз, П.Л. Чебишев, Джованні Пелен, В.Т. Однер, Г. Холлеріт, Ч. Беббідж. Аналітична машина Ч. Беббіджа була першою машиною з програмним керуванням. Вперше програми для неї були складені дочкою лорда Дж. Байрона Адою Лавлейс, яку вважають першим програмістом.

Вперше ЕОМ на основі електромагнітних реле з керуванням від перфострічки були побудовані в 1942 р. К. Цюзе в Німеччині та в 1944 р. Г.Айкеном в США.. Перші ЕОМ з програмним керуванням і програмою, що зберігалась в пам'яті, з'явилися практично одночасно в Англії, США та СССР. Фундаментальний вклад в розвиток вітчизняної обчислювальної техніки вніс акад. С.О.Лебедев. Під його керівництвом в 1949 – 1951 рр. в АН УРСР в Києві була побудована перша в СССР ЕОМ – Мала Електронна Обчислювальна Машина ("МЭСМ"), а в 1952 – 1954 рр. – в Москві – Швидкодіюча Електронна Обчислювальна Машина ("БЭСМ"), яка виконувала 8000 оп/сек. і була в той час однією з найшвидкодіючих ЕОМ в світі. Протягом останніх п'яти десятиріч електронна обчислювальна техніка бурхливо розвивається. Поява нових поколінь ЕОМ викликала розширення галузей і розвитком методів їх застосування, які вимагали більш продуктивних, дешевих і надійних машин.

Покоління ЕОМ визначається сукупністю взаємопов'язаних суттєвих особливостей і характеристик конструктивно-технологічної (в першу чергу елементної) бази і машинної архітектури. Архітектурні особливості ЕОМ відповідно до поколінь приведені в табл.1.1.

Архітектурні особливості поколінь ЕОМ

Покоління	Роки	Елементна база	Архітектурні особливості
1	1946 - 1960	Електронні лампи	Високий рівень сигналів (сотні вольт), навісний монтаж, висока споживча потужність, відносно мала ємність пам'яті, низька надійність, програмування в машинних кодах, низька ефективність в процесі налагодження програм
2	1960 - 1965	Напівпровідникові прилади	Рівень сигналів – десятки вольт, друкований монтаж, ОЗП на магнітних осердях, мови високого рівня, пакетний режим роботи, машини для керування
3	1965 - 1970	Інтегральні схеми	Рівень сигналів – одиниці вольт, серії апаратно та програмно сумісних машин, наявність каналів введення-виведення, операційні системи, режими діалогу, розділення часу, реального часу, мульти-програмний режим, віртуальна пам'ять
4	1970 -	Великі інтегральні схеми	Багатошаровий друкований монтаж, апаратна реалізація операцій, багатопроцесорні машини, підвищення рівня машинної мови, персональні ЕОМ, мікро-ЕОМ
5	1981 - проект	Надвеликі інтегральні схеми	Суттєве поліпшення людино-машинного інтерфейсу, автоматизація складання програм, обробка інформації на основі знань

Узагальнені структури ЕОМ

До основних пристроїв, які входять до складу узагальнених структур різноманітних ЕОМ, відносять:

- **АЛП** – арифметико-логічний пристрій, здійснює перетворення кодів чисел (операндів) шляхом виконання над ними арифметичних, логічних та інших операцій;
- **ПК** – пристрій керування, приймає і інтерпретує коди команд і виробляє послідовності всіх функціональних керуючих сигналів, які визначаються командою;

- **ОЗП** – оперативний запам'ятовувальний пристрій (інакше ОП – оперативна або основна пам'ять), призначений для зберігання програм, даних, проміжної інформації і результатів в процесі вирішення задач;
- **Пвв** – пристрої введення інформації;
- **Пвив** – пристрої виведення інформації;
- **П** – процесор, включає пристрої АЛП та ПК;
- **ПП** – периферійні пристрої;
- **ПКПП** – пристрої керування периферійними пристроями;
- **ЗЗП** – зовнішні запам'ятовувальні пристрої;
- **ЦПК** – центральний пульт керування;
- **СК** – селекторний канал, є периферійним процесором, який в монопольному режимі обслуговує швидкодіючі ПП;
- **МК** – мультиплексний канал, є периферійним процесором, який одночасно в режимі розподілу часу обслуговує повільнодіючі ПП.

На рис. 1.1 приведена структурна схема ЕОМ першого та другого поколінь. Головним протиріччям цієї схеми є протиріччя між відносно високою швидкодією процесора і повільним введенням-виведенням інформації. Оскільки ПК керує введенням-виведенням, в цей час процесор практично простоює. Велику роль в структурі машин першого та другого поколінь мав ЦПК, за допомогою якого здійснювалось налагодження програм та діагностика неполадок. Вищевказане протиріччя було усунуто в ЕОМ третього покоління шляхом введення в структуру периферійних процесорів – каналів введення-виведення, які функціонують незалежно від центрального процесора і дозволяють забезпечити одночасну обробку інформації в центральному процесорі з операціями введення-виведення. Узагальнена структура машин третього покоління приведена на рис. 1.2. В четвертому поколінні ЕОМ широкого розповсюдження набули одношинна структура, показана на рис. 1.3 та структура багатопроцесорної обчислювальної системи, показана на рис. 1.4. Одношинна (магістрально-модульна) структура характеризується простотою зв'язків між пристроями, але вона є ефективною тільки при відносно невеликій кількості блоків, під'єднаних до шини. Така структура широко використовується в мікропроцесорних системах та персональних комп'ютерах. Багатопроцесорна обчислювальна система містить інформаційний комутатор, який дозволяє кожному процесору встановлювати зв'язок з іншими модулями і є досить складним пристроєм (комутація багаторозрядних шин).

Основними шляхами підвищення продуктивності обчислювальних систем є удосконалення технології елементної бази і пристроїв, використання різноманітних форм паралелізму, розвиток систем колективного користування, спеціалізація обчислювальних систем і їх компонент.

Конкретні архітектурні особливості реалізації принципів побудови пристроїв ЕОМ розглядаються в наступних розділах.

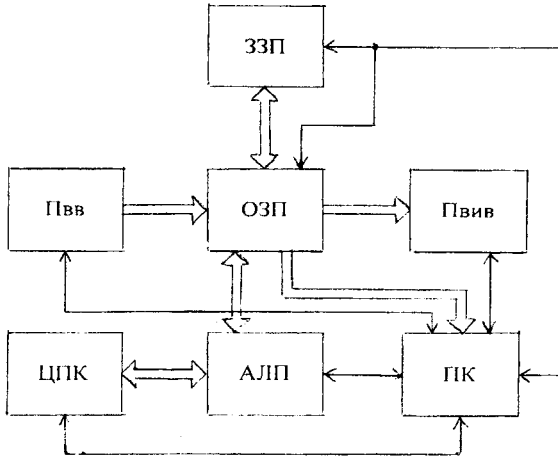


Рис.1.1. Узагальнена структура ЕОМ 1-го та 2-го поколінь

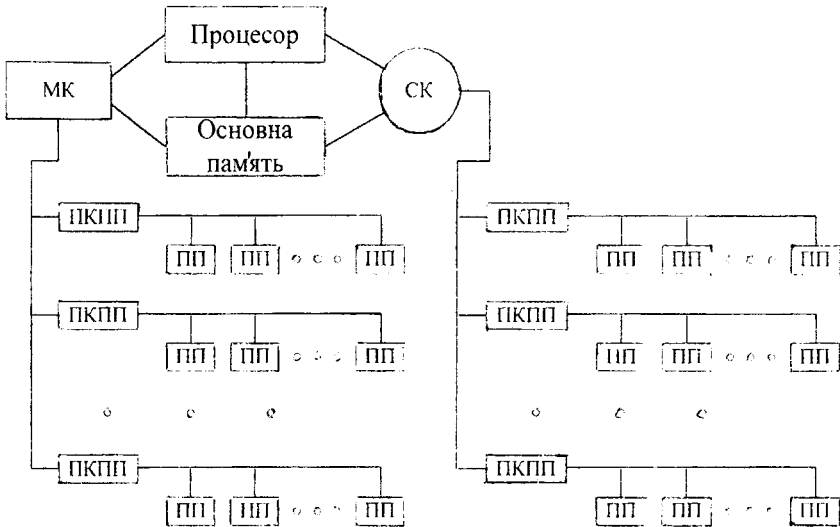


Рис.1.2. Узагальнена структура ЕОМ третього покоління

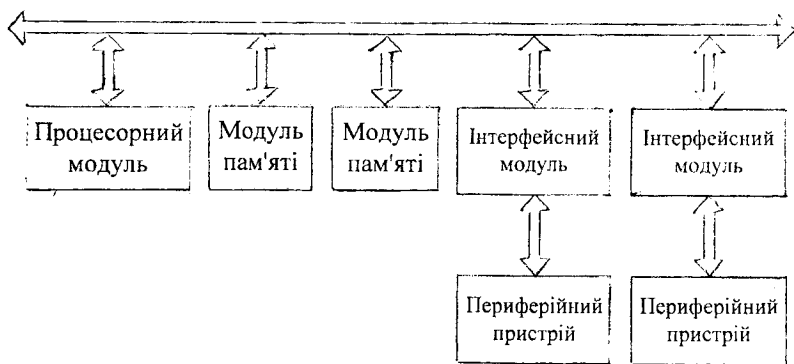


Рис. 1.3. Магістрально-модульна структура ЕОМ

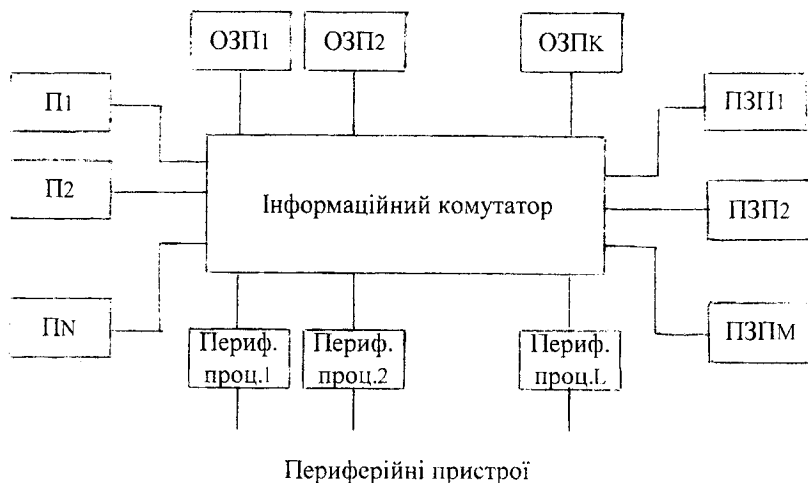


Рис. 1.4. Структура багатопроцесорної обчислювальної системи

Контрольні запитання і задачі

1. Охарактеризуйте основні принципи побудови і функціонування ЕОМ.
2. Приведіть класифікацію ЕОМ з характеристикою їх властивостей.
3. Проаналізуйте основні характеристики сучасних ЕОМ.
4. Приведіть аналіз структурних властивостей ЕОМ різних поколінь.
5. Охарактеризуйте основні блоки структур ЕОМ в сучасній інтерпретації.
6. Що таке ієрархічний спосіб побудови, які рівні ієрархії виділяють для ЕОМ?
7. Хто був першим програмістом?
8. Хто вперше запропонував принцип програмного керування?

2. ЗАПАМ'ЯТОВУВАЛЬНІ ПРИСТРОЇ ЕОМ

Запам'ятовувальні пристрої (ЗП) призначені для фіксації, зберігання і видачі інформації. Основними операціями з ЗП є занесення інформації і її читання, які називають звертанням до пам'яті.

Основними параметрами ЗП є:

- **інформаційна ємність** - максимально можливий обсяг інформації, що може зберігатися. Виражається в бітах, байтах, словах та кратних їм одиницях ($1\text{Кбіт}=2^{10}\text{біт}$, $1\text{Мбіт}=2^{20}\text{біт}$);
- **організація ЗП** - показує максимальну кількість слів, які можуть зберігатися в пам'яті, та їх розрядність і виражається в добуткові цих показників;
- **швидкодія ЗП** оцінюється часом зчитування, занесення і тривалістю циклів читання/занесення;
- **час зчитування** – інтервал часу між моментом подачі сигналу читання і появою прочитаного слова на виході ЗП;
- **час занесення** - інтервал часу між моментом подачі сигналу занесення, достатній для установки запам'ятовувальних комірок в стан, який відповідає вхідному слову інформації;
- **цикл пам'яті** – мінімально припустимий інтервал між послідовними читаннями або занесеннями. Тривалість циклу може перевищувати час читання або занесення, оскільки після цих операцій може виникнути потреба в відновленні початкового стану ЗП.

До другорядних параметрів ЗП (які при певних умовах можуть стати головними) відносять надійність, вагогабаритні показники, споживчу потужність, механічні і кліматичні характеристики, стійкість проти впливу зовнішніх факторів, вартість (питому вартість) та ін. Окрім цього існує ряд так званих режимних параметрів, забезпечення яких необхідне для нормального функціонування ЗП, і які визначають тривалості і обмеження по взаємному положенню керуючих сигналів в часі.

В ієрархічній структурі ЗП в складі ЕОМ розрізняють оперативну або основну пам'ять (ОП), надоперативну пам'ять (НОП, інакше буферну або кеш-пам'ять), зовнішні запам'ятовувальні пристрої (ЗЗП), постійні запам'ятовувальні пристрої (ПЗП).

Оперативна пам'ять призначена для зберігання програм і даних в процесі вирішення задачі. ОП має відносно велику ємність і є зовнішньою по відношенню до процесора. За швидкодією займає проміжне значення між НОП та ЗЗП.

Кеш-пам'ять складається з відносно невеликого швидкодіючого запам'ятовувального пристрою, який вміщує частину інформації ОП. Такою інформацією можуть бути адреси, команди або дані, які найбільш інтенсивно використовуються на поточному етапі вирішення задачі.

Зовнішні запам'ятовувальні пристрої призначені для зберігання великих масивів інформації. Порівняно з ОП вони є менш

швидкодіючими, але звичайно мають значно більшу сміність.

Постійні запам'ятовувальні пристрої використовуються для енергонезалежного збереження системної інформації і в робочому режимі допускають тільки операцію читання.

В залежності від способу організації доступу розрізняють пристрої пам'яті з безпосереднім (довільним), прямим (циклічним) та послідовним доступами. В пам'яті з безпосереднім доступом час доступу, а тому і цикл звертання не залежать від місця розміщення ділянки пам'яті, звідки зчитується або куди записується інформація. В пристроях пам'яті з прямим доступом, до яких належать пристрої з магнітними дисками, завдяки безперервному обертанню носія інформації можливість звертання до деякої ділянки носія циклічно повторюється. В пам'яті з послідовним доступом здійснюється послідовний прохід ділянок носія інформації, поки потрібна ділянка носія не займе деяке положення, з якого відкривається доступ до місця розміщення інформації. Типовим прикладом такої пам'яті є ЗП на магнітній стрічці.

Структури адресних ЗП з довільним доступом

В адресних ЗП звертання до пам'яті здійснюється за адресою, яка визначає номер комірки, де зберігається одне слово пам'яті. Найбільш поширеними структурами адресних ЗП є структури типу 2D, 3D та 2DM (від англійського dimension – розмірність).

Запам'ятовувальний пристрій будь-якого типу складається з масиву запам'ятовувальних елементів, який називають накопичувачем, і блоків, що призначені для пошуку в масиві, занесення і читання інформації.

Структура ЗП типу 2D приведена на рис. 2.1. m -розрядна адреса

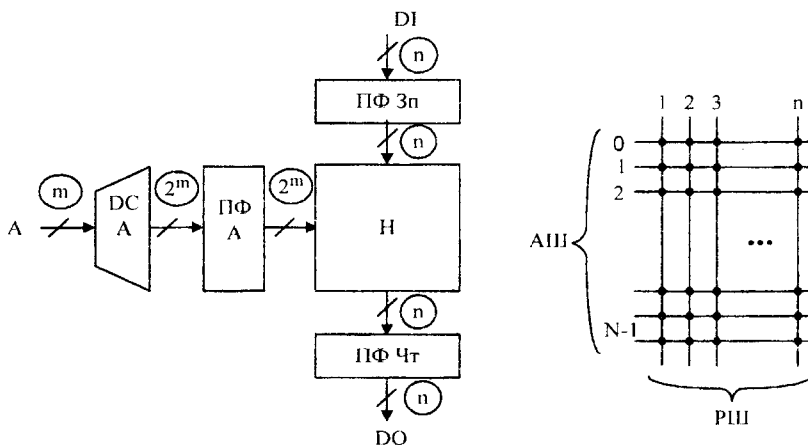


Рис. 2.1. Структура ЗП типу 2D

(А) дешифрується дешифратором адреси (DCA), вихідні сигнали якого підсилювачем-формувавцем (ПФА) перетворюються в сигнали адресної шини (АШ) накопичувача (Н) для активізації відповідної комірки пам'яті. Інформація, яку потрібно записати в пам'ять, поступає на n-розрядну вхідну шину даних (DI), звідки проходить через підсилювачі-формувачі запису (ПФЗп) на розрядні шини запису накопичувача. Прочитана інформація з розрядних шин читання накопичувача через підсилювачі-формувачі читання (ПФЧт) поступає на вихідну шину даних (DO). Підсилювачі-формувачі служать для узгодження параметрів елементів накопичувача з параметрами блоків, зовнішніх по відношенню до нього. Структура накопичувача типу 2D приведена справа на рис. 2.1. Тут крапками показані запам'ятовувальні елементи. Умовно вважають, що накопичувач 3П типу 2D являє собою вузьку довгу стрічку, де вертикальних шин (розрядів) відносно мало (десятки), а горизонтальних (адресних) шин відносно багато (сотні, тисячі, десятки тисяч). В зв'язку з тим, що запам'ятовувальних елементів на адресній шині відносно небагато і ємність шини невелика, 3П типу 2D характеризуються високою швидкодією. Однак необхідність в використанні великої кількості підсилювачів-формувачів адресних шин та складність дешифратора адреси значно обмежують ємність 3П типу 2D.

Структура 3П типу 3D приведена на рис. 2.2. В таких 3П адреса розділяється на дві частини, які незалежно дешифруються, а останній ступінь

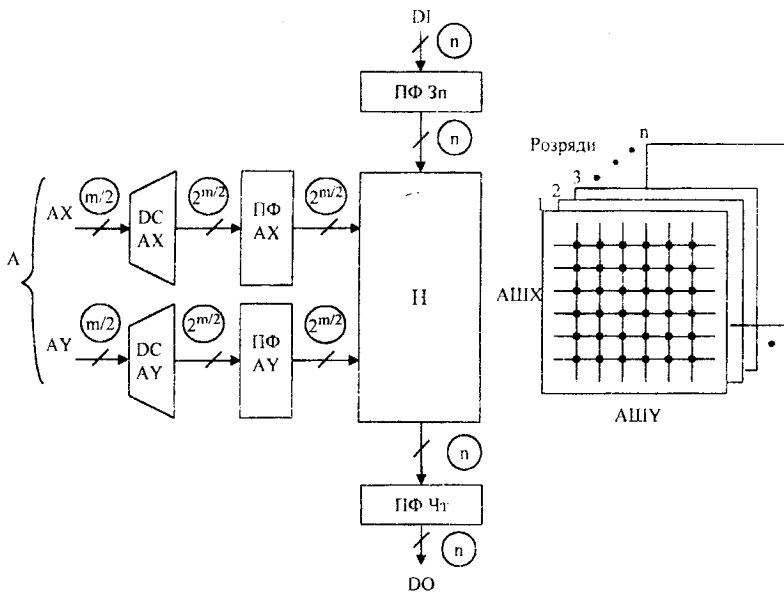


Рис. 2.2. Структура 3П типу 3D

дешифрації адреси реалізується безпосередньо в накопичувачі. При цьому дещо ускладнюється сам накопичувач, але різко скорочується складність дешифраторів адреси і кількість адресних підсилювачів-формуваців, що дає змогу реалізувати ЗП великої ємності. Накопичувач в ЗП типу 3D можна представити в вигляді набору матриць, кожна з яких відноситься до одного розряду. Горизонтальні і вертикальні адресні шини (АШХ та АШУ) пронизують всі матриці. В зв'язку з цим кількість запам'ятовувальних елементів на адресній шині порівняно з ЗП типу 2D значно зростає. Це призводить до підвищення вимог щодо навантажувальної здатності адресних підсилювачів-формуваців і дещо знижує швидкодію.

В ЗП типу 2DM (рис. 2.3) порівняно з ЗП типу 2D кількість горизонтальних (адресних) шин менша в K раз, а кількість вертикальних (які тут називаються адресно-розрядними - АРШ) в стільки ж раз більша. Формування сигналів на АРШ здійснюється за допомогою комутатора (КОМ), який вибирає одну з K вертикальних шин в кожному розряді. Структура ЗП типу 2DM найбільш зручна для побудови напівпровідникових ЗП і широко використовується як в оперативних, так і в постійних ЗП.

Структурна організація ОП

В структурному відношенні стосовно організації ОП розрізняють багатоблокову пам'ять, пам'ять з багатоканальним доступом та пам'ять з розшаруванням звертань.

Багатоблокова пам'ять складається, як правило, з 2^K блоків. Адреса такої пам'яті складається з двох частин, одна з яких визначає K -розрядну адресу блока, а інша – адресу комірки в межах блока. Організація керування пам'яттю передбачає дешифрацію K -розрядної адреси блока i , в відповідності до цього, вибір відповідного блока для звертання, формування сигналів читання/запису, об'єднання відповідно вхідних і вихідних шин даних блоків, адресної шини блоків. При використанні в якості блоків мікросхем пам'яті вибір блока здійснюється по входу вибору кристала (CS), а виходи даних об'єднуються за схемою третього стану. Багатоблокова пам'ять має ємність, яка дорівнює сумарній ємності всіх блоків, а швидкодія наближується до швидкодії одного блока.

В пам'яті з багатоканальним доступом каналами називають споживачів ресурсів пам'яті. Відносно до ОП такими споживачами є процесор, контролер прямого доступу до пам'яті, периферійні процесори, периферійні пристрої. При необхідності звертання до ОП канали виставляють активними сигнали запитів. Найбільш поширена дисципліна обслуговування каналів згідно з їх пріоритетами. Канал отримує доступ до ресурсів пам'яті, якщо нема запитів від каналів з вищими пріоритетами, є запит від даного каналу і не почате обслуговування каналів з нижчими пріоритетами.

Пам'ять з розширенням звертань є багатоблоковою. Вона допускає одночасне звертання декількох каналів до різних блоків. Основною склад-

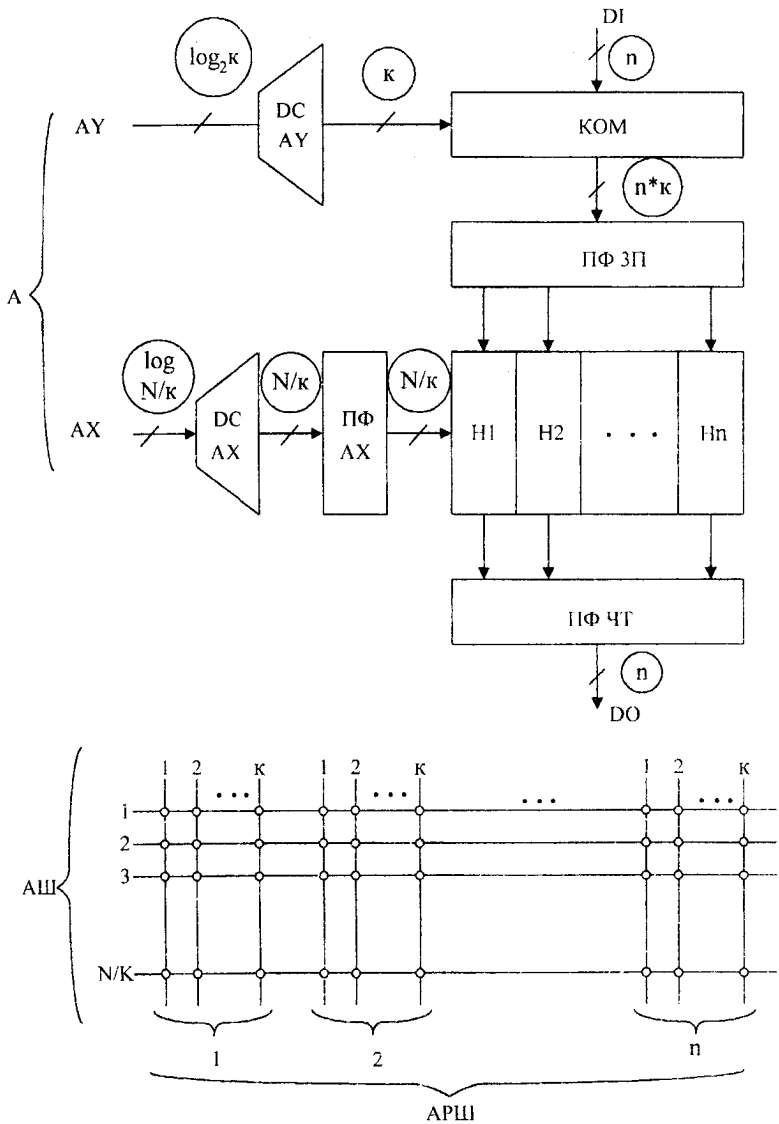


Рис.2.3. Структура 3П типу 2D-M

ністю при організації такої пам'яті є забезпечення відповідної комутації багаторозрядних шин між каналами та блоками. Для ефективного використання ресурсів пам'яті з розширенням звертань потрібна

відповідна організація обчислювального процесу, яка повинна забезпечувати такий розподіл інформації між блоками, щоб при вирішенні задачі одночасно була задіяна максимальна кількість блоків. Середня кількість звертань до пам'яті, які можуть бути прийняті на обслуговування одночасно, характеризується коефіцієнтом розшарування. Мінімальне значення цього коефіцієнта дорівнює одиниці, а максимальне – кількості блоків пам'яті. Експериментальним шляхом встановлено, що в середньому для універсальних ЕОМ коефіцієнт розшарування приблизно дорівнює половині кількості блоків пам'яті.

Стекові ЗП

В стекових ЗП комірки утворюють одновимірний масив, в якому сусідні комірки зв'язані одна з одною розрядними шинами передачі слів.

Стекова пам'ять є безадресною по відношенню до користувача, її ресурсів і має жорстко регламентований порядок звертання. В залежності від цього порядку розрізняють два типи стека: стек FIFO та стек LIFO. Для стека FIFO (First In - First Out) порядок зчитування підлягає правилу: "перший поступив – перший вийшов", а для стека LIFO (Last In - First Out): "останній поступив – перший вийшов".

В структурі стекових ЗП окрім взаємопов'язаних комірок передбачають також лічильник-вказівник стека (ЛВС), який формує ознаки порожнього стека (коли в пам'ять не записано ні одного слова) і заповненого стека (коли всі комірки пам'яті заповнені). Ці ознаки використовуються при керуванні роботою стекових ЗП. З порожнього стека не можна прочитати, а в заповнений стек не можна записати. Узагальнені структури стекових ЗП приведені на рис. 2.4.

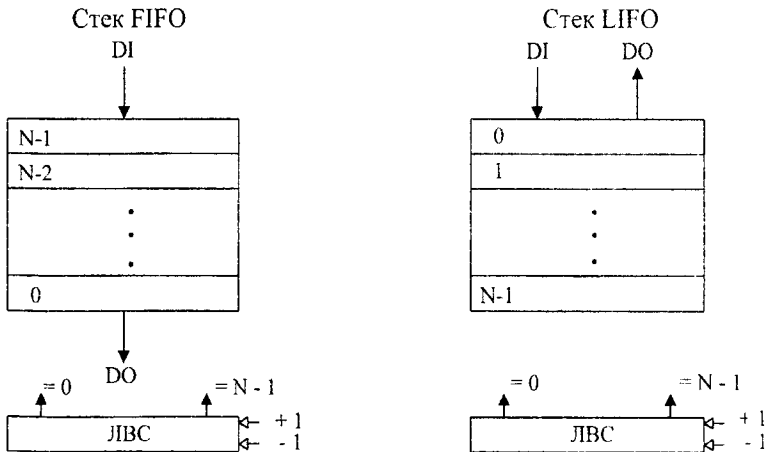


Рис. 2.4. Узагальнені структури стекових ЗП

Стекова пам'ять широко використовується в ЕОМ для апаратної реалізації різноманітних черг (стек FIFO), а також є досить ефективною при обробці вкладених структур даних (стек LIFO).

Асоціативні ЗП

В асоціативних ЗП (АЗП) пошук необхідної інформації здійснюється не за адресою, а за її змістом (за асоціативною ознакою - АО). При цьому пошук за АО (або за окремими розрядами цієї АО) здійснюється одночасно для всіх комірок запам'ятовувального масиву (ЗМ), який в кожній комірці містить схему порівняння.

Типова структура АЗП приведена на рис. 2.5. Вона включає ЗМ, реєстр асоціативної ознаки (PrAO), реєстр маски (PrM), реєстр інформації (PrI), комбінаційну схему (КС), реєстр результатів порівняння (PrРП) та формувач ознак порівняння (ФО). В ЗМ в n -му розряді зберігаються ознаки зайнятості комірок (0 – комірка

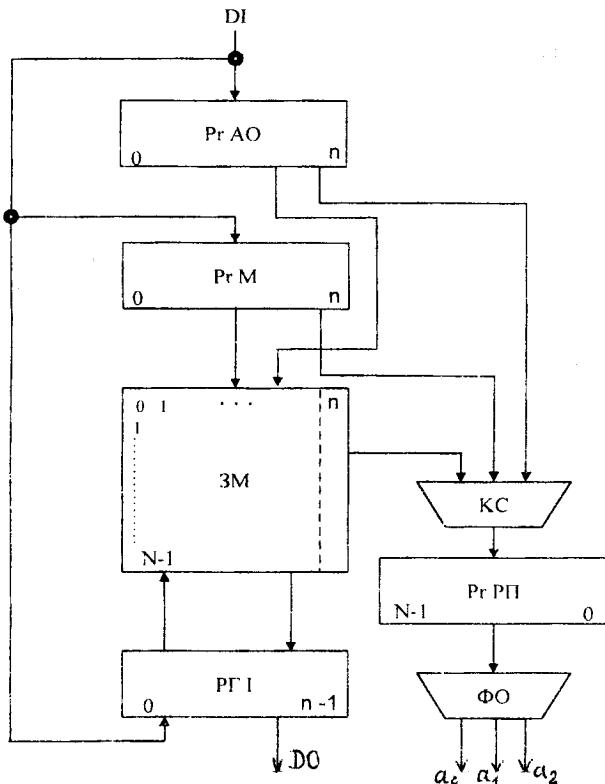


Рис. 2.5. Типова структура асоціативного запам'ятовувального пристрою.

вільна, 1 – зайнята). З вхідної інформаційної шини DI в R_{GAO} (розряди $0 : n-1$) записують n -розрядний асоціативний запит, а в R_{GM} – код маски пошуку, при цьому n -й розряд R_{GM} встановлюють в 0. Асоціативний пошук здійснюється тільки для тих розрядів R_{GAO} , яким відповідають 1 в R_{GM} (незамасковані розряди R_{GAO}). Для слів $ЗМ$, в яких значення в розрядах однакові з незамаскованими розрядами R_{GAO} , $КС$ встановлює 1 в відповідні розряди R_{RPI} і 0 – в решту розрядів. $ФО$ в залежності від стану R_{RPI} формує сигнали a_0 , a_1 , і a_2 , які відповідають випадкам відсутності слів в $ЗМ$, що задовольняють асоціативну ознаку, наявності одного або більше ніж одного такого слова. Операцію формування ознак a_0 , a_1 , і a_2 називають операцією контролю асоціації. Вона є складовою операцій читання і запису та має також самостійне значення.

При читанні спочатку здійснюють контроль асоціації за асоціативною ознакою в R_{GAO} . При $a_0=1$ читання відміняється в зв'язку з відсутністю потрібної інформації в $ЗМ$. При $a_1=1$ розшукане слово записується в R_{GI} і поступає на вихідну шину даних DO , а при $a_2=1$ слово читається із комірки, яка має найменший номер з тих, що відмічені 1 в R_{RPI} .

При запису спочатку розшукується вільна комірка. Для цього здійснюють контроль асоціації при $R_{GAO}=111\dots 10$ і $R_{GM}=000\dots 01$. Для запису вибирається вільна комірка з найменшим номером, куди заноситься слово, яке поступило з DI в R_{GI} .

За допомогою операції контролю асоціації можна, не читаючи слів із пам'яті, визначити по вмісту R_{RPI} кількість слів, які задовольняють асоціативну ознаку. При використанні відповідних комбінаційних схем в АЗП можна ефективно виконувати не тільки операції пошуку, а й досить складні логічні операції [6, 11, 19]. В сучасних ЕОМ АЗП широко використовуються для реалізації кеш-пам'яті.

Організація віртуальної пам'яті

Автоматичне планування передач інформації в багаторівневі пам'яті базується на побудові віртуальної (фіктивної, уявної) однорівневої пам'яті. Віртуальна пам'ять являє собою єдиний адресний простір, в якому фізична обмеженість ємності ОП схована від програміста. Таким чином, для програміста створюється видимість довільної адресації з відсутністю обмежень на ємність пам'яті, що значно полегшує програмування. Окрім цього, використання віртуальної організації пам'яті сприяє підвищенню взаємозамінюваності програм між обчислювальними системами.

Пам'ять, що існує реально, називають фізичною, а її адреси – фізичними, логічну пам'ять – віртуальною, а її адреси – віртуальними (логічними). Відповідність між фізичними і віртуальними адресами встановлюється сумісно апаратними засобами ЕОМ і її операційною системою. Звичайно віртуальний адресний простір розміщується в

зовнішній пам'яті, наприклад, на магнітних дисках. Частина цього простору, необхідна для виконання програм в даний момент, копіюється в ОП.

Для реалізації віртуальної пам'яті необхідно розділити весь адресний простір пам'яті на частини і організувати відповідний обмін між основною і зовнішньою пам'яттю. При цьому адресний простір пам'яті розбивається на сторінки і сегменти.

Сторінками називають частини віртуального і реального адресного простору фіксованої довжини. Адресі кожної сторінки віртуального простору ставиться в відповідність адреса сторінки фізичного адресного простору. Взаємозв'язок між адресами обох типів встановлюється таблицею перетворення адрес, яку називають таблицею сторінок (рис. 2.6). Перенесення сторінки віртуального простору в основну пам'ять називають завантаженням сторінки (довантаження в ОП), а обернену дію – видаленням сторінки (розвантаження з ОП). Обмін сторінками інакше називають свопінгом.

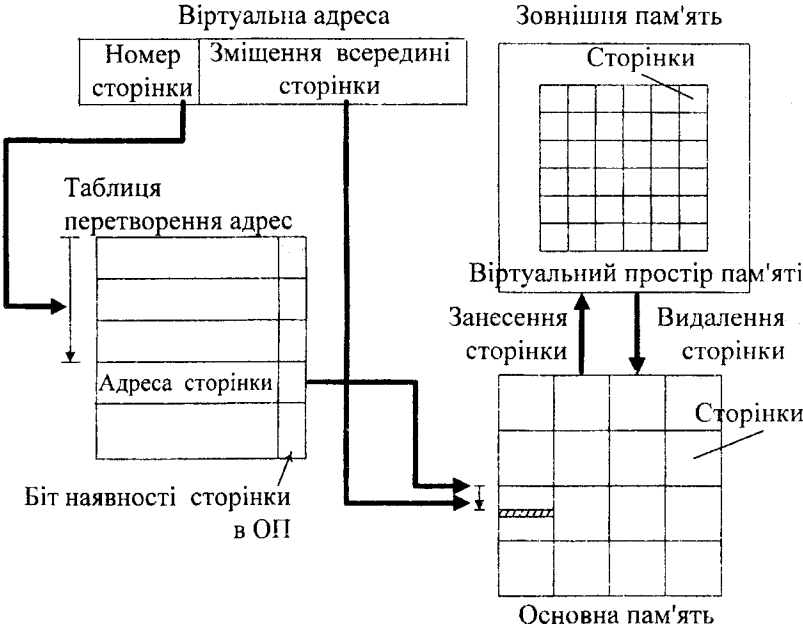


Рис. 2.6. Сторінкова організація пам'яті

Сегментацією називають розподіл адресного простору пам'яті на частини (сегменти) за логічними ознаками, які встановлюються програмістом. Звичайно сегменти відповідають програмі або підпрограмі і мають, на відміну від сторінки, змінну довжину. Віртуальна адреса в цьому випадку складається з номера сегмента і відносної адреси в межах сегмента. Вона перетворюється в фізичну адресу за таблицею сегментів.

Оскільки сегмент є логічною одиницею, можна організувати захист інформації і керування для колективного використання сегментованої інформації. Для кращого керування сегментованою пам'яттю сегменти додатково розбивають на сторінки.

Розширення віртуального простору приводить до збільшення таблиці сторінок. Цього можна уникнути при розподілі віртуального простору на два рівня – сегментів і сторінок, а перетворення віртуальної адреси здійснюється за дворівневою таблицею. Це дає можливість обійтись без ведення таблиці сторінок, що не використовуються, і економить об'єм пам'яті, який виділяється для таблиці сторінок.

Мультипрограмування орієнтовано на паралельну обробку декількох задач і реалізується через віртуальну пам'ять такими двома методами. Перший метод оснований на розподілі віртуального простору між декількома задачами. За другим методом для кожної задачі створюється окремий віртуальний простір адрес. Таку пам'ять називають мультиплексною віртуальною пам'яттю.

Для керування мультиплексним віртуальним адресним простором організується декілька таблиць перетворення адрес, які переключаються при переході від задачі до задачі. Для цього в один з реєстрів пристрою керування процесора вводиться вказівник, за яким вибирається відповідна таблиця перетворення адрес. Переваги цієї системи полягають в тому, що простір пам'яті, який використовується кожною задачею, повністю заповнює рамки віртуального простору. Одночасно забезпечується високоефективний захист пам'яті завдяки тому, що ніяка задача не може сформулювати адресу, яка відноситься до іншої задачі.

Для реалізації віртуальної адресації використовується механізм динамічного перетворення адрес, процедура заміни сторінок та керування розподілом основної пам'яті.

Механізм динамічного перетворення адрес розроблений для прискорення процедури визначення фізичної адреси, оскільки використання таблиці сторінок для цього займає відносно багато часу. Суть цього механізму полягає в тому, що в асоціативній пам'яті попередньо записуються номери сторінок, які найбільш часто використовуються в даний час, і номери блоків, які відповідають цим сторінкам в основній пам'яті. В ході перетворення адрес в першу чергу перевіряється ця асоціативна пам'ять, і якщо вона містить відомості про необхідні сторінки, ці відомості можуть бути зразу ж використані, тобто скорочується три-валість процедури перетворення адрес (рис. 2.7). Такий асоціативний буфер називають буфером динамічної трансляції адрес віртуальної пам'яті TLB.

Процедура заміни сторінок полягає в тому, що коли потрібна сторінка відсутня в основній пам'яті, вона переписується сюди з

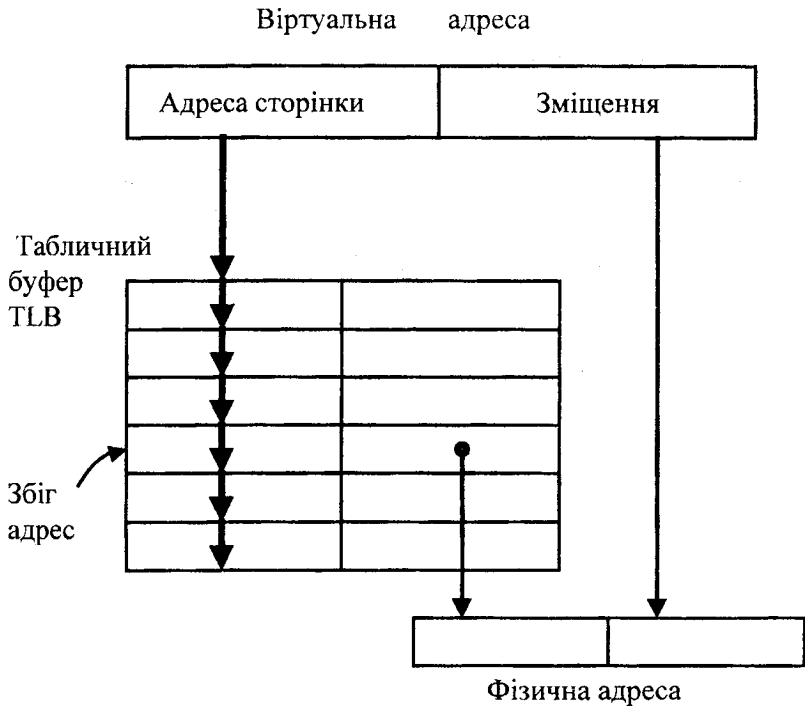


Рис. 2.7. Механізм динамічного перетворення адрес

зовнішньої пам'яті, і якщо в основній пам'яті відсутній вільний блок для завантаження сторінки, необхідно видалити одну з сторінок з основної пам'яті в зовнішню. Відомі такі стратегії видалення сторінок:

- стратегія FIFO, в відповідності з якою із основної пам'яті видаляється сторінка, яка була занесена раніше інших;
- стратегія LRU - видаляється та сторінка, останнє звертання до якої мало місце раніш, ніж до інших;
- стратегія WS (Working Set – робоча множина) – видаляються сторінки, які не містяться в так званій робочій множині, тобто наборі сторінок, до яких за певний інтервал часу зафіксовано звертання;
- стратегія випадкового заміщення - із основної пам'яті видаляється будь-яка сторінка.

Стратегії LPU і WS основані на припущенні, що сторінки, які використовуються останнім часом, будуть використовуватись і в майбутньому. Порівняно з ними стратегія FIFO простіша, але ефективність її відносно нижча. На практиці звичайно використовуються стратегії LPU, WS і їх модифікації.

Керування розподілом основної пам'яті між програмами, що беруть участь у паралельній обробці, полягає в перерозподілі блоків ОП для забезпечення кожної з програм, які паралельно виконуються, певним об'ємом основної пам'яті. Чим більший об'єм ОП, тим рідше потрібно виконувати її перерозподіл. При надмірному збільшенні рівня мультиплексування програм зменшується об'єм пам'яті кожної з них і підвищується частота обміну сторінками, що призводить до зниження ефективності всієї системи мультипрограмною обробки. Виникає так зване подрібнення пам'яті. Для усунення цього явища керівництво заміною сторінок здійснюється з врахуванням забезпечення найвищої ефективності використання центрального процесора. Хоч реалізація цього керування займає час, в цілому ситуація поліпшується завдяки своєчасному скороченню степені мультиплексування.

Метод віртуального представлення може бути розповсюджений не тільки на віртуальну пам'ять обчислювальної системи, але й на системи введення-виведення і архітектуру самих ЕОМ. Узагальнюючи, можна сказати, що віртуальне представлення використовується, коли в процесі логічного аналізу об'єктів не беруться до уваги технічні особливості їх побудови.

Організація кеш-пам'яті ЕОМ

Концепція кеш-пам'яті знайшла практичне використання завдяки двом аспектам: програмному і економічному. Програмний аспект полягає в тому, що більшість стандартних програм користувача створюється в відповідності з "принципом районування", який означає, що в будь-який момент часу частина програми, що адресується, розміщується в деякій обмеженій частині свого адресного простору. Економічний аспект обумовлений відсутністю більш дешевої порівняно з кешем технології, яка б дозволила досягти аналогічних експлуатаційних характеристик. Кеш забезпечує обмін невеликої кількості інформації між основною пам'яттю і процесором з високою швидкістю і тим самим складає враження, що велика основна пам'ять працює зі швидкістю кеша. До тих пір, поки елементна база повільної пам'яті буде значно дешевша, ніж швидкої, кеш-пам'ять буде залишатись економічно ефективною.

Кеш-пам'ять по суті складається з невеликого швидкодіючого ЗП, який вміщує частину інформації ОП. Такою інформацією можуть бути адреси, команди або дані. В відповідності з цим кеш може бути адресним, програмним або кешем даних. Швидкодія цього невеликого ЗП звичайно на порядок вища, а ємність на один-два порядки нижча, ніж у ОП.

В системі з кеш-пам'яттю потрібен ідентифікатор, або "ознакова пам'ять", для того, щоб визначити, які частини інформації ОП були скопійовані в кеш-пам'ять. В залежності від організації обміну ці частини можуть бути словами, блоками або сторінками. Спеціальну пам'ять, яка є асоціативною і вміщує імена частин або адреси, найчастіше називають ознаковою пам'яттю (пам'ять тегів – tag RAM), або довідником.

В кеш-пам'яті повинна бути логічна схема, яка визначає порядок видалення слів або блоків із кеша при необхідності занесення в нього нової інформації із ОП. Її називають пріоритетним списком оновлення даних.

Для синхронізації різних дій, наприклад, звертання в ознакову пам'ять, вибору і переміщення даних, в систему кеш-пам'яті вводиться пристрій керування.

Алгоритми свопінгу, які визначають порядок і спосіб пересилання даних між кеш-пам'яттю і ОП, повинні забезпечувати або отримання інформації за вимогою (правило вибірки за запитами), або попередній розподіл інформації (правило вибірки з упередженням). Стосовно до різних систем пам'яті з буфером найпоширенішими є такі типи алгоритмів свопінгу:

- алгоритм наскрізного запису WT (Write Through) або наскрізного накопичення ST;
- алгоритм простого свопінгу SS (Simple Swap) або оберненого запису в конфліктних ситуаціях CUW (Conflicting Usage Writeback);
- алгоритм свопінгу з ознаками FS (Flagged Swap) або оберненого запису в конфліктних ситуаціях з ознаками CUX (Flagged Conflicting Writeback);
- алгоритм реєстрового свопінгу з ознаками FRS (Flagged Register Swap).

Алгоритм наскрізного запису WT є найпростішим. В відповідності до нього кожен раз при появі запиту на запис за деякою адресою оновлюється вміст області за цією адресою як в швидкій, так і в основній пам'яті, навіть якщо копія вмісту за цією адресою знаходиться в буфері. Таке постійне оновлення вмісту основної пам'яті, як і буфера, при кожному запиті на запис дозволяє постійно підтримувати інформацію в ОП в оновленому стані. Хоч цей алгоритм і простий, але в ньому не закладена тенденція до мінімізації кількості звертань до ОП.

Алгоритм простого свопінгу SS полягає в тому, що звертання до ОП мають місце в тих, і тільки в тих випадках, коли в швидкому буфері не знаходиться потрібне слово. Ця схема свопінгу підвищує продуктивність системи пам'яті за рахунок того, що в ній звертання до ОП не здійснюються при кожному запиті на запис, що має місце при використанні алгоритму наскрізного запису. Однак в зв'язку з тим, що вміст ОП не підтримується в постійно оновленому стані, якщо необхідного слова в швидкому буфері нема, із буфера в ОП потрібно передати застарілі дані для звільнення місця для нових. Таким чином здійснюється дві пересилки між швидким буфером і ОП. При використанні алгоритму WT дані із буфера в ОП не передаються. Суттєвою перевагою алгоритму SS перед алгоритмом WT є те, що частина звертань до ОП в загальній кількості звертань асимптотично наближується до частини звертань до пам'яті, при якій змінюється склад кеша, а не до частини звертань, коли здійснюється запис в пам'ять. Окрім цього, якщо

має місце сумісність дій, центральний процесор може починати виконання операцій зразу ж після отримання нового слова.

Алгоритм свопінгу з ознаками FS передбачає супроводження кожного слова буфера додатковим бітом, значення якого змінюється при зміні слова буфера відносно того моменту, як воно попало в буфер із ОП. Згідно з цим алгоритмом в ОП передаються тільки ті слова, в яких додатковий біт знаходиться в установленому стані. Це дає змогу ще більше знизити кількість звертань до ОП.

Алгоритм регістрового свопінгу з ознаками FRS передбачає використання регістра (регістрів) тимчасового зберігання між буфером і ОП. Тепер, якщо дані повинні бути передані із швидкого буфера в ОП, вони спочатку пересилаються в регістр (регістри) тимчасового зберігання. Нове слово зразу ж пересилається в буфер із ОП, а вже потім слово, яке тимчасово зберігалось в регістрі, записується в ОП. Це забезпечує ще більше підвищення продуктивності порівняно з алгоритмом FS.

В повністю асоціативній кеш-пам'яті FACM (Fully Associated Cache Memory) кожна комірка зберігає дані, а в полі "тег" – повну фізичну адресу інформації, копія якої записана. Фізична адреса потрібної інформації порівнюється з полями "тег" всіх комірок. За результатами порівняння встановлюється сигнал Hit (Hit=1 - кеш-попадання; Hit=0 - кеш-промах). При читанні, коли Hit=1, дані видаються на шину даних, а коли Hit=0 – інформація читається із основної пам'яті і дані разом з адресою розміщуються також в вільній комірці кеш-пам'яті, або в тій, що давно не використовувалась. При запису дані разом з адресою спочатку, як правило, розміщуються в кеш-пам'яті, а копіювання їх в основну пам'ять виконуються тоді, коли нема звертань до пам'яті. Пам'ять типу FACM є досить складним пристроєм і використовується тільки при малих смностях, але вона забезпечує найбільшу функціональну гнучкість і безконфліктність адрес, тому що будь-яку одиницю інформації можна завантажити в будь-яку комірку кеш-пам'яті.

В кеш-пам'яті з прямим розміщенням (з прямим відображенням) декілька сторінок основної пам'яті відповідають одному рядку кеша. В зв'язку з тим, що займати рядок в один і той же час може тільки одна сторінка, потрібна спеціальна ознака – тег. Адреса від процесора розділяється на три частини. Її молодші розряди (зміщення) визначають положення слова в рядку, середні дозволяють вибрати один із рядків кеш-пам'яті, а старші утворюють тег. За адресою рядка здійснюється зчитування. Поле адрес зчитаного рядка порівнюється з теговою адресою, і, якщо вони збіглися, формується сигнал Hit видачі інформації, а мультиплексуванням із рядка вибирається слово. При завантаженні із основної пам'яті замінюється весь рядок. Тег для пам'яті з прямим розміщенням значно скорочується за розрядністю. Такий кеш є економічним за апаратурними витратами, але він має обмеження на розміщення сторінок, що може не дозволити сформуванню в ньому

оптимальний набір сторінок.

Набірно-асоціативний кеш є проміжним за складністю і ефективністю варіантом між структурою FACM і з прямим розміщенням. В цьому варіанті декілька рядків кеша об'єднуються в набори, а середні розряди адреси визначають уже не один рядок, а набір. Кількість наборів невелика і кратна степені два. Сторінку основної пам'яті можна розмістити тільки в тому наборі, номер якого дорівнює адресі сторінки за модулем. Місце сторінки в наборі може бути довільним. Порівняння тегів зі старши-ми розрядами адреси здійснюється тільки для рядків, які входять в набір.

В сучасних мікропроцесорних системах кеш першого рівня L1 (від Level – внутрішній), який реалізований всередині процесора, звичайно має набірно-асоціативну структуру, а кеш другого рівня L2 (зовнішній) – структуру з прямим розміщенням.

Напівпровідникові запам'ятовувальні пристрої

Адресні напівпровідникові запам'ятовувальні пристрої поділяються на оперативні RAM (Random Access Memory) і постійні ROM (Read-Only Memory). Оперативні ЗП допускають зміну даних в довільний момент часу, а в постійних вміст або взагалі не змінюється, або змінюється рідко і в спеціальному режимі.

RAM поділяються на статичні і динамічні. В статичних ЗП SRAM (Static RAM) запам'ятовувальними елементами є тригери, а в динамічних DRAM (Dynamic RAM) дані зберігаються в вигляді зарядів конденсаторів, які утворені елементами МОН-структур. Саморозряд конденсаторів веде до руйнування даних, тому вони повинні періодично (кожних декілька мілісекунд) регенеруватися. В той же час щільність розміщення динамічних елементів пам'яті в декілька раз перевищує щільність розміщення елементів в статичних ЗП. Регенерація даних в динамічних ЗП здійснюється за допомогою спеціальних контролерів. Розроблені також ЗП, які мають внутрішню вмонтовану систему регенерації, у яких зовнішня поведінка відносно керуючих сигналів аналогічна поведінці статичних ЗП. Такі ЗП називають квазістатичними.

Статичні ЗП найчастіше будуються за структурою 2DM, але при невеликій ємності використовується структура 2D. Для їх реалізації використовуються різні технології: TTL(Ш), I²L, ЕПЛ, n-МОН, КМОН та інші. Останнім часом інтенсивно розвиваються статичні ЗП за технологією КМОН. Статичні ЗП в 4-5 раз дорожчі динамічних і приблизно в стільки ж раз менші за ємністю. Їх перевагою є висока швидкодія, а типова область використання – схеми кеш-пам'яті.

Особливістю динамічних ЗП є мультиплексування адресної шини, що приводить до зменшення кількості виводів мікросхеми пам'яті. Адреса розділяється на дві напівадреси, одна з яких являє собою адресу рядка, а інша – адресу стовпця матриці запам'ятовувальних елементів. Напівадреси

подаються на одні і ті ж виводи інтегральної мікросхеми пам'яті по черзі. Подача адреси рядка супроводжується відповідним стробом RAS (Row Address Strobe), а адреса стовпця - стробом CAS (Column Address Strobe). Динамічні ЗП характеризуються великою інформаційною ємністю і невисокою вартістю, тому вони використовуються при побудові ОП ЕОМ.

Останнім часом поширюються динамічні ЗП підвищеної швидкодії, в яких використовуються методи, основані на припущенні купчастості адрес при звертанні до пам'яті. Варіант FPM (Fast Page Mode) є ефективним, коли наступне звертання здійснюється до того ж рядка запам'ятовувальних елементів. В цьому випадку виконується тільки цикл сторінкового режиму, а не звичайний (повний) цикл.

Структури EDORAM (Extended Data Out RAM – пам'ять з розширеним виведенням даних) подібні до структур FPM і відрізняються від них модифікацією процесу виведення даних. В EDORAM дані в підсилювачах-регенераторах не знімаються по закінченню стробу CAS. При цьому на кристалі як би з'являється статичний регістр, де зберігається рядок. При звертаннях в межах цього рядка використовується читання даних з регістра.

В структурі типу BEDORAM (Burst EDORAM, тобто з пакетним розширеним доступом) є додатково лічильник адрес стовпців. При звертанні до групи слів (пакета) адреса стовпця формується звичайним способом тільки на початку пакетного циклу. Для подальших передач вона визначається за допомогою лічильника. Пам'ять BEDORAM не набула широкого розповсюдження завдяки появі сильного конкурента - синхронних DRAM (SDRAM).

В структурах MDRAM (Multibank DRAM – багатобанкові ОЗП) пам'ять поділяється на частини (банки). Звертання до банків здійснюється по черзі, що виключає очікування перезаряду шин. Поки зчитуються дані з одного банку, інші мають час на підготовку, після якої з'являється можливість звертання до них без додаткового очікування. Чим більше банків, тим менше повторних послідовних звертань в один і той же банк з повним циклом.

Пам'ять типу SDRAM (Synchronous DRAM) займає важливе місце в якості швидкодіючої пам'яті з високою пропускну здатністю. В SDRAM синхросигнали пам'яті тісно пов'язані з тактовою частотою системи, в них використовується конвезризація тракту проходження інформації, може використовуватись багатобанкова структура. Тут вхідні сигнали фіксуються фронтами тактових імпульсів, які жорстко задають моменти їх появи та зникнення. До переваг SDRAM відносять відсутність великих проблем по узгодженню взаємного положення в часі вхідних сигналів, що в деяких випадках може бути складним.

Структури типу RDRAM (Rambus DRAM) названі по імені фірми-розроблювача і являють собою байт-послідовну пам'ять з дуже високим темпом передачі байтів. Принципово новими в архітектурному плані є

синхронізація обома фронтами тактових імпульсів і спеціальний новий інтерфейс Rambus Channel, який має всього 13 сигнальних ліній, що значно менше ніж у традиційних мікросхем пам'яті. В інтерфейсі відсутні спеціалізовані адресні лінії. Замість звичайної адресації по інтерфейсу посилаються пакети, які включають команди і адреси. Спочатку посилається пакет запитів, на які пам'ять відповідає пакетом підтвердження, після чого іде пакет даних. Перший доступ до даних має велике запізнення. Тому при читанні окремих слів RDRAM неефективна. Вона ідеально підходить для графічних і мультимедійних застосувань з типовим для них процесом – швидкою видачею довгої послідовності слів для формування зображення на екрані або подібних задач.

В пам'яті DRDRAM (Direct RDRAM) усунений фактор запізнення при першому доступі до даних, що розширило область застосування DRDRAM. DRDRAM з 16-розрядним інтерфейсом при роботі на тактовій частоті 400МГц і схемотехніці DDR (Double Data Rate), яка передбачає тактування процесів обома фронтами імпульсів, забезпечують пропускну здатність всередині пакета 1,6 Гбайт/с.

В структурах CDRAM (Cached DRAM) на одному кристалі з DRAM розміщена статична кеш-пам'ять (кеш першого рівня). При цьому кеш забезпечує швидкий обмін з процесором, якщо інформація знаходиться в кеші, а також швидке оновлення свого вмісту за рахунок того, що розміщення кеша і DRAM на одному кристалі робить зв'язки між ними внутрішніми, тому розрядність шин може бути великою (наприклад 2048) і обмін може здійснюватися великими блоками даних. Як синонім позначення CDRAM іноді використовується позначення EDRAM (Enhanced DRAM).

Постійна пам'ять типу ROM програмується при виготовленні методами інтегральної технології за допомогою масок. Елементами зв'язку в маскових ЗП можуть бути діоди, біполярні та МОП-транзистори та інші елементи. Це дійсно постійна пам'ять, вміст якої споживач змінити не може.

Постійні ЗП, вміст яких може програмуватися користувачем, позначаються PROM (від Programmable). В ЗП типу PROM програмування здійснюється створенням або руйнуванням спеціальних перемичок. В початковій заготовці відсутні (або присутні) всі перемички. Після програмування виникають або залишаються тільки необхідні. Найбільш поширеними є елементи з плавкими перемичками і діодними парами. В PROM вміст може програмуватися однократно.

В EPROM (Electrically Programmable ROM) інформація може стиратися шляхом опромінення кристала ультрафіолетовими променями, а в EEPROM або E²PROM (Electrically Erasable Programmable ROM) – електричними сигналами. Програмування PROM і репрограмування EPROM та EEPROM проводиться в звичайних лабораторних умовах за допомогою або спеціальних програматорів, або спеціальних режимів без

спеціальних приладів (для EEPROM). Занесення даних в EPROM та EEPROM здійснюється електричними сигналами.

Флеш-пам'ять (Flash-Memory) подібна до EEPROM, але має свої особливості. Тут стирання інформації здійснюється або для всієї пам'яті одночасно, або для достатньо великих блоків. Вона є технологічною і має хороші параметри. Широко використовується як носій BIOS.

Енергонезалежність всіх ROM, які зберігають інформацію при відключенні живлення, відкривають можливість економії живлення при їх експлуатації і, відповідно, покращання теплового режиму, що підвищує надійність схем. В режимі імпульсного живлення за допомогою ключів, які керуються від виходів адресного дешифратора (старші розряди адреси), живлення підключається тільки до вибраної мікросхеми. Це дозволяє суттєво зменшити споживчу потужність, але і збільшує час звертання до ЗП при поодиноких довільних звертаннях завдяки тому, що після включення живлення потрібен час для встановлення режиму мікросхеми.

Конкретні характеристики типів пам'яті, які використовуються в сучасних комп'ютерах, і відповідних модулів пам'яті (SIP, SIMM, DIMM, RIMM) приведені в [4, 7, 18].

Один з можливих типових наборів сигналів адресних напівпровідникових ЗП (рис. 2.8) включає [18]:

- **A** – адреса, розрядність якої визначає кількість комірок ЗП, тобто максимальну кількість слів, які можуть тут зберігатися;
- **CS** (Chip Select) або **CE** (Chip Enable) – сигнал, який дозволяє або забороняє роботу даної мікросхеми;

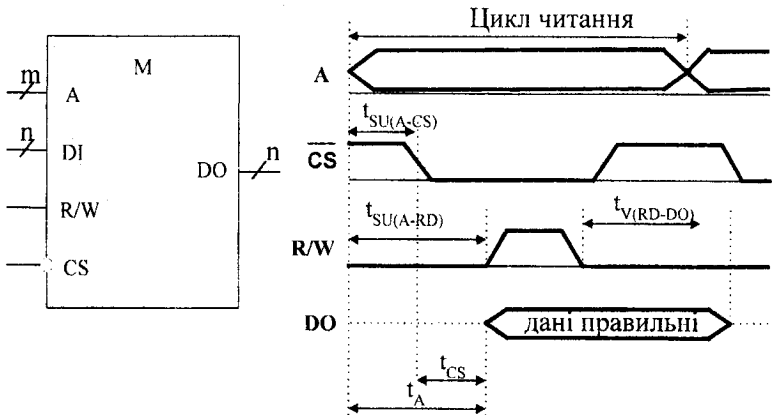


Рис. 2.8. Типові сигнали адресних напівпровідникових ЗП

- **R/W** (Read/Write) – задає тип операції: 1 – читання, 0 – занесення;
- **DI** і **DO** (Data Input, Data Output) шини вхідних і вихідних даних, в деяких ЗП ці лінії об'єднані.

Для ЗП характерна така послідовність сигналів. Перш за все подається адреса, щоб наступні операції не зачепили іншої комірки, окрім вибраної. Потім дозволяється робота мікросхеми сигналом CS і подається строб читання/запису R/W (взаємне розташування сигналів CS і R/W для різних ЗП може бути іншим). Якщо задана, наприклад, операція читання, то після подачі вказаних сигналів ЗП готує дані для читання, що вимагає певного часу. Задній фронт сигналу R/W, положення якого в часі повинне забезпечувати встановлення правильних даних на виході ЗП, зчитує дані. На рис. 2.8 прийняті такі позначення:

- t_A - час доступу відносно сигналу адреси (час вибірки);
- t_{CS} - час доступу відносно сигналу CS (час вибору);
- $t_{SU(A-CS)}$ - час попереднього встановлення сигналу A відносно сигналу CS;
- $t_{SU(A-RD)}$ - час попереднього встановлення сигналу A відносно сигналу R/W;
- $t_{V(RD-DO)}$ - час збереження.

Окрім вказаних параметрів використовується також параметр $t_{H(A-B)}$ - час утримання, який визначає інтервал між початком сигналу A і закінченням сигналу B.

Зовнішні ЗП

До ЗЗП відносяться пристрої, які дозволяють автономно зберігати інформацію незалежно від стану (ввімкнутий чи вимкнутий) комп'ютера. В ЗЗП використовуються методи прямого (дисківі накопичувачі) і послідовного (накопичувачі на магнітній стрічці) доступу до інформації. Характерною особливістю ЗЗП є те, що вони виконують операції над блоками інформації, а не окремими байтами або словами, як це дозволяє ОП. Ці блоки звичайно мають фіксований розмір, кратний степені двійки, але в деяких випадках у пристроїв з послідовним доступом розмір блока може бути змінним.

В дискових накопичувачах шар носія інформації - магнітний, оптичний або інший - нанесений на робочі поверхні дисків. Диски обертаються за допомогою двигуна шпинделя, який забезпечує потрібну частоту обертання в робочому режимі. На диску є індексний маркер, який, проходячи повз спеціальний датчик, відмічає початок кожного оберту диску. Інформація на диску розміщується на концентричних доріжках (треках), нумерація яких починається з зовнішнього треку. Кожен трек розбитий на сектори фіксованого розміру. Сектор і є мінімальним блоком інформації, яка може бути записана на диск, або прочитана з нього. Нумерація секторів починається з одиниці і прив'язується до індексного маркера. Кожен сектор має службову область, яка вміщує адресну інформацію, контрольні коди і деяку іншу інформацію, і область даних, розмір якої традиційно складає 512 байт. Якщо накопичувач має декілька робочих поверхонь, то сукупність всіх треків з однаковими номерами складає циліндр. Для кожної робочої поверхні в накопичувачі є своя

головка, яка забезпечує запис або читання інформації. Для того, щоб здійснити елементарну операцію обміну, шпindel повинен обертатися з заданою швидкістю, блок головок повинен бути підведений до потрібного циліндра, і тільки коли потрібний сектор підійде до вибраної головки, почнеться фізична операція обміну даними між головкою і електронним блоком накопичувача. Для запису інформації використовуються різні методи частотної або фазової модуляції, які дозволяють кодувати і декодувати двійкову інформацію в відповідності до природи носія інформації. Контролер накопичувача збирає і розбирає блоки інформації, включаючи формування і перевірку контрольних кодів, здійснює модуляцію і демодуляцію сигналів головок і керує всіма механізмами накопичувача.

Низькорівневе форматування дисків LLF (Low Level Formatting) – це процедура створення структури секторів диска, яка для кожного накопичувача повинна бути виконана перед його використанням. Процедура зводиться до того, що кожен трек диска розмічається і верифікується. При розмітці треку на ньому створюються заголовки секторів, а в поля даних записується який-небудь код-заповнювач. Відформатувати трек можна тільки цілком – за одну операцію повинні бути записані всі його сектори. В процесі верифікації робиться спроба зчитування кожного сектора, і якщо при зчитуванні виявляється стійка помилка, сектор в заголовку помічається як дефектний. Для гнучких дисків низькорівневе форматування є складовою частиною процесу форматування, який виконується утилітою FORMAT або подібними до неї. Під час форматування жорстких дисків формуються структура пакета дисків зі всіма чередуваннями, зміщеннями та іншими тонкощами. Для покращання звертань до файлів і впорядкування використання простору секторів диску в склад операційної системи входить і файлова система, яка визначає і логічну структуру диску.

Найбільш поширеними в даний час є накопичувачі на гнучких магнітних дисках FDD (Floppy Disk Drive) з дискетами діаметром 3,5 дюйма і форматованою ємністю 1,44 Мбайт, та накопичувачі на жорстких магнітних дисках HDD (Hard Disk Drive), ємність яких вже досягає десятків Гбайт.

Магнітооптичні диски (MOD - Magneto-Optical Drives) використовують оптику в процесі магнітного запису. В цих дисках суттєво зменшені розміри зон, які перемагнічуються, завдяки тому, що магнітна головка при запису діє тільки на мікроскопічну зону носія, що розігрівається лазерним променем. Такі диски стійкі до сильних зовнішніх магнітних полів.

В оптичних дисках CD-ROM (Compact Disk-Read Only Memory) для зберігання інформації використовується зміна оптичних властивостей (в

основному ступені відбивання) поверхні носія. В процесі зчитування при освітленні треку лазерним променем виникає модуляція інтенсивності відбитого променя, який сприймається фотоприймачем. Запис дисків може здійснюватись як друком з матриці, так і безпосереднім записом на носій в спеціальному пристрої. Оптичний диск замість концентричних треків має одну спіральну доріжку. Практично всі пристрої CD-ROM дозволяють відтворювати і аудіо-диски, для чого вони мають вмонтовані цифроаналогові перетворювачі і аналоговий інтерфейс з лінійним виходом стереосигналу рівня 0,2 В. Існує ряд стандартів на компакт-диски, які визначають їх фізичний формат.

Накопичувачі на магнітній стрічці (стрімери) є пристроями послідовного доступу. Носіями тут є касети зі стрічкою різного розміру і ємністю від 20 Мбайт до 2 і більше Гбайт. Найпростіші стрімери мають інтерфейс, сумісний з контролером гнучких дисків, і підключаються до звичайного шлейфу дисководів, а більш складні мають власну інтерфейсну карту або вмонтований контролер.

Принцип дії, конкретні параметри різновидів ЗЗП, формати їх носіїв, інтерфейсів, особливості роботи з ними детально розглянуті в [4, 7].

Контрольні запитання і задачі

1. Приведіть і охарактеризуйте основні параметри ЗП.
2. Дайте характеристики основним типам ЗП.
3. Приведіть структурну схему і охарактеризуйте ЗП типу 2D, 3D, 2D-M.
4. В чому полягають особливості побудови і функціонування стекових ЗП?
5. Дайте характеристики асоціативним ЗП. Що таке контроль асоціації?
6. Яким чином в асоціативну пам'ять записується інформація?
7. Що таке віртуальна пам'ять?
8. Як здійснюється обмін сторінками в віртуальній пам'яті?
9. Охарактеризуйте методи визначення фізичної адреси в віртуальній пам'яті.
10. Що таке сторінка і сегмент, в чому полягає різниця між ними?
11. Що таке кеш-пам'ять? Які алгоритми свопінгу для кеш-пам'яті Ви знаєте? Охарактеризуйте ці алгоритми.
12. Назвіть основні типи напівпровідникових ЗП, охарактеризуйте їх властивості.
13. Приведіть характеристику ЗП типу FPM, EDORAM, BEDORAM, MDRAM, SDRAM, RDRAM, DRDRAM, PROM, EPROM, EEPROM.
14. Що таке Флеш-пам'ять? Які особливості вона має?
15. Охарактеризуйте типовий набір сигналів напівпровідникових ЗП.
16. Охарактеризуйте основні типи зовнішніх ЗП.

3. АРИФМЕТИКО-ЛОГІЧНІ ПРИСТРОЇ ЕОМ

Арифметико-логічні пристрої (АЛП) призначені для виконання арифметичних і логічних операцій над операндами. До арифметичних операцій відносять додавання, віднімання, множення і ділення, до логічних – диз'юнкцію, кон'юнкцію та порівняння кодів. Спеціальні арифметичні операції включають нормалізацію, арифметичний зсув (зсуваються тільки цифрові розряди), логічний зсув (зсуваються всі розряди разом зі знаковим). Розрізняють також операції над алфавітно-цифровими полями.

В залежності від способу дії над операндами АЛП поділяються на послідовні, паралельні і послідовно-паралельні. В послідовних АЛП операнди представлені в послідовних кодах, а операції виконуються послідовно в часі над їх окремими розрядами. В паралельних АЛП операнди представлені паралельним кодом і операції здійснюються паралельно в часі над всіма розрядами операндів. В послідовно-паралельних АЛП операнди розділяються на групи розрядів і обробляються в межах виділеної групи паралельним способом, а між групами – послідовним. Паралельні АЛП забезпечують високу швидкість, але вимагають і великих апаратних витрат на їх реалізацію, а послідовні – навпаки.

За формою подання чисел розрізняють АЛП для чисел з фіксованою комою (крапкою), з плаваючою комою та десяткових чисел. В залежності від прийнятої системи числення АЛП розділяють на двійкові, трійкові, десяткові, АЛП в СОК, в надлишкових системах числення, в системах зі штучним порядком ваг і т.п.

За характером використання елементів і вузлів АЛП поділяються на блокові та багатофункціональні. В блоковому АЛП кожна операція або група операцій виконуються окремими блоками, при цьому підвищується швидкість завдяки одночасному виконанні відповідних операцій в різних блоках, але значно виростають апаратні витрати. В багатофункціональних АЛП операції виконуються одними і тими ж схемами, котрі налаштовуються потрібним чином в залежності від режиму роботи.

За структурою розрізняють АЛП з закріпленими мікроопераціями (універсальні) та АЛП зі спільними мікроопераціями (магістральні). В АЛП з закріпленими мікроопераціями в залежності від необхідних елементарних дій над операндами (мікрооперацій) виділяються відповідні базові вузли (операційні елементи), встановлюються зв'язки між ними і включаються логічні схеми, які формують ознакові сигнали. В таких АЛП на основі принципу закріплення мікрооперацій за регістрами для кожної мікрооперації існує логічна схема і з'являється велика кількість ідентичних схем, пов'язаних з різними регістрами. В АЛП зі спільними мікроопераціями використовуються спільні схеми для виконання мікрооперацій над словами, які зберігаються в різних регістрах. Найчастіше такі АЛП поділяються на дві частини: комбінаційну і запам'ятовувальну. В середньому затрати часу на виконання операцій в

АЛП зі спільними мікроопераціями більші, ніж в АЛП з закріпленими мікроопераціями в зв'язку зі збільшенням тривалості такту (росте кількість рівнів в схемах виконання мікрооперацій) і збільшенням кількості тактів, що пов'язано з обмеженнями на сумісність мікрооперацій і наявність мікрокоманд, які служать для формування необхідних ознакових сигналів.

За своїми функціями АЛП є операційним блоком, який виконує мікрооперації, що забезпечують прийом операндів з інших пристроїв, їх перетворення і видачу результатів. АЛП керується пристроєм керування, який генерує послідовність керуючих сигналів в відповідності до алгоритму виконання операції і ознакових сигналів, які виробляються в АЛП.

Список операцій, які виконує АЛП, визначається класом алгоритмів, для виконання яких призначена ЕОМ. В універсальних ЕОМ використовується найбільш широкий список операцій, достатній для ефективної реалізації різноманітних алгоритмів. Спеціалізація ЕОМ звужує клас алгоритмів, в результаті чого деякі операції стають нетиповими, що дозволяє виключити їх зі списку машинних операцій. Операції в ЕОМ можуть бути реалізовані як апаратними, так і програмними засобами. Розширення списку операцій призводить до збільшення апаратних витрат. При наявності обмежень на кількість обладнання можна відказатись від апаратної реалізації найбільш складних операцій, які можуть бути реалізовані шляхом використання програмних засобів, які інтерпретують складні операції в термінах більш простих операцій. При цьому, звичайно, знижується продуктивність ЕОМ.

До основних параметрів АЛП відносять розрядність, повноту списку операцій, апаратні витрати на реалізацію, швидкодію.

Швидкодія операційного пристрою характеризується часом виконання операції, який визначається добуток тривалості такту операційного пристрою на кількість тактів, необхідних для реалізації алгоритму. Оскільки кількість тактів залежить від значень операндів, їм приписується випадковий характер, а значення кількості тактів визначається як середня кількість тактів, які витрачає пристрій на реалізацію алгоритму з операндами – випадковими величинами з заданими законами розподілу.

Мікрооперації і мікропрограми

За формою запису мікрооперація – це оператор присвоювання, за допомогою якого слову присвоюється значення двійкового виразу. Запис найбільш поширених мікрооперацій розглянемо на прикладах:

• мікрооперація установки

$A:=0$ – слову А приймає нульове значення;

$V:=C$ – слову V присвоюється значення слова C;

$A[1]:=0$ – перший розряд слова А приймає нульове значення;

$V[1:8]:=A[17:24]$ – розряди 1-8 слова V відповідно приймають значення розрядів 17-24 слова A;

- **мікрооперація інвертування**

$A := \bar{A}$, або $A := \neg A$, або $A := \bar{N}A$ – зміна значення слова A на інверсне;

$C[1] := \bar{C}[1]$ – зміна значення першого розряду слова C на інверсне;

- **мікрооперація зсуву**

$A := 0.R1(A)$ – зсув слова A вправо на один розряд з прийомом нульового значення в крайній зліва розряд;

$A := L1(A).x1$ – зсув слова A вліво на один розряд з прийомом значення $x1$ в крайній справа розряд;

$A := x1x2.R2(A)$ – зсув слова A вправо на два розряди з прийомом значень $x1x2$ в крайні зліва розряди;

- **мікрооперація лічби**

$A := A+1$ – збільшення значення слова A на одиницю;

$A := A-1$ – зменшення значення слова A на одиницю;

- **мікрооперація додавання (підсумовування)**

$C := A+B$ – слово C приймає значення суми кодів A і B ;

- **бінарні логічні мікрооперації**

$C := A \wedge B$ – слово C приймає значення порозрядної кон'юнкції кодів A і B ;

$C := A \vee B$ – слово C приймає значення порозрядної диз'юнкції кодів A і B ;

- **комбіновані мікрооперації (включають декілька мікрооперацій різних класів)**

$A := L1(\bar{A}).1$ – зсув вліво з одночасним інвертуванням слова A .

Мікрооперації реалізуються базовими вузлами ЕОМ. Деякі із мікрооперацій, що використовуються в мікропрограмі, можуть виконуватись одночасно, а інші – тільки послідовно. Властивість сукупності мікрооперацій, яка гарантується можливістю їх паралельного виконання, називається сумісністю. Мікрооперації, які не мають такої властивості, називаються несумісними.

Сумісність мікрооперацій обумовлена, по-перше, змістом операторів, які представляють мікрооперації – так звана функціональна сумісність – і, по-друге, структурою операційного пристрою, яка допускає або виключає можливість паралельного виконання декількох мікрооперацій, – так звана структурна сумісність. В функціональних мікропрограмах, які описують алгоритми виконання операцій безвідносно до структури пристрою, паралельно можуть виконуватись тільки ті мікрооперації, які мають властивість функціональної сумісності. Структура пристрою може внести обмеження на кількість мікрооперацій, які виконуються паралельно. Структурна несумісність мікрооперацій пов'язана з використанням мікроопераціями спільного обладнання, що виключає можливість сумісного виконання мікрооперацій. Таким чином, якщо структура операційного пристрою не визначена, то сумісними називаються функціонально сумісні мікрооперації. Якщо структура задана, то сумісними називаються структурно сумісні мікрооперації.

Логічна умова – це бульова функція, де в якості булевих виразів використовуються однорозрядні слова, поля і відношення, наприклад:

$A[0]$ – однорозрядне поле, значенням 0 і 1 якого відповідають такі ж значення логічної умови;

$A[0] \vee A[1]$ – умова, приймає значення 0, якщо змінні $A[0]$, $A[1]$ одночасно дорівнюють нулю і 1 в інших випадках;

$A=0$ – умова, представлена відношенням рівності, приймає значення 1 при $A=0$ і 0 в протилежному випадку;

$C \leq 127$ – умова, представлена відношенням "не більше", приймає значення 1, коли $C \leq 127$ і 0 в протилежному випадку.

Для інженерних застосувань алгоритми виконання операцій в пристрої зручно описувати змістовною граф-схемою алгоритму (ГСА) (інша назва – граф мікропрограми). При побудові ГСА використовуються вершини чотирьох типів (початкова, кінцева, операційна, умовна) і дуг, які їх пов'язують.

ГСА є коректною, якщо виконуються умови:

- 1) ГСА включає тільки одну початкову і одну кінцеву вершини;
- 2) в будь-яку вершину, крім початкової, повинна входити хоча б одна дуга, яка виходить з іншої вершини;
- 3) з кожного виходу будь-якої вершини, крім кінцевої, повинна виходити одна дуга яка веде до деякої вершини графа;
- 4) при всіх можливих значеннях слів повинен існувати шлях з початкової вершини до кінцевої.

АЛП і алгоритми для виконання арифметичних і логічних операцій

При побудові АЛП з закріпленими мікроопераціями спочатку будують узагальнений алгоритм виконання операції, на основі його аналізу і аналізу критеріїв проектування вибирають базові вузли, визначають необхідні мікрооперації, потім з врахуванням вибраних мікрооперацій будують змістовну ГСА і в кінці проводять аналіз роботи пристрою з визначенням основних параметрів. Розглянемо приклади побудови операційних пристроїв з закріпленими мікроопераціями і ГСА для реалізації основних арифметичних операцій над числами з фіксованою комою, а також особливості реалізації пристроїв для обробки чисел з плаваючою комою і пристроїв для реалізації логічних операцій.

Операція додавання чисел зі знаками найбільш часто зустрічається в програмах. При її реалізації для подання від'ємних чисел використовується обернений або доповняльний код. Для додатних чисел прямий, обернений та доповняльний коди однакові. Щоб отримати обернений код від'ємного числа, потрібно проінвертувати значення цифрових розрядів (окрім знакового). Доповняльний код від'ємного числа можна отримати шляхом додавання одиниці молодшого розряду до оберненого коду. Для контролю за переповненням розрядної сітки використовується модифікований код, у якому для знака виділяється два двійкових розряди.

Узагальнений алгоритм операції додавання чисел з фіксованою комою, з використанням доповняльного коду можна сформулювати так:

- 1) доданки вступають в операцію в доповняльному модифікованому коді;
- 2) здійснюється додавання кодів доданків по всіх розрядах, включаючи знакові;
- 3) якщо значення знакових розрядів суми 00, то сума додатна і представлена в прямому коді, якщо ж ці значення 11, сума від'ємна і представлена в доповняльному коді; при значеннях знакових розрядів 01 і 10 сума переповнює розрядну сітку.

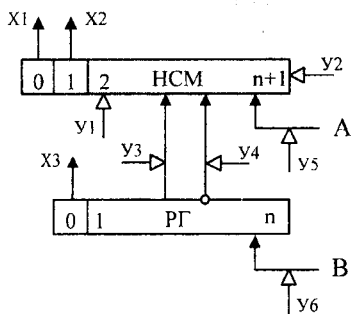
При використанні в операційному пристрої накопичувального суматора в нього можна попередньо занести один з доданків. Для зберігання іншого доданку потрібно виділити регістр. Якщо числа містять n цифрових розрядів, в суматорі потрібно передбачити $n+2$ розряди (модифікований код), а в регістрі достатньо $(n+1)$ -го розряду. Щоб отримати доповняльний код доданку, який знаходиться в суматорі, можна для суматора передбачити мікрооперацію інвертування цифрових розрядів, а додавання одиниці молодшого розряду можна здійснити, використовуючи вхід перенесення суматора. А щоб отримати доповняльний код доданка, який знаходиться в регістрі, можна в процесі додавання мультиплексувати з регістра на суматор обернений код, а на вхід перенесення суматора давати 1. В пристрої також потрібно передбачити мікрооперації занесення кодів. Структурна схема операційного пристрою, перелік мікрооперацій та логічних умов, а також ГСА додавання чисел з фіксованою комою приведені на рис. 3.1. Тут мається на увазі, що операнди і результат представлені в прямих кодах.

Операція віднімання чисел зі знаками може бути зведена до операції додавання з попереднім інвертуванням знака другого операнда.

Множення двійкових чисел з фіксованою комою зводиться до обчислення добутку модулів співмножників і присвоюванню добутку знака плюс, якщо знаки співмножників однакові, і мінус в протилежному випадку. В залежності від способу формування суми часткових добутків розрізняють чотири основних методи виконання операції множення, які розрізняються в залежності від того, з яких розрядів (старших чи молодших) починається аналіз множника, і що зсувається в процесі накопичення суми часткових добутків (множник або сума часткових добутків). Найбільше поширення отримав метод множення, починаючи з молодших розрядів множника зі зсувом суми часткових добутків вправо (в бік молодших розрядів), оскільки він вимагає суматора одинарної довжини, а решта методів – суматора подвійної довжини. За цим методом узагальнений алгоритм операції множення чисел з фіксованою комою можна сформулювати так:

- 1) беруться абсолютні значення модулів співмножників;
- 2) початкове значення суми часткових добутків приймається рівним 0;
- 3) аналізується чергова цифра множника: якщо вона дорівнює 1, то до суми часткових добутків додається множене, якщо ж ця цифра дорівнює 0, множене не додається;
- 4) сума часткових добутків зсувається вправо на один розряд;
- 5) пункти 3 і 4 послідовно виконуються для всіх цифрових розрядів

- множника, починаючи з молодшого;
 б) добутку присвоюється знак плюс, якщо знаки співмножників однакові і мінус в протилежному випадку.



$Y1 := HCM[2n+1] := HCM[2n+1]$
 $Y2 := HCM := HCM + 1$
 $Y3 := HCM + PG$
 $Y4 := HCM + \overline{PG}$
 $Y5 := HCM := A$
 $Y6 := PG := B$

$X1 := HCM[0]$
 $X2 := HCM[1]$
 $X3 := PG[0]$

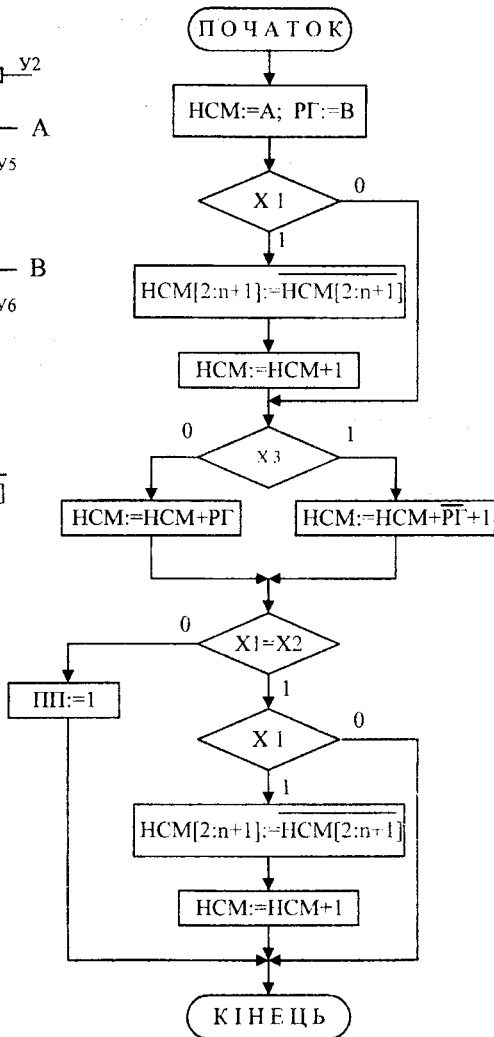


Рис. 3.1. Структурна схема ОП і ГСА операції додавання чисел з фіксованою комою

Враховуючи узагальнений алгоритм, в операційному пристрої для множення чисел з фіксованою комою слід передбачити два регістри для зберігання операндів, накопичувальний суматор для формування суми часткових добутоків та лічильник для визначення моменту закінчення циклу накопичення суми часткових добутоків. В регістрі множника та в накопичувальному суматорі слід передбачити мікрооперацію зсуву вправо на один розряд відповідно для зсуву множника з метою аналізу його чергового розряду та зсуву суми часткових добутоків. При зсуві суми часткових добутоків її молодші розряди послідовно можна розміщувати в тих розрядах регістра множника, які вивільнюються в процесі зсуву множника. Структурна схема операційного пристрою, перелік мікрооперацій та логічних умов, а також ГСА множення чисел з фіксованою комою приведені на рис. 3.2.

Операція множення відноситься до класу так званих “довгих” операцій, тому що виконується протягом великої кількості тактів в зв'язку з наявністю циклу в алгоритмі. Для прискорення її виконання використовують алгоритмічні (логічні), апаратні та табличні методи [6, 14, 16].

Серед алгоритмічних методів найбільше поширення мають методи, які дозволяють за один такт обробити декілька розрядів множника (з одночасним аналізом 2-х, 3-х, 4-х) розрядів множника. За одним з методів множення з одночасним аналізом двох розрядів множника, починаючи з його молодших розрядів, в залежності від результату аналізу пари розрядів множника передбачуються такі дії. При 00 сума часткових добутоків зсувається на два розряди вправо. При 01 і 10 до суми часткових добутоків додається відповідно множене або подвоєне множене і сума часткових добутоків зсувається на два розряди вправо. При 11 із суми часткових добутоків віднімається множене і сума часткових добутоків зсувається на два розряди вправо. В останньому випадку результат неправильний, він повинен бути відкоригований при наступному крокові. В зв'язку з тим, що при 11 із суми часткових добутоків віднімається множене замість додавання потроєного множеного, для коригування результату до суми часткових добутоків перед виконанням зсуву потрібно було б додати збільшене в чотири рази множене. Але після зсуву на два розряди вправо сума часткових добутоків зменшується в чотири рази, так що для коригування його на наступному крокові повинно бути додано одинарне множене. Це враховується при обробці наступної пари розрядів. Якщо наступна пара 00, 01, 10, 11 то вона обробляється відповідно як 01,10, 11, 00. В останньому випадку фіксується необхідність корекції при обробці наступної пари. Подвоєне множене може бути отримано його зсувом, а ознаку необхідності корекції можна запам'ятовувати в окремому тригері. Даний метод множення вимагає корекції результату, якщо старша пара розрядів множника 11 або 10 і стан тригера корекції одиничний.

До апаратних методів прискорення виконання операції множення відносяться методи, основані на прискоренні виконання

операцій додавання і зсуву, введенні додаткових схем зсуву, які дозволяють за один такт здійснювати зсув на декілька розрядів, організації зміщення в часі операцій додавання і зсуву, використанні "дерева" суматорів та "матриці" суматорів, побудові комбінаційних схем пристроїв для множення. Так, наприклад, в складі мікропроцесорного набору ВІС серії 1802 є помножувачі восьмирозрядних, дванадцяти розрядних та

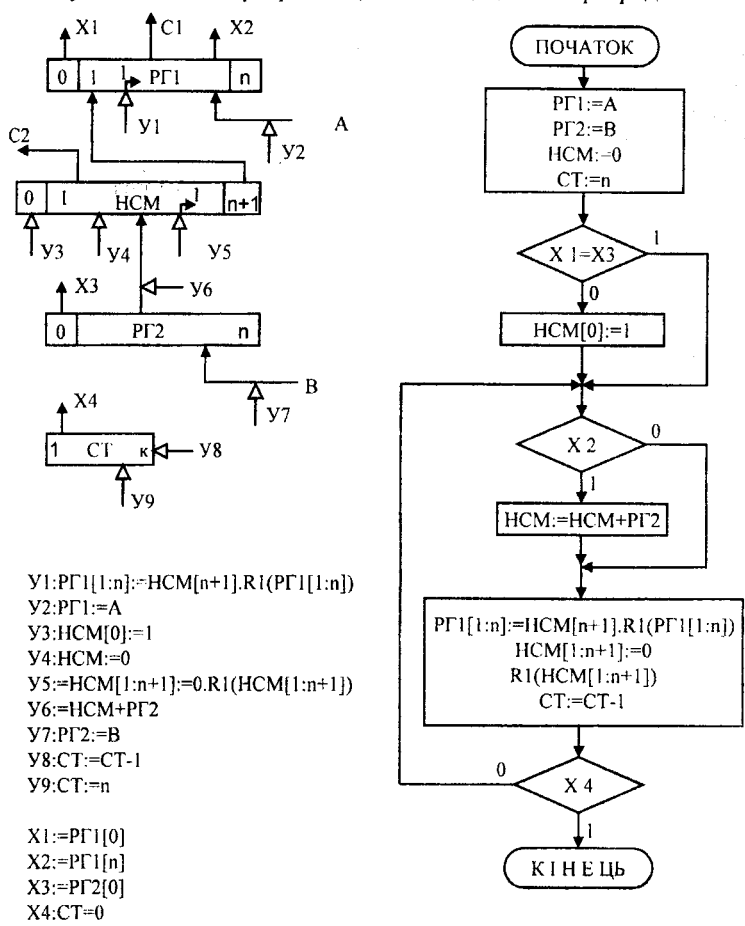


Рис. 3.2. Структурна схема ОП і ГСА операції множення чисел з фіксованою комою

шістнадцятирозрядних кодів або чисел в доповняльному коді з можливостями нарощування розрядності (КР1802ВР3, КР1802ВР4, КР1802ВР5).

Табличний метод множення може бути реалізований, наприклад, за допомогою постійного запам'ятовувального пристрою, в комірках якого записані добутки, а операнди утворюють адресу. При цьому множення зводиться до читання готового добутку з пам'яті. Однак використання цього методу в даний час припустимо тільки для операндів відносно невеликої розрядності.

Ділення двійкових чисел зводиться до обчислення частки модулів діленого та дільника і присвоювання частці знака плюс, якщо знаки операндів однакові і знака мінус – в протилежному випадку. Цифри частки визначаються послідовно, починаючи з цифри старшого розряду. Ділення дробів і цілих чисел здійснюється за різними алгоритмами.

Розрізняють алгоритми ділення за методами з відновленням та без відновлення остачі. Найбільш поширеним є алгоритм ділення дробів без відновлення остачі:

- 1) від діленого віднімається дільник і якщо результат додатний, то частка переповнює розрядну сітку;
- 2) остача подвоюється зсувом вліво;
- 3) якщо остача додатна, то дільник віднімається від остачі, а якщо від'ємна, то додається до остачі;
- 4) черговій цифрі частки присвоюється значення, інверсне по відношенню до знака остачі;
- 5) дії пунктів 2, 3, 4 повторюються до визначення необхідної кількості цифр частки.

В операційному пристрої для ділення дробів потрібно передбачити накопичувальний суматор, де розміщується спочатку ділене, а потім остачі, регістр для дільника та регістр для частки, а також лічильник для визначення моменту закінчення операції в залежності від необхідної кількості цифр частки. Структурна схема операційного пристрою, перелік мікрооперацій та логічних умов, а також ГСА операції ділення дробів приведені на рис. 3.3.

Для ділення цілих двійкових чисел може використовуватись такий алгоритм:

- 1) якщо дільник дорівнює нулю, то ділення неможливе;
- 2) ділене нормалізується; кількість зсувів, які виконуються при цьому, збільшується на одиницю і визначає p молодших цифр частки, які можливо не дорівнюють нулю, решта $(n-p)$ старших цифр частки дорівнюють нулю;
- 3) послідовно, починаючи від старших, визначаються значення p цифр частки шляхом виконання кроків 3, 4 і 5 раніше розглянутого алгоритму ділення дробів.

Операційний пристрій та ГСА алгоритму ділення цілих чисел пропонується побудувати самостійно. Методи підвищення швидкодії

операційних пристроїв для ділення чисел розглянуті в [16].

3 логічних операцій найчастіше в АЛП реалізуються диз'юнкція, кон'юнкція, сума за модулем два та порівняння кодів. Ці операції є порозрядними. Вони можуть виконуватись як над паралельними, так і над послідовними кодами. Часто для їх реалізації будують спеціальну комбінаційну схему одnobайтових логічних операцій, до якої послідовно підключають байти слів, що обробляються, і послідовно знімають байти результатів. Арифметичні операції над двійково-десятковими числами виконуються аналогічно операціям над цілими двійковими числами.

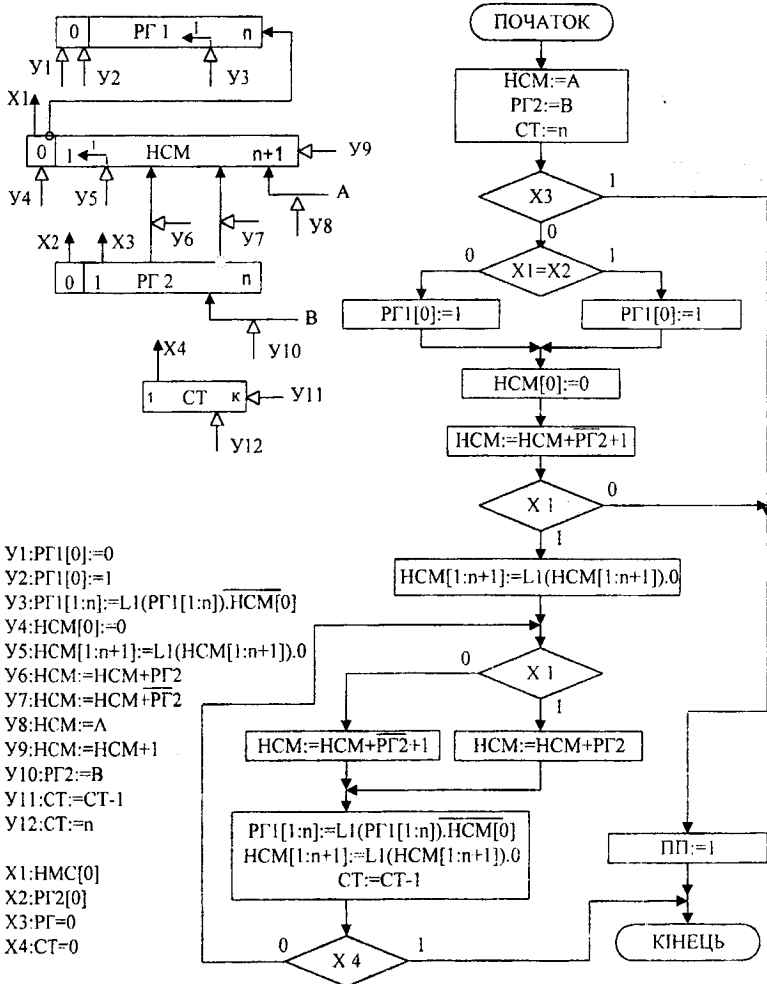


Рис. 3.3. Структурна схема ОП і ГСА операції ділення чисел з фіксованою комою

Основою АЛП десяткової арифметики є суматор двійково-десяткових кодів. Він може бути побудований на основі двійкового суматора з використанням спеціальної схеми корекції результату.

Множення десяткових чисел можна виконати методом послідовного додавання. В цьому випадку послідовно обробляються цифри множника, причому для кожного його розряду додавання виконується стільки раз, скільки одиниць міститься в даному розряді. Для керуванням кількістю додавань можна, наприклад, скористатись десятковим лічильником з початковим занесенням цифри множника і після кожного додавання зменшувати стан лічильника на 1 до нульового значення стану. Узагальнений алгоритм множення двійково-десяткових кодів має вигляд:

- 1) сума часткових добутків приймається рівною нулю;
- 2) аналізується чергова цифра (тетрада) множника, і множене додається до суми часткових добутків стільки раз, яке значення має цифра множника;
- 3) сума часткових добутків зсувається вправо на одну тетраду, і повторюються дії, вказані в п.2, для всіх цифр множника.

Для прискорення виконання цієї операції часто окремо формуються кратні множника $8X$, $4X$, $2X$, $1X$, при наявності яких зменшується кількість додавань при виконанні п.2.

Двійково-десяткове ділення виконується шляхом багатократних віднімань, подібно до того, як це робиться при звичайному діленні.

Двійково-десяткові АЛП часто будуються як послідовно-паралельні, що здійснюють послідовну обробку байт.

Арифметичні операції над числами з плаваючою комою є більш складними, ніж операції над числами з фіксованою комою. Число з плаваючою комою можна записати в вигляді:

$$A = M_A + q^P A$$

де M_A – мантиса числа A (дріб); P_A – порядок числа A (ціле); q – основа системи числення. Одне і те ж число може по різному бути представленим по формі з плаваючою комою. Число вважається нормалізованим, якщо старша цифра мантиси є значущою.

Додавання (віднімання) чисел з плаваючою комою передбачає вирівнювання порядків до більшого шляхом зсуву вправо мантиси меншого числа, додавання (віднімання) мантис і нормалізації результату. Додавання (віднімання) мантис здійснюється за правилами додавання (віднімання) чисел з фіксованою комою. При виконанні операції формуються ознаки переповнення та зникнення порядку. Операції додавання (віднімання) чисел з плаваючою комою виконуються наближено, оскільки при вирівнюванні порядків можуть бути втрачені молодші розряди одного з операндів.

При множенні чисел з плаваючою комою порядки співмножників додаються, а мантиси перемножуються. Добуток нормалізується, і йому присвоюється знак плюс, якщо співмножники мають однакові знаки, і знак мінус, якщо знаки співмножників різні. Якщо мантиса одного з співмножників дорівнює нулю, то добутку можна присвоїти нульове значення без виконання множення мантис. Якщо при додаванні порядків

виникло переповнення і порядок від'ємний, то добуток менший за мінімальне число, яке може бути представлено в машині, і результатом операції може бути записаний 0 без перемноження мантис. Якщо ж при додаванні порядків виникає переповнення і порядок додатний, може виявитись, що результат все-таки знаходиться в діапазоні чисел, які можуть бути представлені в машині, оскільки при множенні мантис можливе порушення нормалізації вправо, і після нормалізації мантиси переповнення в порядку може зникнути.

При діленні чисел з плаваючою комою мантиса частки дорівнює частці від ділення мантиси діленого на мантису дільника, а порядок частки – різниця порядків діленого і дільника. Частка нормалізується, і їй присвоюється знак плюс, якщо знаки операндів однакові, і знак мінус, якщо вони різні. Якщо ділене дорівнює 0, то в частку може бути записаний 0 без виконання ділення. Якщо ж при відніманні порядків має місце переповнення з додатним знаком, або дільник дорівнює нулю, формується сигнал переривання. При діленні нормалізованих чисел з плаваючою комою може статися, що мантиса дільника більше мантиси діленого, і мантиса частки переповнює розрядну сітку. Для усунення цього явища перед діленням мантис порушують нормалізацію дільника зсувом на розряд вліво.

В операційному пристрої для виконання арифметичних операцій над числами з плаваючою комою можуть бути передбачені вузли як для зберігання і обробки мантис, так і для зберігання і обробки порядків.

Проектування АЛП включає вибір кодів для представлення даних, визначення алгоритмів виконання окремих операцій, структур операційних блоків і наборів мікрооперацій, які реалізують ці блоки. Потім здійснюють з'єднання окремих операційних блоків і відповідних наборів мікрооперацій в один багатофункціональний операційний блок або декілька блоків для окремих груп операцій. В багатофункціональних АЛП операції над числами з фіксованою і плаваючою комою, десятковими числами і алфавітно-цифровими полями виконуються в основному одними і тими ж схемами.

Для мікропроцесорів типовою є така організація, при якій їх внутрішні регістри використовуються з різною метою. Система зв'язку у цих регістрів, як правило, централізована (магістральна). Вона забезпечує можливість різнобічних міжрегістрових пересилань, в тому числі і передач як в АЛП, так і з АЛП. В зв'язку з цим часто власні регістри АЛП, які забезпечують тільки виконання арифметичних і логічних операцій, в мікропроцесорах відсутні. Це дає привід розглядати АЛП мікропроцесорів як комбінаційну схему, яка виконує арифметичні і логічні операції над операндами, що знаходяться в регістрах мікропроцесора. Результат операції також розміщується в одному з регістрів мікропроцесора.

Основні методи контролю АЛП

При роботі АЛП і ЕОМ в цілому характерним є наявність відмов і збоїв, які приводять до помилок. Під відмовою розуміють таке порушення працездатності, для відновлення якого потрібні певні дії по ремонту, заміні і регулюванню несправного елемента, вузла або пристрою. Збій –

короткочасне порушення нормального функціонування системи завдяки короткочасній дії зовнішніх завад або змін параметрів її елементів, яке усувається саме по собі. Надійність обчислювальної машини визначається безвідмовністю, достовірністю функціонування і ремонтоздатністю. До основних параметрів надійності відносять середній час напрацювання на одну відмову, середній час усунення несправності, середній час напрацювання на один збій, середній час відновлення достовірності інформації після збою.

Метою методів контролю АЛП є виявлення помилок. Вони дозволяють вилучити вплив виявлених помилок на правильність результатів, які видаються з АЛП. По суті всіх методів контролю їм характерна наявність надлишковості – інформаційної, апаратної, часової. Методи контролю АЛП можна розділити на дві групи: тестові та апаратні.

Тест для перевірки АЛП – це випробувальна програма, за результатами виконання якої судять про працездатність системи в момент контролю. До тестів АЛП висуваються певні вимоги: перевірка максимальної кількості вузлів та схем АЛП в найбільш завантажених режимах, мінімум команд в тесті, циклічне повторення тесту, фіксація команд, при виконанні яких виявляються помилки. Найпростіша організація тесту для контролю АЛП може бути такою. Вибирається обмежена кількість операндів, над якими послідовно виконуються арифметичні і логічні операції з метою отримання правильних результатів. Ці операнди і правильно отримані результати є еталонами. Потім за спеціальною програмою над відповідними парами вибраних операндів послідовно виконуються всі операції, які реалізує АЛП, з порівнянням результатів кожної операції з еталонами. Звичайно, такі тести виявляють тільки частину неполадок і, як правило, є основою тестового контролю АЛП в режимі профілактики. Різновидом програмного контролю є програмно-логічний контроль, який використовує способи подвійного і потрійного прорахунку робочих задач з порівнянням отриманих результатів.

Для контролю передачі інформації найбільше розповсюдження мають методи інформаційної надлишковості, які використовують коди з виявленням і корекцією помилок. Код з перевіркою парності (контроль парності, непарності, паритету) утворюється, коли до групи інформаційних розрядів (найчастіше – байта) додається ще один надлишковий (контрольний) розряд. При формуванні коду слова в контрольний розряд записується 0 або 1 таким чином, щоб кількість одиниць в слові, включаючи контрольний розряд, була парною (при контролі на парність) або непарною (при контролі на непарність). В подальшому при всіх передачах слово передається разом з контрольним розрядом. Якщо приймальний пристрій виявляє, що в прийнятому слові не виконується умова парності, це сприймається як сигнал помилки. Код з перевіркою парності дозволяє виявити всі одиничні (в одному розряді) помилки, а також всі випадки непарної кількості помилок. Код Хемінга дозволяє виправляти помилки. При його побудові до інформаційних розрядів слова добавляється певна кількість контрольних розрядів [6]. При зчитуванні

слова контрольна апаратура утворює із інформаційних і контрольних розрядів коригуюче число, яке дорівнює 0 при відсутності помилки або вказує місце помилки, наприклад, порядковий номер помилкового розряду в слові. Помилковий розряд автоматично коригується зміною його стану на протилежний.

Контроль виконання арифметичних операцій можна здійснити за допомогою контрольних кодів, які являють собою остачі від ділення чисел на деякий модуль R (контроль за модулем R). При контролі за модулем використовується властивість однаковості значень остачі за модулем R від результату виконання операції над операндами з остачею за модулем R від результату виконання цієї ж операції над остачами за модулем R операндів. Для двійкових чисел контроль за модулем можливий при $R \geq 3$. В ЕОМ часто використовують контроль за модулем 3. Він дозволяє виявляти будь-які одиничні помилки і частину подвійних помилок (тих, при яких для правильних і помилкових результатів остачі від ділення на 3 не збігаються). При використанні контролю за модулем, більшим за 3, збільшується кількість кратних помилок, які можуть виявлятися системою контролю, але також ростуть і апаратні витрати.

Контрольні запитання і задачі

1. Приведіть класифікацію АЛП з характеристикою їх властивостей.
2. Що таке мікрооперація, які бувають типи найбільш поширених мікрооперацій, як вони записуються?
3. Які мікрооперації називають сумісними?
4. Що таке логічні умови?
5. Приведіть умови коректності ГСА.
6. Що таке модифікований код, для чого він застосовується?
7. Сформулюйте узагальнений алгоритм виконання арифметичних операцій над числами з фіксованою і плаваючою комою.
8. Виконайте приклад додавання, множення та ділення чисел $+13$ та -7 як двійкових чисел з фіксованою комою, представлених 5-ма цифровими розрядами.
9. Які особливі випадки бувають при діленні чисел з фіксованою комою?
10. Як обробляються мантиси в операціях з плаваючою комою?
11. В чому полягають особливості обробки двійково-десяткових чисел?
12. Приведіть основні особливості АЛП мікропроцесорів.
13. Які методи контролю АЛП Ви знаєте?
14. Як здійснюється контроль з перевіркою парності?
15. Побудуйте структурну схему суматора з контролем за модулем 3.
16. Як обробляються порядки при множенні і діленні чисел з плаваючою комою?

4. ПРИСТРОЇ КЕРУВАННЯ

Час виконання однієї команди в ЕОМ називають циклом машини.

Протягом циклу ЕОМ виконує такі дії:

- зчитує чергову команду з ОП;
- визначає адреси операндів і адресу результату;
- зчитує операнди із ОП (для групи команд, що оперують з операндами, які знаходяться в ОП) і пересилає їх в АЛП;
- виконує операцію, код якої вказаний в команді;
- записує в ОП результат операції (якщо це потрібно);
- формує адресу наступної команди.

При цьому основними функціями пристрою керування (ПК) є читання чергової команди із ОП, зберігання команди протягом циклу, перетворення адресної частини команди (формування адрес, читання операндів і запис результатів), перетворення операційної частини команди (формування необхідної для даної операції послідовності керуючих сигналів, формування адреси наступної команди). До апаратних засобів, які реалізують перераховані функції, відносять блок керування командами (БКК), блок керування операціями (БКО) та блок центрального керування (БЦК) [16].

Необхідними вузлами БКК є реєстр команд, в якому зберігається команда протягом циклу, та лічильник команд, де формується і зберігається адреса наступної команди. Перетворення адресної та операційної частини команди здійснюється БКО. БЦК призначений для формування тактових сигналів, сигналів синхронізації та узгодження роботи БКК та БКО.

В залежності від способів керування операціями в ЕОМ розрізняють синхронний, асинхронний та змішаний принципи керування. Для блоків центрального керування, які реалізують синхронний принцип керування, характерним є однакова тривалість циклу, яка визначається часом, потрібним для виконання найбільш триваломісткої операції та однакова тривалість тактових проміжків часу. В блоках місцевого керування, які реалізують принцип асинхронного керування, для будь-якої операції в циклі використовується стільки тактів, скільки необхідно для виконання цієї операції. Наступна команда починає виконуватись зразу ж після формування спеціального сигналу закінчення попередньої операції. При змішаному керуванні виконанням операцій стандартний цикл встановлюється для груп операцій, які за своїм характером вимагають приблизно однакового часу їх виконання. Для виконання операцій, які потребують більших інтервалів часу, використовують вузли місцевого керування.

Вищим рівнем внутрішньої мови керування в машині є мова команд програми. Керування на цьому рівні здійснюється читанням записаних в ОП команд і їх виконанням. Кожній команді відповідає мікропрограма – послідовність мікрокоманд, виконання яких приводить до виконання

операції, заданої командою. Мікрокоманда керує виконанням однієї або декількох мікрооперацій. Мікрокомандний рівень керування розглядається як останній ступінь деталізації процесу керування операційними блоками на рівні елементарних дій.

Алгоритми виконання операцій можуть бути представлені в словесній формі, в операторній формі [16], в вигляді граф-схем і графів. ГСА, в яких в вершинах записується зміст мікрооперацій і логічних умов, називають змістовними. При побудові керуючих автоматів зручніше користуватись закодованими ГСА, де в вершинах проставлені умовні позначення мікрооперацій та логічних умов (прийнято позначати мікрооперацію u_i , а логічну умову x_i).

Пристрої керування, що генерують послідовність керуючих сигналів, яка визначається алгоритмом виконання операції та ознаковими сигналами, що поступають з операційного блока, називають керуючими або мікропрограмними автоматами. Розрізняють два основних типи керуючих автоматів: з "жорсткою" логікою та з програмованою логікою.

Керуючі автомати з "жорсткою" логікою

Для аналізу і синтезу автоматів з "жорсткою" логікою використовують апарат теорії абстрактних автоматів. Стосовно до керуючих автоматів автомат задається множиною вхідних сигналів X (ознакові сигнали), множиною вихідних сигналів Y (керуючі сигнали), множиною станів A та функціями переходу і виходу. Функція переходу f_1 визначає залежність наступного стану автомату від попереднього стану і вхідних сигналів, а функція виходу f_2 визначає залежність вихідних сигналів від попереднього стану і вхідних сигналів. В структурному плані автомат можна розділити на дві частини: комбінаційну КЧ і запам'ятовувальну ЗЧ (рис.4.1).

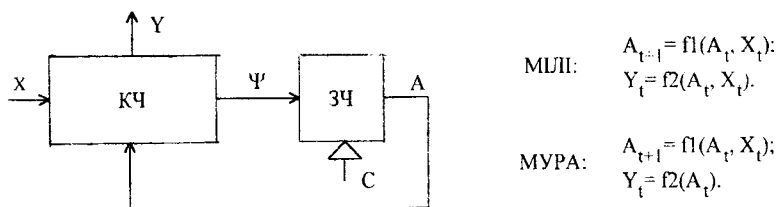


Рис. 4.1. Узагальнена структура цифрового автомата

Для автомата Мілі вихідні сигнали залежать від стану та вхідних сигналів, а для автомата Мура – тільки від стану автомата. Автомат може бути заданим в аналітичній формі, в табличній формі та направленим графом.

Канонічний метод структурного синтезу автомата, запропонований

В.М.Глушковым, дозволяє звести задачу синтезу автомата до задачі синтезу комбінаційної схеми шляхом побудови системи бульових функцій, які виражають залежність вихідних сигналів і сигналів збудження елементів пам'яті від вхідних сигналів і стану автомата. Відповідно до цього методу можна запропонувати такий порядок синтезу автомата:

- 1) перехід від змістовної ГСА до закодованої;
- 2) розмічування закодованої ГСА станами автомата;
- 3) побудова графа автомата;
- 4) вибір елементів пам'яті для ЗЧ і кодування станів автомата;
- 5) побудова суміщеної таблиці переходів і виходів;
- 6) визначення та мінімізація логічних функцій збудження елементів пам'яті і вихідних сигналів;
- 7) приведення визначених функцій до заданого базису;
- 8) побудова функціональної схеми автомата;
- 9) аналіз побудованої схеми і визначення основних параметрів пристрою.

Розглянемо деякі етапи синтезу більш детально. Розмічування ГСА станами автомата для автоматів Мілі і Мура здійснюється по різному. Для автомата Мура, в зв'язку з тим, що вихідні сигнали пов'язані тільки зі станами, правила розмічування такі:

- символом початкового стану a_0 відмічаються початкова і кінцева вершини;
- кожна операторна вершина відмічається символом стану a_i ;
- дві різні операторні вершини не можуть бути відмічені однаковими символами стану.

Для автомата Мілі правила розмічування ГСА станами автомата мають вигляд:

- символом a_0 відмічається вхід першої вершини, яка слідує за початковою, і вхід кінцевої вершини;
- входи вершин, які слідує за операторними вершинами, відмічаються символами стану a_i ;
- входи двох різних вершин, за виключенням кінцевої, не можуть бути відмічені однаковими символами;
- вхід вершини може відмічатись тільки одним символом.

Автомат Мілі може інтерпретувати мікропрограму коректно тільки в тому випадку, якщо для всіх переходів виконується умова незалежності логічних умов від мікрооперацій, що виконуються під дією керуючих сигналів, які відповідають цим переходам. Якщо умова незалежності порушується, усунути негативні явища цих випадків можна шляхом запам'ятовування логічних умов на проміжок такту, введенням додаткових станів або реалізації автомата Мура.

При побудові графа автомата стани утворюють вершини графа. Для автомата Мура в вершинах також проставляють вихідні сигнали, які пов'язані зі станами. Дуги між станами визначають переходи. Над дугами вказують логічні умови, при яких здійснюються відповідні переходи. Для автомата Мілі вихідні сигнали вказують над дугами поряд з логічними

умовами. Безумовні переходи позначають символом прочерку "-".

Елементами пам'яті ЗЧ автомата, як правило, є тригери, які утворюють регістр. Для вилучення наявності явища "гонок" (за рахунок зворотного зв'язку в пристрої керування), слід використовувати двоступеневі тригери або тригери з динамічним керуванням. Якщо нема необхідності використовувати так зване протигонкове кодування [14], стани кодують двійковими кодами. Це дозволяє використовувати дешифратор для отримання сигналів, що відповідають станам автомата.

В суміщеній таблиці переходів і виходів передбачаються стовпці, які відповідають попередньому стану і його коду, стану переходу і його коду, вхідним сигналам, вихідним сигналам та функціям збудження елементів пам'яті. Будувати таку таблицю зручно, використовуючи граф автомата, хоча можна її отримати і на основі закодованої і розміченої ГСА. Суміщена таблиця переходів і виходів може бути прямою і зворотною. В прямій таблиці впорядковані попередні стани і розглядаються переходи з одного початкового стану у всі можливі наступні. В зворотній таблиці впорядковані наступні стани і для кожного наступного стану розглядаються переходи в нього з усіх відповідних початкових. При побудові таблиці в стовпцеві, де вказуються вхідні сигнали, проставляють позначення логічних умов, які викликають відповідні переходи. В стовпцеві, де вказуються вихідні сигнали, для автомата Мілі проставляють відповідні вихідні сигнали, які з'являються на відповідних переходах. Для автомата Мура вихідні сигнали пов'язують з впорядкованими станами. Для стовпця, де вказуються функції збудження елементів пам'яті, визначають значення керуючих входів тригерів таким чином, щоб на цих входах в попередньому стані були такі значення, які призведуть до переходу в потрібний наступний стан.

Після побудови таблиці записують функції виходів і функції збудження елементів пам'яті. В зв'язку з тим, що станів і логічних умов може бути багато, повна мінімізація вручну може бути трудомісткою. В цьому випадку функції записують "по одиницях" (пряме значення функції) або "по нулях" (інверсне значення функції) в залежності від того, більше одиниць чи нулів у значеннях функції, що приведені в таблиці. Потім за допомогою аксіом і законів алгебри логіки отримані функції спростують таким чином, щоб мінімальною була їх сумісна реалізація. Іноді, коли критерієм проектування є максимум швидкодії, функції представляють таким чином, щоб витратилось найменше часу на їх формування в схемі. При перетворенні функцій їх приводять до заданого або зручного в реалізації базису.

При побудові функціональної схеми автомата передбачують установку його в початковий стан. Для цього, як правило, використовують додаткові "установочні" асинхронні входи тригерів. Для формування сигналів станів автомата часто використовують дешифратор. В зв'язку з тим, що дешифратори можуть мати інверсні виходи, переглядають функції, щоб уникнути лишніх інверсій в схемі. Слід також врахувати характер вихідних сигналів (потенціал протягом такту чи діючий фронт). Коли потрібен діючий фронт, вихідний сигнал відповідним чином логічно

перемножують з тактовим сигналом.

Аналіз роботи побудованої схеми проводять з метою перевірки її працездатності. Поряд з таким аналізом доцільно провести дослідження схеми шляхом її моделювання за допомогою сучасних програмних засобів. З основних параметрів визначають швидкодію (мінімальний період або максимальну частоту тактових сигналів, при яких схема залишається працездатною) та апаратні витрати на реалізацію пристрою.

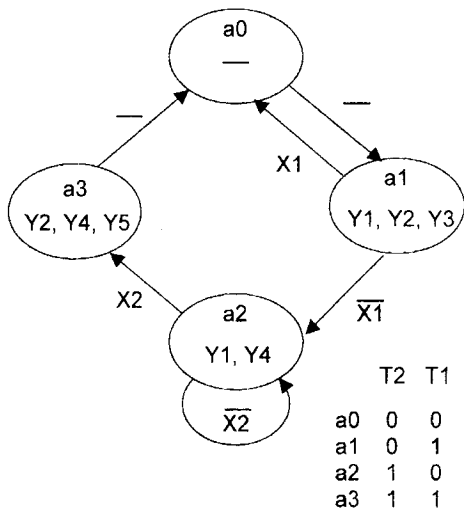
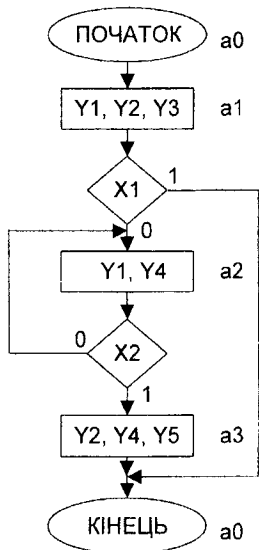
Методика і приклади побудови автоматів з жорсткою логікою приведені в [1]. Приклади синтезу автоматів Мілі і Мура показані відповідно на рис.4.2 та 4.3.

Одним з різновидів автоматів з "жорсткою" логікою є керуючі автомати, побудовані за принципом розподілення сигналів. Вони використовуються для реалізації так званих лінійних мікропрограм, в яких групи операторів Y_i ($i=1,2,3,\dots,N$) пов'язані з переходами на розміченій ГСА типу $a_iX_iY_{i+1}$ або $a_NX_NY_{N+1}$. Y_i – множина вихідних сигналів, яка може бути і порожньою.

Розподілювачем сигналів називають схему формування послідовності сигналів $a_1, a_2, a_3, \dots, a_N$, які відповідно приймають значення 1 в послідовні моменти часу $t=1, 2, 3,\dots,N$. Розподілювач сигналів може бути побудований на основі зсувного регістру, лічильника з дешифратором, лічильника Джонсона. При побудові керуючого автомата за принципом розподілення сигналів розмічування ГСА станами проводять за правилами, характерними для автомата Мілі. Оскільки переходи в автоматі чітко визначені, для отримання функцій для вихідних сигналів будують таблицю виходів. Використовуючи спеціальні структурні рішення, основані на аналізі мікропрограми, з використанням розподілювачів сигналів можна реалізовувати алгоритми керування, які містять розгалуження і цикли. В зв'язку з тим, що послідовність станів в автоматах на основі розподілювачів сигналів чітко регламентована, спрощуються засоби формування функцій збудження, тобто скорочуються апаратні витрати на реалізацію автомата. Однак при наявності непродуктивних переходів (під час яких не формується жоден керуючий сигнал) такий автомат вносить додаткову затримку в процес обробки інформації, чим знижує швидкодію всієї системи.

Керуючі автомати з програмованою логікою

В автоматах з "жорсткою" логікою алгоритм керування в кінцевому результаті представлений схемою і зміна такого алгоритму потребує заново здійснювати синтез схеми автомата. Це не завжди зручно, особливо на стадії налагодження алгоритмів керування. Цього можна уникнути, використовуючи для побудови керуючого автомата принцип програмного керування з операційно-адресною структурою керуючих слів. В цьому



Попередній стан	Код ПС	Стан переходу	Код СП	Вхідні сигнали	Вихідні сигнали	Функції збудження елементів пам'яті	
						D2	D1
a0	00	a1	01	-	-	0	1
a1	01	a2	10	X1	Y1, Y2, Y3	1	0
		a0	00	X1	-	0	0
a2	10	a3	11	X2	Y1, Y4	1	1
		a2	10	X2	-	1	0
a3	11	a0	00	-	Y2, Y4, Y5	0	0

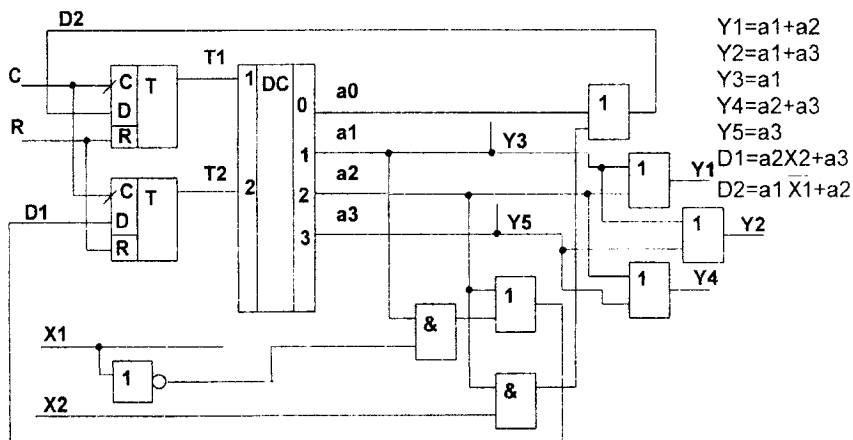


Рис. 3.2. Приклад побудови автомата Мура

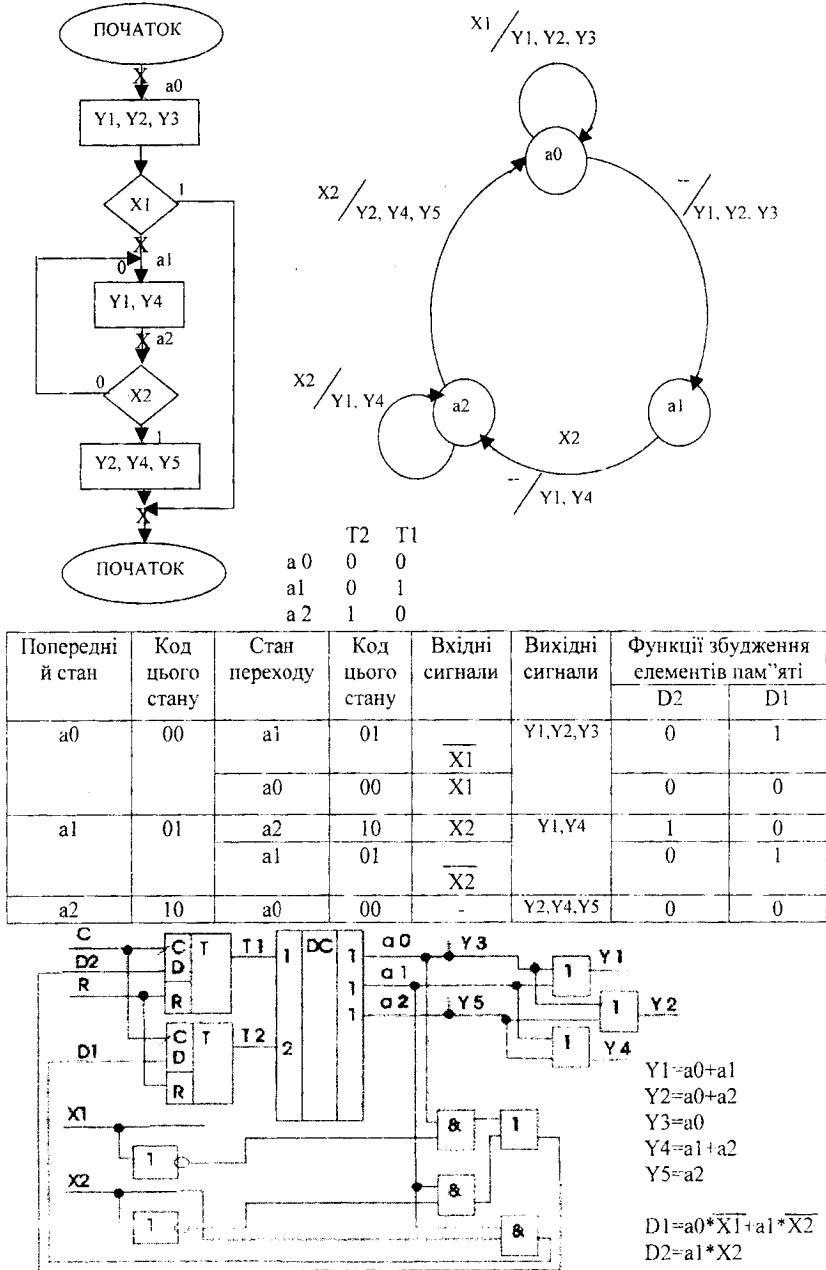


Рис.4.3. Приклад побудови автомата Мілі

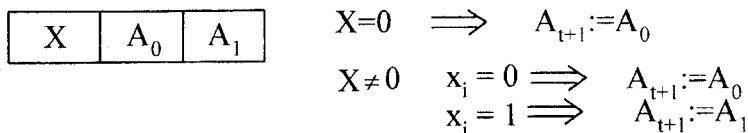
випадку алгоритм керування представляється впорядкованою сукупністю керуючих слів (мікрокоманд). Мікрокоманда містить інформацію про мікрооперації, які повинні виконуватись в даному такті роботи пристрою, і інформацію про адресу наступної мікрокоманди. Сукупність мікрокоманд, які виконують одну машинну команду або процедуру, утворює мікропрограму. Мікропрограма може зберігатись в ЗП. Для мікропрограм, що не змінюються в робочому режимі, використовують ПЗП. Звичайно мікрокоманда складається з двох частин: операційної та адресної.

В операційній частині мікрокоманди записуються коди керуючих сигналів. В залежності від способу кодування керуючих сигналів розрізняють горизонтальне, вертикальне та змішане (комбіноване) мікропрограмування. При горизонтальному мікропрограмуванні керуючі сигнали кодують унітарними кодами, тобто кожному керуючому сигналу відповідає один розряд в операційній частині мікрокоманди. При необхідності одночасного формування декількох керуючих сигналів одиничні значення проставляють в відповідних розрядах операційної частини мікрокоманди. Цей спосіб кодування керуючих сигналів зручний при їх відносно невеликій кількості. При великій кількості керуючих сигналів для зберігання мікропрограми необхідна пам'ять великої ємності та багаторозрядний регістр мікрокоманди. Вертикальне мікропрограмування передбачає керуючі сигнали кодувати двійковими кодами. В цьому випадку значно скорочується операційна частина мікрокоманди. Але для забезпечення одночасного формування декількох керуючих сигналів потрібно кодувати не тільки самі керуючі сигнали, але й ті їх сполучення, які разом зустрічаються в операційних вершинах. Окрім цього потрібен дешифратор на операційну частину мікрокоманди, який може бути досить складним. На практиці часто користуються комбінованим мікропрограмуванням, при якому операційна частина розділяється на поля, в межах яких керуючі сигнали кодуються вертикальним способом, тобто позиційними кодами.

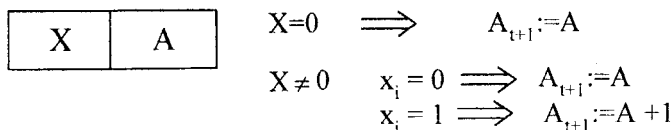
Адресна частина мікрокоманди призначена для визначення адреси наступної мікрокоманди. В найпростішому випадку, коли автомат для переходу на наступну мікрокоманду допускає аналіз тільки однієї логічної умови, в адресній частині виділяють поле умов, де вказується код цієї логічної умови. Окрім цього адресна частина містить поле адрес, де вказуються адреси мікрокоманд, що слідують за поточною мікрокомандою. Іноді, з метою скорочення довжини мікрокоманди, вказують тільки одну адресу, а іншу отримують з неї шляхом додавання одиниці за допомогою комбінаційного лічильника. Розглянутий спосіб називають примусовою адресацією з одним або двома адресними полями. В випадках, коли в алгоритмі зустрічається відносно мало логічних умов, адресу наступної мікрокоманди можна визначати за допомогою лічильника, використовуючи натуральну адресацію. При цьому використовують два типи мікрокоманд: операційну ОМК та керуючу

КМК. Вони представляються в рамках одних і тих же розрядів і розрізняються за значенням додаткового біта. Це дозволяє значно скоротити розрядність мікрокоманд і відповідно знизити ємність пам'яті мікропрограм. В операційній мікрокоманді вказують коди керуючих сигналів, а в адресній – інформацію про адресу наступної мікрокоманди, коли ця адреса не є більшою на 1 від попередньої. На рис.4.4. показані приклади форматів адресної частини мікрокоманди з умовами переходу для розглянутих способів адресації.

Примусова адресація з двома адресними полями



Примусова адресація з одним адресним полем

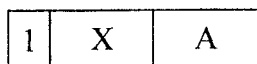


Натуральна адресація

$$\text{ОМК} : A_{t+1} := A_t + 1$$

КМК

$$\text{КМК} : X=0 \implies A_{t+1} := A$$



$$X \neq 0 \quad \begin{matrix} x_i = 0 \implies A_{t+1} := A \\ x_i = 1 \implies A_{t+1} := A_i + 1 \end{matrix}$$

Рис. 4.4. Формати адресної частини мікрокоманди з умовами переходу для різних способів адресації

Узагальнена структура керуючого автомата з програмованою логікою при комбінованому мікропрограмуванні і примусовою адресацією з двома адресними полями приведена на рис. 4.5. Її основними складовими є регістр мікрокоманди (РГМК), призначений для зберігання мікрокоманди протягом такту, пам'ять мікропрограм (ПМП), дешифратори операційних полів (DC1, DC2, DC3), мультиплексор логічних умов (MS) та мультиплексор адресних полів (показаний "жирною" лінією). В пристрої мікрокоманди послідовно, починаючи з початкової, читаються з ПМП в РГМК. Протягом такту за станом РГМК дешифраторами формуються відповідні керуючі сигнали, а за допомогою мультиплексорів в залежності від значень логічних умов визначається адреса наступної мікрокоманди.

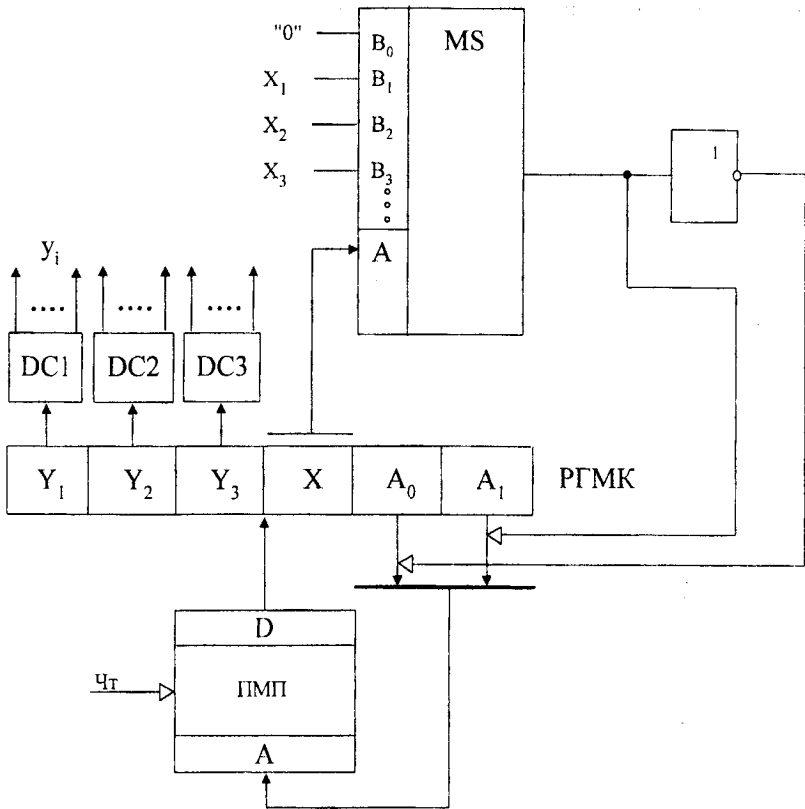


Рис. 4.5. Узагальнена структура керуючого автомата з програмованою логікою

При побудові пристроїв керування з програмованою логікою важливим етапом є розробка мікропрограми, яку іноді традиційно називають картою прошивки ПЗП.

До основних етапів синтезу керуючих автоматів з програмованою логікою слід віднести:

- вибір способу кодування керуючих сигналів, типу адресації та умов переходу;
- розподіл керуючих сигналів по полях операційної частини мікрокоманди (для комбінованого мікропрограмування) і кодування керуючих сигналів;
- кодування логічних умов;
- складання мікропрограми;
- побудова структурної та функціональної схеми автомата;

- аналіз результатів проектування і визначення основних параметрів пристрою.

У випадку комбінованого мікропрограмування при розподілі керуючих сигналів по полях операційної частини мікрокоманди враховують, що керуючі сигнали, які разом зустрічаються в операційних вершинах, повинні бути розміщені в різних полях. Для такого розміщення можна скористатися методом прямого включення [14, 16], який передбачає формалізацію процесу розподілу. Слід також враховувати, що для більш ефективного використання розрядної сітки бажано, щоб кількість керуючих сигналів в одному полі наближалась до величини $2^k - 1$, де k – розрядність цього поля (нульовою комбінацією кодується відсутність керуючих сигналів в полі).

При складанні мікропрограми використовують закодовану ГСА, коди керуючих сигналів і логічних умов та умови переходу. Мікропрограма має структуру мікрокоманди. Приклад розробки мікропрограми приведений на рис. 4.6 при комбінованому мікропрограмуванні і примусовій адресації з двома адресними полями.

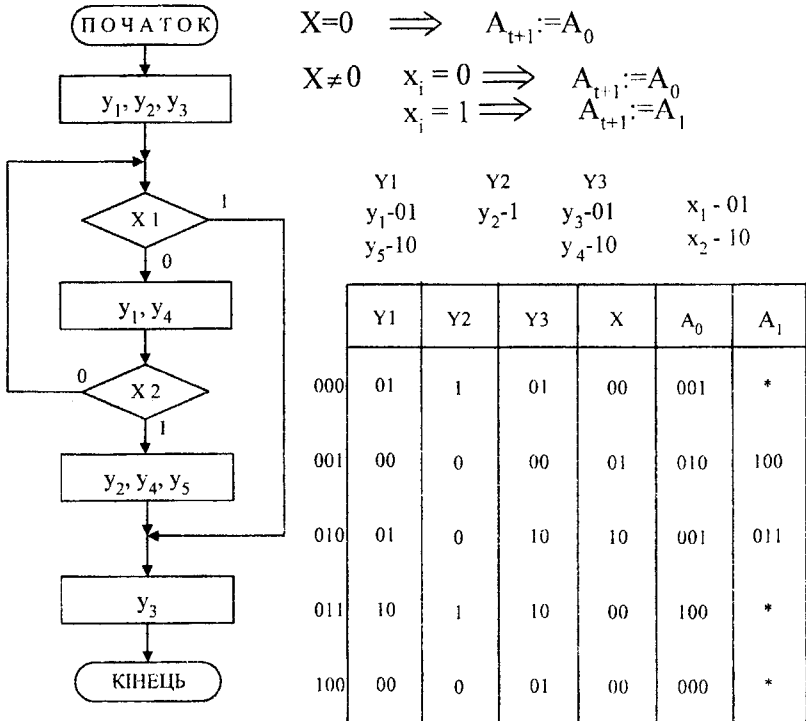


Рис. 4.6. Приклад розробки мікропрограми для керуючого автомата з програмованою логікою

Для покращання параметрів керуючих автоматів з програмованою логікою використовують сегментацію ПЗП, формування адрес з одночасним аналізом декількох умов, паралельну вибірку мікрокоманд [14, 16].

Мікропрограмний принцип керування широко використовується в мікропроцесорних пристроях секційного типу. В склад таких мікропроцесорних наборів включаються спеціальні блоки мікропрограмного керування (K584ВУ1, K589ИК01, K1800ВУ1, K1804ВУ1, K1804ВУ2, K1804ВУ4), призначені для визначення адреси наступної мікрокоманди. Ці блоки дозволяють організувати досить гнучку систему переходів в мікропрограмах.

Зручними для використання в схемах керуючих автоматів є програмовані логічні матриці (ПЛМ, PLA). Мікросхеми ПЛМ реалізують системи перемикальних функцій, представлених в ДНФ, кожна з яких складається з індивідуального набору відносно невеликої кількості кон'юнктивних термів. Основа ПЛМ – послідовне включення програмованої матриці елементів "1" та фіксованої матриці елементів "АБО". Принципи побудови керуючих автоматів з використанням ПЛМ розглянуті в [6, 18].

Порівнюючи керуючі автомати з "жорсткою" та програмованою логікою, можна зробити висновок, що для досить складних алгоритмів керування автомати з "жорсткою" логікою вимагають більших апаратних витрат, але є більш швидкодіючими. Нижча швидкодія автоматів з програмованою логікою пояснюється наявністю непродуктивних переходів в мікропрограмах. Але вони забезпечують більшу гнучкість внесення змін в закон функціонування автоматів при зміні алгоритму керування.

Контрольні запитання і задачі

1. Які дії виконуються протягом циклу машини?
2. Назвіть основні функції ПК.
3. Що таке синхронний, асинхронний та змішаний принципи керування?
4. Які форми представлення алгоритмів Ви знаєте?
5. Чим відрізняються автомати Мура від автоматів Мілі?
6. В чому полягає канонічний метод структурного синтезу автоматів?
7. Сформулюйте правила розмітки ГСА станами автомата Мілі та Мура.
8. Що вказують в суміщеній таблиці переходів і виходів?
9. В чому полягають особливості автоматів, побудованих за принципом розподілення сигналів?
10. В чому полягає основний принцип побудови керуючих автоматів з програмованою логікою?
11. Які способи кодування керуючих сигналів Вам відомі?
12. Назвіть основні етапи синтезу керуючих автоматів з програмованою логікою.
13. Дайте порівняльну характеристику автоматів з ЖЛ та ПЛ.

5. ПРОЦЕСОРИ

Пристрій керування разом з операційним пристроєм прийнято називати процесорним пристроєм або процесором в зв'язку з тим, що тут проходять основні процеси обробки інформації. Процесор дешифрує і виконує команди програми, організовує звертання до оперативної пам'яті, в потрібних випадках ініціює роботу периферійних пристроїв, сприймає і обробляє запити переривання. Процесор займає центральне місце в структурі ЕОМ, оскільки він здійснює керування взаємодією всіх пристроїв, що входять до складу ЕОМ.

Виконання команди (машинної операції або інструкції) може бути розділено на більш дрібні етапи (мікрооперації), під час яких виконуються певні елементарні дії. Конкретний склад мікрооперацій і послідовність їх виконання визначаються системою команд, логічною структурою і особливостями даної ЕОМ. Для визначення часових співвідношень між різними етапами операції використовується поняття "машинний такт". Машинний такт визначає інтервал часу, протягом якого виконується одна або одночасно декілька мікрооперацій процесора. Межі тактів задаються синхросигналами, що виробляються спеціальним генератором. Таким чином, може бути встановлена ієрархія етапів виконання програм в процесорі: програма, команда (мікропрограма), мікрооперація (мікрокоманда).

Можна виділити три способи ініціювання виконання машинних команд. В відповідності з першим з них машинні команди векторних операцій суперкомп'ютерів і машинні команди універсальних обчислювальних машин поступають в процесор в порядку їх розміщення в програмі і виконуються послідовно. Другий і третій способи передбачають незалежне ініціювання і одночасне виконання машинних команд. Так працюють машини, які керуються даними, і машини, які керуються запитами.

В комп'ютерних системах (з контролерним керуванням) виконання програм, представлених в машинних кодах, здійснюється під керуванням спеціального пристрою керування. Машинна команда, адреса якої вказана в програмному лічильнику, читається із основної пам'яті, декодується. Потім дані, вказані в адресній частині команди, читаються з регістрів і основної пам'яті, і в операційному пристрої над ними виконуються дії, які задані в операційній частині команди. Результати операцій записуються в регістри або в основну пам'ять. Команда, яка буде виконуватись в наступному циклі, звичайно читається за адресою, отриманою шляхом додавання одиниці до вмісту програмного лічильника. В випадку команд безумовного або умовного переходу адреса наступної команди вказується в адресній частині. При виникненні переривань здійснюється перехід до підпрограм обробки переривань. Таким чином, в системі з контролерним керуванням виконання машинних команд здійснюється послідовно.

Для вирішення будь-яких задач ЕОМ повинні мати алгоритмічно

повну систему команд. Теоретично доказано, що мінімальна алгоритмічно повна система команд складається з однієї універсальної команди. Але використання системи з малим набором команд призводить до неекономічних за обсягом пам'яті і часом виконання "довгих" програм. Тому в ЕОМ в основному використовуються системи команд, які містять декілька десятків або сотень команд. При цьому вдається отримати компактний запис алгоритмів і відповідно ефективні програми. При проектуванні процесорів вирішують задачу визначення наборів команд, які реалізуються апаратним та програмним способом. Апаратна реалізація команд дозволяє підвищити швидкість пристрою, але недоцільна в випадках, коли малий обсяг обчислювальних операцій даного типу призводить до низької ефективності використання апаратних засобів.

В залежності від повноти набору команд процесора розрізняють RISC (Reduced Instruction Set Computer) та CISC (Complex Instruction Set Computer) архітектури.

В архітектурі RISC закладено три основних принципи: використання простого набору команд, які реалізуються апаратними засобами; організація пам'яті і введення-виведення, що дозволяють виконати більшість команд за один такт; орієнтація системи на підтримку мови високого рівня за допомогою компілятора. Скорочення набору команд здійснюється за рахунок команд, які рідко використовуються (в 80% обчислювальних операцій універсальних процесорів використовується 20% команд всього набору).

CISC-процесори, до яких відноситься і сімейство x86, мають повний набір команд. Склад і призначення їх регістрів суттєво неоднорідні, широкий набір команд ускладнює їх декодування, на що витрачаються апаратні ресурси. Зростає кількість тактів, необхідних для виконання команд.

В процесорах сімейства x86, починаючи з 486, використовується комбінована архітектура - CISC-процесор має RISC-ядро.

Систему команд процесора часто поділяють на групи команд: команди загального призначення (пересилання, перетворення форматів, присвоювання констант); арифметичні (додавання, віднімання, множення, ділення, приріст, обчислення остачі, здобування квадратного кореня, зміна знака); порівняння (порівняння з нулем, з іншим операндом, для співвідношень "більше", "менше", "дорівнює"); логічні ("І", "АБО", "Інверсія", переміщення біт); передачі керування (безумовні і умовні переходи, переходи на підпрограму і повернення з підпрограми); команди керування (зупинка, міжпроцесорна сигналізація, прямі читання і запис, робота з портами введення-виведення).

Домінуюче положення на ринку універсальних процесорів в мікропроцесорному виконанні займають мікропроцесори фірми Intel з системою команд x86. Решта виробників універсальних мікропроцесорів випускають RISC-процесори, сумарна доля яких складає 10% ринку.

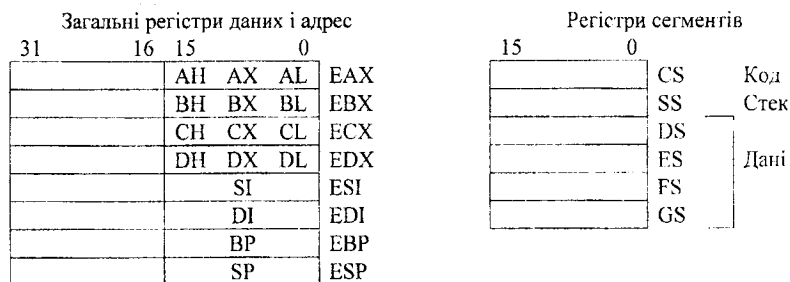
Характеристика основних типів мікропроцесорів подається в розділі 8.

В подальшому архітектура процесора розглядається на прикладі базової програмної моделі 32-розрядних процесорів Intel, яка є спільною для всіх існуючих на даний момент 32-розрядних процесорів: 80386, 486, Pentium, Pentium Pro, Pentium II і Celeron. Ця модель охоплює набір регістрів процесора, організацію пам'яті і введення-виведення, типи даних, систему команд, переривання і винятки.

32-розрядні процесори можуть працювати в одному з двох режимів і переключатись між ними достатньо швидко як в один, так і в інший бік. Режим реальної адресації (або просто реальний режим -- Real Mode) повністю сумісний з 8086. В цьому режимі можлива адресація до 1 Мб фізичної пам'яті. Захищений режим віртуальної адресації (або просто захищений режим -- Protected Mode) дозволяє адресувати до 4 Гбайт фізичної пам'яті, через які при використанні механізму сторінкової адресації можуть відображатись до 64 Тбайт віртуальної пам'яті кожної задачі. Суттєвим доповненням є режим віртуального процесора 8086 (Virtual 8086 Mode). Цей режим є особливим станом задачі захищеного режиму, в якому процесор функціонує як 8086.

Регістри процесора

Основні регістри процесора показані на рис. 5.1. Регістри загального призначення включають відповідні регістри 16-розрядних процесорів, але тут вони мають 32 розряди. До попереднього позначення їх імен додалась приставка E (Extended - розширений). Відсутність приставки в імені означає посилання на молодші 16 біт розширених регістрів. Можливе незалежне звертання до молодшого і старшого байтів регістрів AX, BX, CX, DX. Розрядність операнду даних може складати 1, 8, 16, 32 або 64 біт, бітового поля 1-32 біта, адреси 16 або 32 біта.



Вказівник інструкцій і регістр ознак

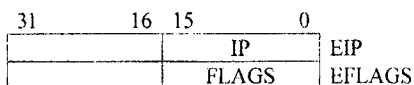


Рис. 5.1. Основні регістри 32-розрядних процесорів

Вказівник інструкцій EIP містить зміщення наступної інструкції, що виконується, відносно бази сегмента коду. При 16-бітній адресації використовуються тільки молодші 16 біт (IP).

Регістр ознак EFLAGS (рис. 5.2) має 32 біти, їх призначення таке:

- **ID** (Id Flag - ознака доступності команди ідентифікації CPUID (P5+ і деякі 486);
- **VIP** (Virtual Interrupt Pending) - віртуальний запит переривання (P5+);
- **VIF** (Virtual Interrupt Flag) – віртуальна версія ознаки (дозвіл на переривання) для багатозадачних систем (P5+);
- **AC** (Alignment Check) - ознака контролю вирівнювання;
- **VM** (Virtual 8086 Mode) - в захищеному режимі включає режим віртуального 8086;
- **RF** (Resume Flag) – ознака відновлення, використовується сумісно з регістрами точок зупинки;
- **IOPL** (Input/Output Privilege Level) - рівень привілей введення-виведення;
- **NT** (Nested Task Flag) - ознака вкладеної задачі;
- **OF** (Overflow Flag) - ознака переповнення;
- **DF** (Direction Flag) - ознака керування напрямом в рядкових операціях;
- **IF** (Interrupt-enable Flag) - ознака керування перериваннями;
- **TF** (Trap Flag) - ознака трасування (покрокового режиму);
- **SF** (Sign Flag) - ознака знака;
- **ZF** (Zero Flag) - ознака нульового результату;
- **AF** (Auxiliary Flag) - ознака додаткового перенесення в тетрадї для десяткової арифметики;
- **PF** (Parity Flag) - ознака парності;
- **CF** (Carry Flag) - ознака перенесення.

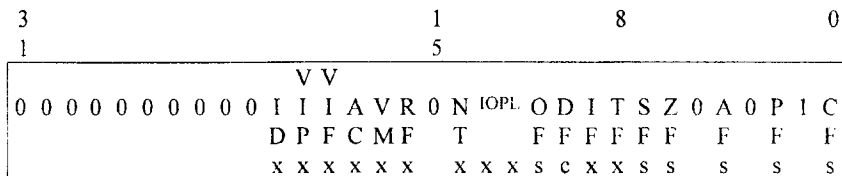


Рис. 5.2. Регістр ознак EFLAGS (x – системна ознака, s - ознака стану, c - керуюча ознака)

Регістри сегментів містять 16-бітові вказівники (в реальному режимі) або селектори дескрипторів (в захищеному режимі) сегментів кодів команд CS (Code Segment), стека SS (Stack Segment), даних DS (Data Segment), додаткових сегментів даних ES, FS, GS. Використання сегментних реєстрів визначається типом звернень до пам'яті. З кожним з шести сегментних реєстрів пов'язані програмно недоступні сховані реєстри дескрипторів, які автоматично завантажуються при завантаженні

- Сполучення MP=0, EM=0, TS=0, яке встановлюється при апаратній установці початкового стану, забезпечує повну сумісність з 8086/88, сполучення MP=1, EM=0 використовується при наявності співпроцесора, а MP=0, EM=1 при його програмній емуляції.

Регістр CR1 не використовується. Регістр CR2 зберігає 32-бітову лінійну адресу, за якою відбулась остання відмова сторінки пам'яті. Регістр CR3 в старших 20 бітах зберігає фізичну базову адресу таблиці каталогу сторінок. З молодших 12 біт в 486+ використовуються: PCD (Page-Level Cache Disable) – заборона кешування сторінки; PWT (Page-Level Writes Trough) – кешування сторінки з наскрізним записом.

Регістр CR4 (присутній в процесорах Pentium і старше) містить біти дозволу архітектурних розширень: VME (Virtual-8086 Mode Extensions) – дозвіл використання віртуальної ознаки переривання в режимі V86, що дозволяє підвищити продуктивність за рахунок скорочення викликів монітора віртуальних машин; PVI (Protected-Mode Virtual Interrupts) – дозвіл використання віртуальної ознаки переривання в захищеному режимі; TSD (Time Stamp Disable) – перетворення інструкції RDTSC (читання лічильника міток реального часу) в привілейовану; DE (Debugging Extensions) – розширення налагодження (дозвіл точок зупинки на інструкціях звертання до заданих портів введення-виведення); PSE (Page Size Extension) – розширення розміру сторінки (4 Кбайт і 4 Мбайт); PAE (Physical Address Extension) – розширення фізичної адреси (сторінки 4 Кбайт і 2 Мбайт. 36-бітна адресація); MCE (Machine-Check Enable) – розширення машинного контролю; PGE (Paging Global Extensions) – розширення глобальності в сторінковій переадресації; PCE (Performance-monitoring Counter Enable) – дозвіл звертання до лічильників подій на будь-якому рівні привілей.

Системні адресні реєстри призначені для посилань на сегменти і таблиці в захищеному режимі (рис. 5.4).

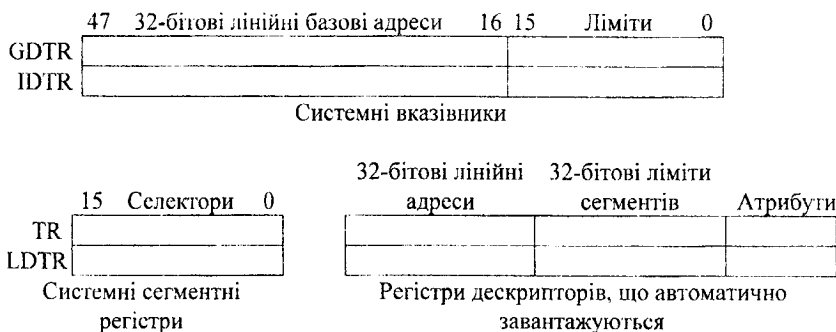


Рис. 5.4. Системні адресні реєстри

Регістри GDTR (Global Descriptor Table Register) і IDTR (Interrupt Descriptor Table Register) програмно завантажуються 6-байтовими операндами, які включають 32-бітову лінійну базову адресу і 16-бітовий ліміт глобальної таблиці дескрипторів і таблиці дескрипторів переривань. В регістр задачі TR (Task Register) і регістр селектора локальної таблиці дескрипторів LDTR (Local Descriptor Table Register) завантажуються 16-бітові селектори дескрипторів сегменту стану задачі TSS і локальної таблиці дескрипторів LDT. Це завантаження викликає автоматичне завантаження самих дескрипторів, які містять 32-бітові поля лінійної базової адреси і ліміту, а також полів атрибутів в пов'язані з ними невидимі регістри дескрипторів.

Регістри налагодження (Debug Register) призначені для завдання і керування точками зупинки при налагодженні. Склад регістрів тестування (Test Register) змінюється в залежності від типу процесора. Модельно-специфічні регістри MSR (Model-Specific Registers) призначені для керування розширеннями налагодження, моніторингом продуктивності, машинним контролем, кешуванням областей фізичної пам'яті та іншими функціями. Їх призначення прив'язується до архітектури конкретного процесора, склад змінюється від моделі до моделі, доступ привілейований. Доступність регістрів різних груп залежить від режиму роботи процесора і рівня привілей задачі.

Організація пам'яті

При взаємодії з пам'яттю розрізняють байти (8 біт), слова (16 біт) та подвійні слова (32 біта). Слова записуються в двох суміжних байтах, починаючи з молодшого. Адресою слова є адреса його молодшого байта. Подвійні слова записуються в чотирьох суміжних байтах, починаючи з молодшого, адреса якого і є адресою подвійного слова.

Більш великими одиницями є сторінки і сегменти. Пам'ять може логічно організуватись в вигляді одного або множини сегментів змінної довжини (в реальному режимі – фіксованій). Сегменти можуть вивантажуватись на диск і по мірі необхідності з нього довантажуватись в фізичну пам'ять. Окрім сегментації, в захищеному режимі можливий розподіл логічної пам'яті на сторінки розміром 4 Кбайт, кожна з яких може відобразитись на будь-яку область фізичної пам'яті. Починаючи з 5-го покоління, є можливість збільшення розміру сторінки до 4 Мбайт. Сегментація і розподіл на сторінки можуть використовуватись в будь-яких сполученнях. Сегментація є засобом організації логічної пам'яті, що використовується на прикладному рівні. Розподіл на сторінки використовується на системному рівні для керування фізичною пам'яттю.

Розрізняють три адресних простори: логічний, лінійний та фізичний. Основним режимом роботи 32-розрядних мікропроцесорів є захищений режим, в якому працюють всі механізми перетворення адресних просторів.

Логічна адреса, яку також називають віртуальною, складається із селектора сегмента (в реальному режимі – просто адреса сегмента) і ефективної адреси, яку називають також зміщенням (offset). Ефективна адреса формується додаванням компонентів base, index, displacement з врахуванням масштабу scale. Оскільки кожна задача може мати до 16 селекторів, а зміщення, обмежене розміром сегмента, може досягати 4 Гбайт, логічний адресний простір для кожної задачі може досягати 64 Тбайт. Весь цей простір віртуальної пам'яті в принципі доступний програмісту при умові підтримки з боку операційної системи.

Блок сегментації транлюс логічний адресний простір в 32-бітовий простір лінійних адрес, які утворюються додаванням базової адреси сегмента з ефективною адресою (зміщенням). Базова адреса сегмента в реальному режимі утворюється множенням вмісту сегментного реєстра на 16 (як і в 8086). В захищеному режимі базова адреса завантажується із дескриптора, який зберігається в таблиці, по селектору, що завантажений в сегментний реєстр, який використовується.

32-бітова фізична адреса пам'яті утворюється після перетворення лінійної адреси блоком сторінкової переадресації. Вона виводиться на зовнішню шину адреси процесора. В простому випадку (при відключеному блоку сторінкової переадресації) фізична адреса збігається з лінійною. Включений блок сторінкової переадресації здійснює трансляцію лінійної адреси в фізичну розміром 4 Кбайт (можливий розмір 2 або 4 Мбайти для старших поколінь процесорів). Блок забезпечує розширення розрядності фізичної адреси процесорів шостого покоління до 36 біт. Блок переадресації може включатись тільки в захищеному режимі.

Система команд 32-розрядних процесорів передбачає 11 режимів адресації. При звертанні до пам'яті ефективна адреса обчислюється з використанням таких компонентів:

- Зміщення (Displacement або Disp) - 8-, 16- або 32-бітове число, включене в команду.
- База (Base) - вміст базового реєстра. Звичайно використовується для визначення початку деякого масиву.
- Індекс (Index) - вміст індексного реєстра. Звичайно використовується для вибору елемента масиву.
- Масштаб (Scale) - множник (1, 2, 4 або 8), вказаний в кодї інструкції. Цей елемент використовується для визначення розміру елемента масиву, доступний тільки при 32-бітній адресації.

Ефективна адреса обчислюється за формулою:

$$EA = Base + Index * Scale + Disp.$$

Окремі доданки в цій формулі можуть бути відсутніми. Можливі режими адресації приведені в табл. 5.1. Процесор може працювати з 32- або 16-бітовою адресацією. 16-бітова адресація працює так само, як і в 8086 і 80286, при цьому в якості компонентів адреси використовуються молодші

16 біт відповідних регістрів. При 32-бітовій адресації використовуються розширені 32-розрядні регістри і застосовуються додаткові режими з масштабуванням індексу.

Таблиця 5.1

Режими адресації пам'яті 32-розрядних процесорів

Режим	Адреса
Пряма адресація	$EA = Disp$
Непряма регістрова адресація	$EA = Base$
Базова адресація	$EA = Base + Disp$
Індексна адресація	$EA = Index + Disp$
Масштабована індексна адресація	$EA = Index * Scale + Disp$
Базово-індексна адресація	$EA = Base + Index$
Масштабована базово-індексна адресація	$EA = Base + Index * Scale$
Базово-індексна адресація зі зміщенням	$EA = Base + Index + Disp$
Масштабована базово-індексна адресація зі зміщенням	$EA = Base + Index * Scale + Disp$

Введення-виведення

Як і процесор 8086, 32-розрядні процесори дозволяють адресувати до 64 К одnobайтових або 32 К двобайтових регістрів у просторі, окремому від пам'яті. Додатково є можливість звертання до 32-бітових портів. При операціях введення-виведення лінії $A[16:31]$ не використовуються. Адреса пристрою задається або в команді (тільки молодший байт, старший - нульовий), або береться з регістра DX (повна 16-бітова адреса).

В захищеному режимі інструкції введення-виведення є привілейованими. Це означає, що вони можуть виконуватись задачами тільки з певним рівнем привілей, який визначається полем IOP регістра ознак або бітовою картою дозволу введення-виведення, що зберігається в сегменті стану задачі.

Переривання і винятки

Переривання і винятки порушують нормальний хід виконання програми для обробки зовнішніх подій або повідомлення про виникнення особливих умов або помилок.

Переривання бувають апаратні (що маскуються або не маскуються), які викликаються електричними сигналами на входах процесора, і програмні, які виконуються за командою $INTxx$. Програмні переривання по суті такими не являються - це лиш своєрідний спосіб виклику процедур, але процесором вони обробляються як різновид переривань. Процесор може сприймати переривання після виконання кожної команди, довгі рядкові команди мають для сприйняття переривань спеціальні вікна. Апаратні переривання викликаються сигналами на входах $INTR$ і NMI , а для процесорів старших поколінь вони можуть приходити по шині $APIC$.

Переривання, що маскуються, викликаються перехоном в високий

рівень сигналу на вході INTR при встановленій ознаці дозволу (IF=1). Переривання, що не маскуються, виконуються незалежно від стана ознаки IF за сигналом NMI.

Випятки, або особливі випадки, розділяються на відмови, пастки і аварійні завершення.

Відмова (fault) - це виняток, який виявляється і обслуговується до виконання інструкції, що викликала помилку. Після обслуговування цього винятку керування повертається знову на ту ж інструкцію, яка викликала відмову.

Пастка (trap) - це виняток, який виявляється і обслуговується після виконання інструкції, що його викликала. Після обслуговування цього винятку керування повертається на інструкцію, яка слідує за тією, що викликала пастку. До класу пасток відносяться і програмні переривання.

Аварійне завершення (abort) - це виняток, який не дозволяє точно встановити інструкцію, що його викликала. Воно використовується для повідомлень про серйозну помилку, таку як апаратна помилка або пошкодження системних таблиць.

Якщо під час обслуговування винятків відмови сторінки відбудеться ще одна відмова сторінки, то станеться аварійна зупинка процесора. Під час аварійної зупинки ніякі нові інструкції не виконуються. З цього стану процесор можна вивести тільки апаратно сигналом NMI, залишаючи його в захищеному режимі, або сигналом RESET, що переводить процесор в реальний режим.

Типи даних

32-розрядні процесори безпосередньо підтримують такі типи даних:

- **Біт** – одиниця інформації. Біт в пам'яті задається базою (адресою слова) і зміщенням (номером біта в слові).
- **Бітове поле** – група до 32 суміжних біт, які розміщуються не більше ніж в 4 байтах.
- **Бітовий рядок** – набір суміжних біт довжиною до 4 Гбіт.
- **Байт** – 8 біт.
- **Числа без знака**: байт/слово/подвоєне слово/четвірне слово – 8/16/32/64 біт.
- **Цілі числа зі знаком**: байт/слово/подвоєне слово/четвірне слово. Одиничне значення найстаршого біта (знак) є ознакою від'ємного числа, яке зберігається в доповняльному коді.
- **Дійсні числа в форматі з плаваючою точкою**:
 - одинарної точності – 32 біти (23 біти мантиса і 8 біт порядок);
 - подвійної точності – 64 біти (52 біти мантиса і 11 біт порядок);
 - підвищеної точності – 80 біт (64 біти мантиса і 15 біт порядок).
- **Двійково-десяткові числа**:
 - 8-розрядні упаковані, які містять два десяткових розряди в одному байті;
 - 8-розрядні неупаковані, які містять один десятковий розряд в байті;

- 80-розрядні упаковані (обробляються тільки співпроцесором).
- **Рядки байт, слів і подвійних слів.**
- **Вказівники:**
 - довгий вказівник (48 біт) – 16-бітовий селектор (або сегмент) і 32-бітове зміщення;
 - короткий вказівник – 32-бітове зміщення;
 - просто вказівник (32 біти, єдиний тип вказівника для 8086 і 80286) -- 16-бітовий селектор (або сегмент) і 16-бітове зміщення.

В сімействі 80x86 прийнято, що слова записуються в двох суміжних байтах пам'яті, починаючи з молодшого. Адресою слова є адреса його молодшого байта. Подвійні слова записуються в чотирьох суміжних байтах, знову-таки починаючи з молодшого байта, адреса якого і є адресою подвійного слова.

Система команд

Система команд 32-розрядних процесорів є суттєво розширеною системою команд 8086/80286. Розширення стосуються збільшення розрядності адрес і операндів, більш гнучкої системи адресації, появи принципово нових типів даних (бітові рядки і поля) і команд.

Команди (інструкції) містять одно- або двобайтний код інструкції, за яким може розміщуватись декілька байт, що визначають режим виконання команди і операнди. Команди можуть використовувати до трьох операндів (або жодного). Результат виконання записується в місце, яке визначається інструкцією, або в місце, задане операндом призначення. Операнди можуть знаходитись в пам'яті, регістрах процесора або безпосередньо в команді. Перед будь-якою інструкцією може бути використаний префікс переключення розрядності адреси або слова. При адресації пам'яті використання сегментного регістра, передбаченого командою, в ряді інструкцій може подавлятися префіксом зміни сегмента.

Команди 32-розрядних процесорів можуть бути розділені на групи: пересилання даних; двійкової арифметики; десяткової арифметики; логічних операцій; зсуви і повертання (циклічні зсуви); обробки біт і байт; передачі керування; рядкових операцій; операцій з ознаками; завантаження вказівників; інструкції MMX; математичного співпроцесора; системні. Детально система команд розглядається в [5].

Основні поняття захищеного режиму

Захищений режим призначений для забезпечення незалежності виконання декількох задач, що має на увазі захист ресурсів однієї задачі від можливого впливу іншої (під задачами маються на увазі як прикладні програми, так і задачі операційної системи). Основним ресурсом, що захищається, є пам'ять, де зберігаються коди, дані і різні системні таблиці. Захищати потрібно і апаратуру, що сумісно використовується.

Захист пам'яті оснований на сегментації. Сегмент – це блок простору пам'яті певного призначення. Максимальний розмір сегмента – 4 Гбайт

(для процесорів 8086 і 80286 – всього 64 Кбайт). Сегменти пам'яті виділяються задачам операційною системою, але в реальному режимі будь-яка задача може перевизначити значення сегментних реєстрів, які задають положення сегмента в просторі пам'яті, і "залізи" в чужу область даних або коду. В захищеному режимі сегменти теж розподіляються операційною системою, але прикладна програма зможе використати тільки дозволені для неї сегменти пам'яті, вибираючи їх за допомогою селекторів із попередньо сформованих таблиць дескрипторів сегментів.

Процесор може звертатись тільки до тих сегментів пам'яті, для яких є дескриптори в таблицях. Механізм сегментації формує лінійну адресу за схемою: логічна адреса (селектор і зміщення) – селектор – таблиця дескрипторів – дескриптор сегмента (базова адреса) – (базова адреса + зміщення) – лінійна адреса.

Дескриптори - це 8-байтові структури даних, які використовуються для визначення властивостей програмних елементів (сегментів, вентилів і таблиць). Дескриптор визначає положення елемента в пам'яті, розмір області, що займає цей елемент (ліміт), його призначення і характеристики захисту. Всі дескриптори зберігаються в таблицях, звертання до яких підтримується процесором апаратно.

Захищений режим має засоби переключення задач. Стан кожної задачі (значення всіх пов'язаних з нею реєстрів процесора) може бути збережений в спеціальному сегменті стану задачі TSS, на який показує селектор в реєстрі задачі TR. При переключенні задач достатньо завантажити новий селектор в реєстр задачі, і стан поточної задачі автоматично збережеться в її TSS, а в процесор завантажиться стан нової (можливо, раніше перерваної) задачі, і почнеться її виконання.

Ієрархічна система привілеїв має чотири рівні. Вона призначена для керування використанням привілейованих інструкцій і доступом до дескрипторів. Рівні привілеїв нумеруються від 0 до 3, нульовий рівень відповідає максимальним (необмеженим) можливостям доступу і відводиться для ядра операційної системи. Рівень 3 має найобмеженіші права і звичайно виділяється прикладним задачам. Систему захисту звичайно зображують в вигляді концентричних кілець, які відповідають рівням привілеїв, а самі рівні привілеїв іноді називають кільцями захисту. Сервіси, які виділяються задачам, можуть знаходитись в різних кільцях захисту. Передача керування між задачами контролюється вентилями, які називають також шлюзами. Вони перевіряють правила використання рівнів привілеїв. Через вентилі задачі можуть отримати доступ тільки до дозволених їм сервісів інших сегментів.

Дескриптори і привілеї є основою системи захисту: дескриптори визначають структури програмних елементів, а привілеї – можливість доступу до дескрипторів і виконання привілейованих інструкцій. Будь-яке порушення захисту призводить до виникнення спеціальних винятків, які обробляються ядром операційної системи.

Існує три типи таблиць дескрипторів – локальна таблиця дескрипторів LDT (Local Descriptor Table), глобальна таблиця дескрипторів GDT (Global Descriptor Table) і таблиця дескрипторів переривань IDT (Interrupt Descriptor Table). Розміри таблиць можуть знаходитись в межах 8 байт – 64 Кбайт, що відповідає кількості елементів в таблиці від 1 до 8К. З кожною з цих таблиць пов'язаний відповідний регістр процесора (LDTR, GDTR, IDTR). Глобальна таблиця містить дескриптори, доступні всім задачам. Локальна таблиця може бути власною для кожної задачі і містить тільки дескриптори сегментів, вентилів задачі і викликів. Сегмент недоступний задачі, якщо його дескриптора нема ні в GDT ні в LDT. Таблиця дескрипторів переривань, яка використовується в захищеному режимі, може містити описи до 256 переривань. Формати дескрипторів приведені в [5].

В архітектурі процесорів п'ятого і шостого поколінь (Pentium, Pentium Pro, Pentium MMX, Pentium II) суттєве значення має реалізація різних способів конвеєризації і розпаралелювання обчислювальних процесів. Скалярним називають процесор з єдиним конвеєром. Суперскалярний процесор має більше одного конвеєра, які здатні обробляти інструкції паралельно. Pentium є двопотоковим процесором (має два конвеєри), Pentium Pro - трипотоковим.

Порівняльна характеристика процесорів сімейства x86 приведена в табл. 5.2.

Контрольні запитання і задачі

1. Який пристрій називають процесором і чому?
2. Охарактеризуйте властивості RISC- та CISC-процесорів.
3. На які групи команд поділяють систему команд процесора?
4. Охарактеризуйте склад основних регістрів 32-розрядних процесорів.
5. Охарактеризуйте ознаки регістра ознак EFLAGS.
6. Яке призначення мають керуючі регістри 32-розрядних мікропроцесорів?
7. Які бувають системні адресні регістри? Для чого вони призначені?
8. Охарактеризуйте організацію пам'яті 32-розрядних процесорів.
9. Які режими адресації Ви знаєте? Як визначається адреса в цих режимах?
10. Що таке переривання і винятки? В чому полягають їх особливості?
11. Які типи даних підтримують 32-розрядні процесори?
12. Що таке захищений режим? Яка основна мета цього режиму?
13. Що характеризують дескриптори?
14. Які типи таблиць дескрипторів Ви знаєте?
15. Як організована система привілей в 32-розрядних процесорах?
16. Дайте порівняльну характеристику процесорів сімейства x86.

Таблиця 5.2.

Порівняльна характеристика процесорів сімейства x86

Тип	Перший випуск	Кільк. транз.	Розр. зовн. ШД	Частота ядра, МГц	Новинки архітектури
8086	1978	29000	16	5, 8, 10	Захищений режим, віртуальна пам'ять до 1 Гбайта, механізм переключення задач, математичний співпроцесор
80286	1982	134000	16	12,5	
80386	1985	275000	32	20	Режим V86, сторінкове керування пам'яттю, нові реєстри, нові 32-бітові операції, знято обмеження на довжину сегмента пам'яті
80486	1989	1,2 млн.	32	25, 40, 100	Вмонтований математичний співпроцесор, внутрішній кеш першого рівня, RISC-ядро, пакетні цикли локальної шини, нові інструкції, суттєве скорочення часу виконання інструкцій, збільшена черга інструкцій
Pentium	1993	3,1 млн.	64	60, 66, 75, 90, 100, 120, 133, 150, 166, 200	Суперскалярна архітектура (за один такт дві інструкції), підтримка мультипроцесорних конфігурацій, підвищення продуктивності математичного співпроцесора в 2-10 раз, внутрішній контроль паритету і шини адреси, конвеєрна адресація шинних циклів, оперування сторінками розміром 4 Мб в режимі сторінкової перекладачі
Pentium Pro	1995	5,5 млн.	64	150, 166, 180, 200	Нова конструкція корпусу, рознесена архітектура, динамічне виконання команд, подвійна незалежна шина
Pentium MMX	1997	4,5 млн.	64	166, 200, 233	Мультимедійні розширення команд, подвоєні обсяги кеш-пам'яті даних і команд, розширена конвеєризація, паралельна обробка групі операндів однією інструкцією
Pentium II	1997	?	64	233, 266, 300, 333	Більш дешева кеш-пам'ять 2-го рівня, розширений кеш 1-го рівня, подання архітектури Pentium Pro з технологією MMX, нова технологія корпусу

6. ОРГАНІЗАЦІЯ ВВЕДЕННЯ-ВИВЕДЕННЯ В ЕОМ

Введенням інформації в обчислювальних системах називають процес передачі інформації від зовнішнього по відношенню до ядра обчислювальної системи джерела інформації в основну (оперативну) пам'ять системи. Виведення інформації – це процес передачі інформації з оперативної пам'яті системи до зовнішнього приймача інформації, наприклад, в іншу систему або на зовнішній носій інформації. Введення-виведення інформації (ВВ) здійснюється за допомогою пристроїв введення-виведення (ПВВ). ПВВ можуть здійснювати перетворення інформації з коду даної машини до потрібної форми. В спеціалізованих ЕОМ, пов'язаних з реальними об'єктами керування, пристрої введення перетворюють фізичні величини в коди даної ЕОМ, а пристрої виведення – перетворюють результати обчислень в форму, зручну для подальшого використання, наприклад, для керування яким-небудь регулятором в системі керування реальним об'єктом.

В сучасних ЕОМ використовуються різні ПВВ – клавіатури, "миші", джойстики, трекболи, пристрої CD-ROM, накопичувачі на гнучких і жорстких магнітних дисках, монітори, принтери, плотери, стримери та інші пристрої. Вибір черговості обслуговування ПВВ визначається особливостями їх функціонування. За властивостями ПВВ можна розділити на дві групи: ті, які можуть знаходитись в режимі чекання обслуговування, і ті, що не допускають чекання обслуговування. Вищий пріоритет на обслуговування отримують пристрої, що не допускають чекання обслуговування. Окрім того, швидкодіючі ПВВ мають пріоритет перед повільнодіючими.

Основні режими введення-виведення

В ЕОМ використовується три основних режими введення-виведення:

- програмне ВВ (програмно-кероване або нефорсоване);
- за перериваннями (форсоване ВВ);
- прямий доступ до пам'яті.

При програмному ВВ периферійні пристрої пасивні. Вони тільки сигналізують про свій стан. Всі дії по ВВ реалізуються командами прикладної програми. При цьому в кожний момент часу може виконуватись тільки одна команда, яка визначає або обчислювальну операцію, або операцію ВВ. Розрізняють синхронне ВВ для завжди готових ПП і асинхронне ВВ з опитуванням готовності ПП. При асинхронному ВВ процесор перевіряє ознаку готовності за допомогою однієї або декількох команд. Якщо ознака встановлена, ініціюється власне ВВ одного або декількох слів даних. Коли ж ознака не встановлена, процесор виконує ще декілька команд з повторною перевіркою ознаки готовності. Це продовжується до тих пір, поки пристрій не буде готовим до операцій ВВ. Даний цикл називають циклом очікування. Основний

недолік програмного ВВ пов'язаний з непродуктивними втратами часу в циклах очікування. Достоїнством програмного ВВ є простота його реалізації, яка не потребує додаткових апаратних засобів.

Введення-виведення в режимі переривань

В цьому режимі непродуктивні втрати часу процесора в циклах очікування різко скорочуються. ПП стають активними. Кожний ПП може посилати в процесор сигнал запиту переривання (або запиту на обслуговування), коли він готовий до операцій ВВ, тобто потребує негайної реакції процесора. Оскільки заздалегідь невідомо, в якій точці програми і які ПП ініціюють переривання, безпосередньо в програмі команди ВВ використати не можна.

Послідовність реакції процесора на сигнал запиту переривання така:

- ПП генерує сигнал запиту переривання;
- процесор, завершуючи виконання поточної команди, і, якщо переривання дозволені (не замасковані), формує сигнал підтвердження переривання;
- здійснюється запам'ятовування активних регістрів процесора в стеку, причому стан програмного лічильника звичайно запам'ятовується автоматично;
- процесор ідентифікує ПП для переходу до відповідної підпрограми обслуговування;
- виконується підпрограма обслуговування переривання;
- відновлюється стан перерваної програми;
- відновлюється виконання перерваної програми.

Система переривань називається багаторівневою, якщо процесор має декілька ліній (рівнів запитів переривання) і якщо підпрограми обслуговування одного рівня можуть перериватись запитами на іншому рівні. Переривання підпрограм обслуговування переривань називаються вкладеними перериваннями. Для організації вкладених переривань в мікропроцесорних комплексах ВІС наявні спеціальні ВІС пріоритетних переривань (К589 ІК14, КР580ВН59), основними функціями яких є обробка пріоритетів запитів переривань, синхронізація моментів початку їх обробки і формування вектора переривання. Розрізняють абсолютний і відносний пріоритети. Запит, який має абсолютний пріоритет, перериває програму, що виконується. Запит з відносним пріоритетом є першим кандидатом на обслуговування після завершення виконання поточної програми. При векторному перериванні вилучається опитування джерел переривання. Переривання називають векторним, якщо джерело переривання, виставляючи запит, посилас в процесор код адреси в пам'яті свого вектора переривання (початкову адресу підпрограми обслуговування переривання).

До основних характеристик систем переривання відносяться:

- загальна кількість запитів переривання (входів в систему переривань);
- час реакції - проміжок часу між появою запиту і початком виконання

- підпрограми обслуговування переривання;
- витрати часу на переключення програм;
- глибина переривання – максимальна кількість програм, що можуть переривати одна одну;
- насичення системи переривань – якщо запит буде не обслуженим до моменту приходу нового запиту від того ж джерела;
- допустимі моменти переривання програм;
- кількість класів (рівнів) переривання.

Основним недоліком ВВ в режимі переривань є витрати часу на переключення програм.

Прямий доступ до пам'яті

Прямим доступом до пам'яті (ПДП, DMA – Direct Memory Access) називають спосіб обміну даними, який забезпечує автономно від процесора встановлення зв'язку і передачу даних між ОП і ПП. В режимі прямого доступу вилучаються витрати часу на переключення програм. Процесор вивільняється від керування операціями ВВ, під час обміну даними між ОП і ПП при певних умовах він може реалізовувати свою програму, незалежно від обміну даними. При прямому доступі обмін даними здійснюється зі швидкістю, яка обмежується тільки пропускною здатністю ОП або ПП. Таким чином, ПДП, розвантажуючи процесор від обслуговування операцій ВВ, сприяє підвищенню загальної продуктивності ЕОМ. Підвищення граничної швидкості ВВ робить ЕОМ більш пристосованою для роботи в системах реального часу.

Прямим доступом до пам'яті керує контролер ПДП, який виконує такі функції:

- керування передачею даних між ОП і ПП;
- задання розміру блока даних, який потрібно передати, і області пам'яті, що використовується при передачі;
- формування адрес комірок ОП, які беруть участь в передачі;
- підрахунок кількості одиниць даних (байт, слів), які передаються, і визначення моменту завершення операції ВВ.

Вказані функції реалізуються контролером ПДП звичайно за допомогою одного або декількох буферних регістрів, регістра – лічильника поточної адреси даних і поточного лічильника даних. Узагальнені структури з використанням контролера ПДП приведені на рис. 6.1.

Контролер ПДП звичайно має більш високий пріоритет щодо зайняття циклу пам'яті порівняно з процесором. Керування пам'яттю переходить до контролера ПДП, як тільки закінчиться цикл її роботи для поточної команди процесора. Для організації вкладених переривань в мікропроцесорних комплектах ВІС наявні спеціальні ВІС контролерів ПДП (КР580ВТ57, 8237А).

ПДП забезпечує високу швидкість обміну даними за рахунок того, що керування обміном здійснюється апаратними засобами.

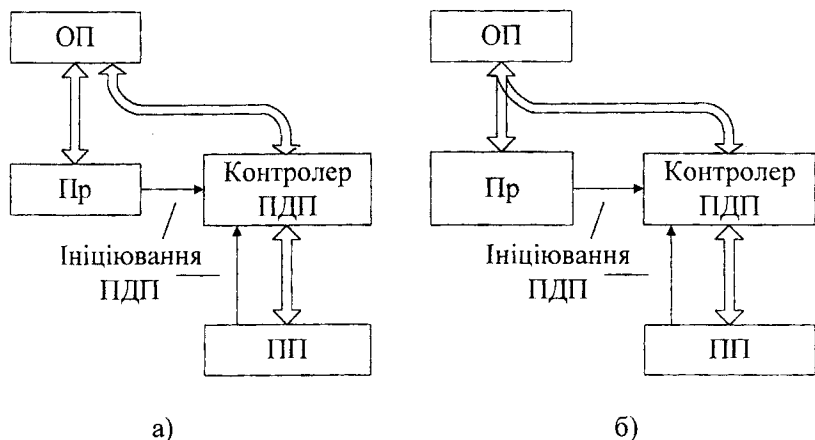


Рис. 6.1. Узагальнені структури з використанням контролера ПДП

Режими і методи передачі інформації

Розрізняють такі режими обміну даними: сімплесний, напівдуплексний, дуплексний і мультиплексний (рис. 6.2). Вони задають можливі напрями і суміщення в часі передач між функціональними блоками Б. В сімплесному режимі передача по інтерфейсу здійснюється тільки одним абонентом від пристрою, що передає, до пристрою, що приймає. В напівдуплексному режимі будь-який із абонентів може виконувати передачу по одних і тих же двонаправлених лініях, якщо лінія зв'язку вільна. В дуплексному режимі будь-який із абонентів може виконувати передачу в довільний момент часу по роздільних лініях зв'язку. Мультиплексний режим передбачає в кожний момент часу здійснення передач між парою абонентів в будь-якому напрямі по лініях зв'язку, що є спільними для багатьох абонентів.

Розрізняють два методи передачі дискретних сигналів: синхронний і асинхронний.

При синхронному методі (рис. 6.3а) передавач П1 встановлює один із двох можливих станів сигналу (0 або 1) і підтримує його протягом певного заздалегідь вибраного часу, після закінчення якого стан сигналу передавачем може бути змінений. Час передачі сигналу, який складається з часу розповсюдження сигналу по лінії L_0 і часу розпізнавання і фіксації сигналу в реєстрі приймального пристрою П2 залежить від параметрів лінії зв'язку і характеристик приймального і передавального пристроїв. Для періоду синхронної передачі τ повинна виконуватись умова $\tau \geq T$, де T – максимальний час передачі сигналів з врахуванням можливих найгірших умов передачі.

При асинхронній передачі (рис. 6.3б) пристрій П1 встановлює на лінії L_0 стан сигналу, що відповідає коду, який передається, а пристрій П2

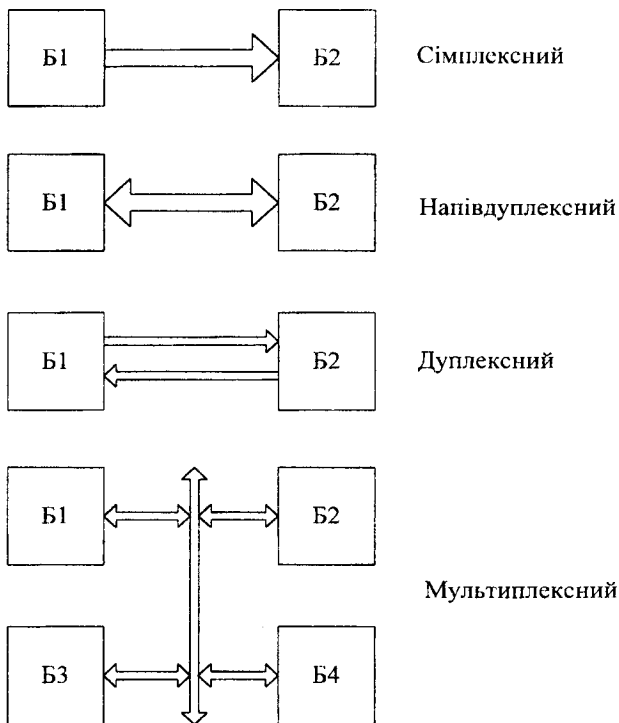


Рис. 6.2. Режими обміну даними

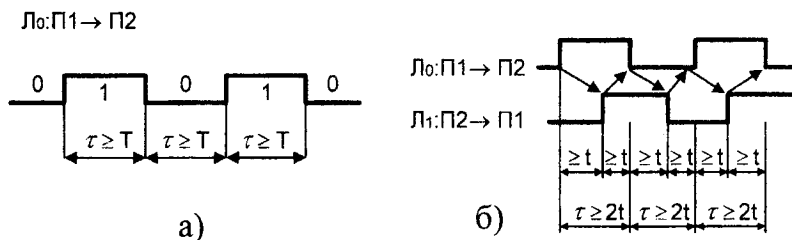


Рис. 6.3. Методи передачі даних: а) – синхронний, б) - асинхронний

після прийому нового стану сигналу інформує про це пристрій П1 зміною стану сигналу на лінії Л₁. Передавач, отримавши сигнал про прийом, знімає сигнал, що передається. Таким чином, період τ , протягом якого передавач повинен підтримувати стан сигналу, є змінним і залежить від характеристик ліній передач.

При передачі паралельного коду по паралельних лініях сигнали

надійдуть до прийомного пристрою в різний час в зв'язку з розкидом параметрів ліній інтерфейсу і елементів, які формують сигнали. Для передачі паралельного коду по декількох лініях розрізняють передачу зі стробуванням, яка використовує синхронний метод, або передачу з підтвердженням, при якій використовується асинхронний метод. Передача зі стробуванням використовується головним чином для пересилань всередині пристрою, наприклад між регістрами. В цьому випадку ще в процесі проектування пристрою для передачі інформації можна вибрати такі моменти часу, коли елементи, які беруть участь в передачі, вільні і готові до цієї операції. Передача з підтвердженням звичайно використовується, коли приймальний пристрій не завжди готовий до прийому інформації (зайнятий виконанням інших операцій).

Інтерфейси ЕОМ

В широкому розумінні інтерфейсом називають сукупність різних засобів, що забезпечують взаємодію пристроїв або систем. Стосовно до області обчислювальної техніки інтерфейсом називають сукупність уніфікованих апаратних, програмних і конструктивних засобів, які забезпечують встановлення зв'язків і взаємодію різних функціональних пристроїв ЕОМ і систем.

В складі ЕОМ або системи є декілька інтерфейсів, в простому випадку – два: внутрішній або машинний і інтерфейс зовнішніх пристроїв. Внутрішній інтерфейс дозволяє підключити до процесора різну кількість блоків пам'яті, а також периферійні пристрої через спеціальні інтерфейсні карти (адаптери). Такий інтерфейс називають магістральним. Зовнішній інтерфейс периферійних пристроїв, або інтерфейс ВВ забезпечує підключення до ЕОМ значної кількості ПП через контролери ВВ. Системний або міжмашинний інтерфейс забезпечує з'єднання ЕОМ між собою в багатомашинних комплексах і мережах за допомогою контролерів системного інтерфейсу і апаратури передачі даних.

Основні функції інтерфейсу полягають в забезпеченні інформаційної, електричної і конструктивної сумісності між функціональними елементами системи. Інформаційна сумісність – це узгодженість взаємодії функціональних елементів систем в відповідності з сукупністю логічних умов, які визначають структуру і склад шин, набір процедур з реалізації взаємодії і послідовність їх виконання, спосіб кодування і формати даних, команд, адресної інформації, часові співвідношення між керуючими сигналами. Електрична сумісність – це узгодженість статичних і динамічних параметрів електричних сигналів в системі шин з врахуванням обмежень на просторове розміщення пристроїв інтерфейсу і технічну реалізацію приймальних і передавальних елементів. Під конструктивною сумісністю розуміють узгодженість конструктивних елементів інтерфейсу, призначених для забезпечення механічного контакту електричних з'єднань і механічної заміни схемних елементів, блоків і вузлів.

При побудові інтерфейсів враховують чотири основних взаємопов'язаних принципи: груповий, агрегування, уніфікації та взаємозамінності. Принцип групового проектування полягає в створенні ряду (сімейства) функціонально і конструктивно подібних пристроїв певного призначення. Принцип агрегування (модульної побудови) полягає в раціональному розподілі системи на сукупність більш простих функціонально і конструктивно закінчених блоків (модулів) з метою поліпшення їх технічних характеристик, високопродуктивних способів виробництва і обслуговування. Принцип уніфікації полягає в мінімізації номенклатури складових вузлів, блоків, модулів, зв'язків між ними при умові раціонального компонування і ефективного функціонування пристрою або системи. Принцип взаємозамінності оснований на здатності модуля виконувати в пристрої установочні функції без додаткової конструкторської доробки. Взаємозамінність визначає міру універсальності пристрою і є результатом процесу уніфікації.

Тип інтерфейсу залежить від складу і типу ліній зв'язку, формату даних, що передаються, команд і послідовності операцій встановлення зв'язку і передачі даних, параметрів електричних сигналів і вимог до них, передавальних і приймальних електронних схем для узгодження сигналів і їх характеристик, конструктивних рішень з'єднувальних засобів і т. п. Існують різні класифікації інтерфейсів за тими чи іншими ознаками.

В залежності від типу організації і топології зв'язків розрізняють лінійні (магістральні), радіальні, деревовидні, ланцюжкові, довільні та інші інтерфейси. В системі з магістральною структурою є колективні шини, до яких під'єднуються всі пристрої (П). В кожний момент часу тільки один із пристроїв може бути активним (здійснити захват шин магістралі). В системах з радіальним інтерфейсом центральний пристрій (ЦП) пов'язаний з кожним з підлеглих пристроїв за допомогою індивідуальної групи шин. Ланцюжковий тип інтерфейсу передбачає послідовне з'єднання пристроїв. В залежності від способу передачі кодів даних інтерфейси бувають паралельні, послідовні і паралельно-послідовні.

Взаємодія між пристроями системи здійснюється сигналами, які передаються по лініях інтерфейсу (електричних або оптичних). Сукупність ліній, які згруповані за функціональним призначенням, називають шиною інтерфейсу. Розрізняють такі шини інтерфейсу (і відповідно групи сигналів): адресні, інформаційні, керуючі, ідентифікації, командні, сповіщення, арбітражу, додаткові. Часто в інтерфейсах використовують розподіл в часі функцій одних і тих же шин (мультиплексування шин), що скорочує кількість ліній в інтерфейсі.

При організації зв'язку групи пристроїв виникає необхідність в адресації та ідентифікації пристроїв. Адресація полягає в виборі центральним пристроєм одного з інших пристроїв, а ідентифікація – в визначенні центральним пристроєм, який із пристроїв подав запит на зв'язок. Адресація і ідентифікація здійснюється шляхом передачі відповідної інформації по лініях інтерфейсу. Розрізняють індивідуальні,

колективні та комбіновані структури ліній і шин інтерфейсу. Індивідуальні шини забезпечують кращу надійність, при їх використанні спрощена адресація та ідентифікація, але потрібна відносно велика кількість обладнання.

Важливим параметром для інтерфейсів є пропускна здатність, яка визначає кількість інформації, що передається по шині інтерфейсу за одиницю часу (секунду). Інший важливий параметр – допустиме віддалення пристроїв, яке обмежується частотними властивостями кабелів і завадостійкістю інтерфейсу. Важливим є також наявність гальванічної розв'язки.

Зовнішні інтерфейси сучасних ЕОМ

Для підключення різних адаптерів ПП використовуються шини розширення. В архітектурі PC найбільш поширеними є шини ISA, EISA, PCI, AGP, VLB. Шини розширення конструктивно оформлені в вигляді щілинних роз'ємів (слотів) для установки плат адаптерів.

ISA Bus (Industry Standard Architecture) - шина розширення, яка використовувалась в перших моделях PC і стала промисловим стандартом. В комп'ютерах XT використовувалась шина з розрядністю даних 8 біт і адреси 20 біт. В комп'ютерах AT шину розширили до 16 біт даних і 24 біт адреси. Конструктивно шина виконана в вигляді двох слотів. Логічно еквівалентною шині ISA є шина PC-104, призначена для побудови невеликих вмонтованих контролерів.

EISA Bus (Extended ISA) – жорстко стандартизоване розширення ISA до 32 біт. Конструктивне виконання забезпечує сумісність з нею і звичайних ISA –адаптерів. Розширення шини стосується не тільки збільшення розрядності даних і адреси: для режимів EISA використовуються додаткові керуючі сигнали, які забезпечують можливість використання більш ефективних режимів передачі. В звичайному (не пакетному) режимі передачі за кожен пару тактів може бути передано до 32 біт даних (один такт на фазу адреси, один – на фазу даних). Максимальну продуктивність шини реалізує пакетний режим – швидкісне пересилання пакетів даних без вказування поточних адрес всередині пакета. EISA - дорога архітектура, яка використовується в багатозадачних системах, на файл-серверах і там, де потрібно високоефективне розширення шини ВВ. Перед шиною PCI в неї є деяка перевага в кількості слотів, яка для однієї шини PCI не перевищує 4, а у EISA може досягати 8.

Шина MCA (MicroChannel Architecture) – мікроканальна архітектура – була введена для PS/2, починаючи з моделі 50. Вона абсолютно несумісна з ISA/EISA та іншими адаптерами. Склад керуючих сигналів, протокол і архітектура орієнтовані на асинхронне функціонування шини і процесора, що знімає проблеми узгодження швидкостей процесора і ПП. При всій прогресивності архітектури (відносно ISA) шина MCA не користується популярністю через вузькість кола виробників MCA-пристроїв і повної несумісності з ISA-системами. Однак вона ще знаходить використання в потужних файл-серверах, де потрібно

забезпечувати високонадійне продуктивне ВВ.

VLB (VESA Local Bus) – стандартизована 32-бітна локальна шина, яка практично являє собою системну шину процесора 486. Шину звичайно використовували для підключення графічного адаптера і контролера дисків. Подальшого розвитку в нових процесорах шина не має.

PCI (Peripheral Component Interconnect) local bus – шина з'єднання периферійних компонентів. Називаючись локальною, ця шина займає особливе місце в сучасній PC-архітектурі, будучи мостом між системною шиною процесора і шиною ВВ ISA/EISA або MCA. Шина PCI є стандартизованою високопродуктивною і надійною шиною розширення ВВ. Шина є синхронною – фіксація всіх сигналів здійснюється по передньому фронту сигналу CLK. В кожній транзакції (обміні по шині) бере участь два пристрої – ініціатор обміну і цільовий пристрій. Шина PCI всі транзакції трактує як пакетні. На одній шині PCI може бути не більше чотирьох пристроїв. Для підключення її до інших шин використовуються спеціальні апаратні засоби – мости шини PCI. Головний міст використовується для підключення PCI до системної шини (шини процесора), а одноранговий міст – для з'єднання двох шин PCI. Спеціальні мости використовуються для підключення шин ISA/EISA. Кожний міст програмується – йому вказуються діапазони адрес просторів пам'яті і ВВ, які відведені абонентам його шин.

Магістральний інтерфейс AGP (Accelerated Graphic Port – прискорений графічний порт) - новий стандарт підключення графічних адаптерів, розроблений на базі шини PCI 2.1. "Прискореність" порту забезпечується конвеєризацією операцій звертання до пам'яті, подвоєними передачами даних та демультимплексуванням шин адреси і даних.

Організація PCMCIA (Personal Computer Memory Card International Association) - міжнародна асоціація виробників карт пам'яті для персональних комп'ютерів) ввела ряд стандартів на шини розширення блокнотних комп'ютерів. Перший з них називався PCIMCIA, а потім був переіменований в стандарт PC Card. Шина PC Card дозволяє підключати розширювачі пам'яті, модеми, контролери дисків і стримерів, мережеві адаптери та інші пристрої. Розрізняють 4 типи PC Card. Вони електрично ідентичні і розрізняються за габаритами та сумісні знизу догори (менші адаптери встають в більші гнізда). Всі пристрої PC Card мають мінімальне енергоспоживання. Існують передумови для введення шини PC Card як додаткової і в настільні PC.

Шина USB (Universal Serial Bus) була розроблена фірмами-лідерами комп'ютерної та телекомунікаційної промисловості - Compaq, DEC, IBM, Microsoft, NEC і Northern Telecom – для підключення ПП зовні корпусу PC. До комп'ютерів, які мають шину USB, можна підключати ПП (клавіатуру, мишу, джойстик, принтер), не виключаючи живлення. Всі ПП повинні бути обладнаними роз'ємами USB і підключатись до PC через окремий винесений блок, який називають USB –хабом або концентратором, і за допомогою якого можна підключити до 127 ПП.

Основні характеристики розповсюджених шин розширення приведені в табл.6.1, а склад сигналів, параметри, властивості, особливості

Характеристики шин розширення

Шина	Пропускна здатність, Мбайт/с	Канали DMA	Розрядність даних	Розрядність адреси	Частота, МГц
ISA-8	4	3	8	20	8
ISA-16	8/(16)	8	16	24	8/(16)
EISA	33,3	8	32	32	8,33
MCA-16	16	-	16	24	10
MCA-32	20	-	32	32	10
VLB	132	-	32/64	32	33-50(66)
PCI	132/264	-	32/64	32	33(66)
PCMCIA	-	-	16	26	33

Зовнішні інтерфейси дозволяють розширювати функціональні можливості комп'ютера шляхом підключення до нього різних ПП і забезпечення комунікації з іншими комп'ютерами. З традиційних інтерфейсів найбільш поширеними є LPT-, COM-, GAME-, MIDI-порти.

Порт паралельного інтерфейсу був введений в PC для підключення принтера – звідси і пішла його назва - LPT-порт (Line PrinTer – порядковий принтер). Адаптер паралельного інтерфейсу являє собою набір регістрів, розміщених в просторі ВВ. Регістри порту адресуються відносно базової адреси, стандартними значеннями якої є 3BCh, 378h, 278h. Порт може використовувати лінію запиту апаратного переривання. Порт має зовнішню 8-бітову шину даних, 5-бітову шину сигналів стану і 4-бітову шину керуючих сигналів. BIOS підтримує до чотирьох LPT-портів (LPT1-LPT4) своїм сервісом – перериванням INT17h, яке забезпечує через них зв'язок з принтерами по інтерфейсу Centronics [4]. Поняття Centronics відноситься як до набору сигналів і протоколу взаємодії, так і до 36-контактного роз'єму, який встановлюється на принтерах.

Керування паралельним портом розділяється на два етапи – попереднє конфігурування (Setup) апаратних засобів і поточне (оперативне) переключення режимів роботи прикладним або системним програмним забезпеченням. Оперативне переключення можливо тільки в межах режимів, які дозволені при конфігуруванні. Спосіб і можливості конфігурування LPT-портів залежать від його виконання і місцеположення. Порт, розміщений на платі розширення, який встановлюється в слот ISA або ISA+ VLB, звичайно конфігурується перемикачами на самій платі. Порт, розміщений на системній платі, звичайно конфігурується через BIOS Setup. Конфігуруванню підлягають параметри: базова адреса, лінія запиту переривання, дозвіл і номер каналу DMA.

Послідовний інтерфейс для передачі даних в один бік використовує

одну сигнальну лінію, по якій інформаційні біти передаються один за одним послідовно. Такий спосіб передачі і визначає назву інтерфейса і порта, який його реалізує: COM-порт (COMmunication Port – послідовний порт). Послідовна передача даних може здійснюватись як в асинхронному, так і в синхронному режимах.

При асинхронній передачі кожному байту передує старт-біт, який сигналізує приймачу про початок чергової посліжки, за яким поступають біти даних і, можливо, біт паритету (контролю парності). Завершує посліжку стоп-біт, який гарантує певну витримку між сусідніми посліжками. Старт-біт наступного байта, який посилається, може посилатись в будь-який момент після закінчення стоп-біту, тобто між передачами можливі паузи довільної тривалості. Старт-біт, який має завжди визначене значення (логічний 0), забезпечує простий механізм синхронізації приймача по сигналу від передавача. Припускається, що приймач і передавач працюють на одній швидкості обміну, яка вимірюється кількістю біт, що передаються, за секунду. Внутрішній генератор синхронізації приймача використовує лічильник-дільник опорної частоти, який встановлюється в нуль в момент прийому початку старт-біта. Цей лічильник генерує внутрішні строби, по яких приймач фіксує біти, що приймаються. В ідеальному випадку ці строби розміщуються в середині бітових інтервалів, що забезпечує можливість прийому даних і при деякій неузгодженості швидкостей приймача і передавача. Формат асинхронної передачі показано на рис. 6.4. Для асинхронного режиму прийнятий ряд стандартних швидкостей обміну : 50, 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 і 115200 біт/сек. Асинхронний обмін в РС реалізується за допомогою COM-порту з використанням протоколу RS-232C.

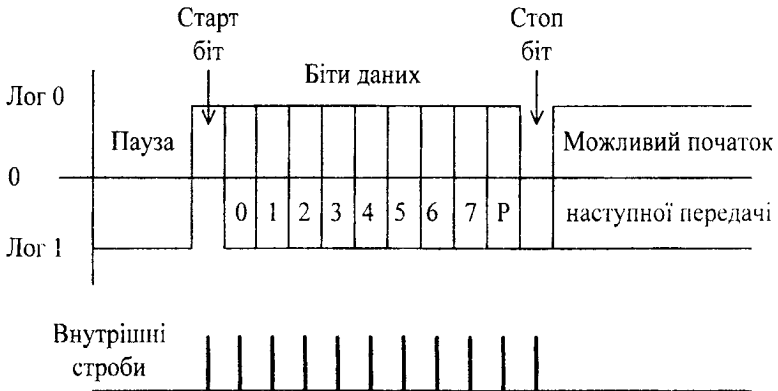


Рис. 6.4. Формат асинхронної передачі
Синхронний режим передачі припускає постійну активність каналу

зв'язку. Спочатку посилається синхробайт, за яким слідує потік інформаційних біт. Якщо у передавача немає даних для передачі, він заповнює паузу безперервним посиланням байтів синхронізації. При передаванні великих масивів даних витрати на синхронізацію в даному режимі обміну будуть нижчі, ніж в асинхронному. Однак в синхронному режимі необхідна зовнішня синхронізація приймача з передавачем, оскільки навіть мале відхилення частот призведе до накопичення похибки і спотворення даних, що приймаються. Зовнішня синхронізація може бути реалізована або за допомогою окремої лінії для передачі сигналу синхронізації, або з використанням спеціального кодування даних, що самосинхронізується, при якому на приймальному боці із прийнятого сигналу можуть бути виділені і імпульси синхронізації.

Послідовний інтерфейс на фізичному рівні може мати різні реалізації, які розрізняються способами передачі електричних сигналів. Для нього існує ряд міжнародних стандартів: RS-232C, RS-423A, RS422A, RS-485. Несиметричні лінії інтерфейсів RS-232C і RS-423A мають найнижчу захищеність від синфазної завади. Кращі параметри має двоточковий інтерфейс RS422A і його магістральний родич RS-485, які працюють на симетричних лініях зв'язку. В них для передачі кожного сигналу використовуються диференціальні сигнали з окремою (крученою) парою провідників.

Розповсюдженим варіантом послідовного інтерфейсу є "струмова петля". В цьому інтерфейсі електричним сигналом є не рівень напруги відносно загального проводу, а струм в двопровідній лінії, яка з'єднує приймач і передавач. Звичайно логічній одиниці (і стану "включено") відповідає протікання струму 20 мА, а логічному нулю – відсутність струму. Це дозволяє виявити стан обірваності лінії – приймач виявить відсутність стоп-бігу (обірваність діє як логічний нуль). Струмова петля звичайно припускає гальванічну розв'язку вхідних кіл приймача (оптрона) від схеми пристрою. При цьому джерелом струму в петлі є передавач (активний передавач). Можливе і живлення від приймача (активний приймач), при цьому вихідний ключ (оптронний) передавача може бути також гальванічно розв'язаним з рештою схеми передавача. Струмова петля з гальванічною розв'язкою дозволяє передавати сигнали на відстань до одиниць кілометрів. Допустима відстань визначається опором пари провідників і рівнем завад.

Цифровий інтерфейс музикальних інструментів MIDI (Musical Instrument Digital Interface) є двонаправленим послідовним асинхронним інтерфейсом з частотою передачі 31,25 Кбіт/сек. В інтерфейсі використовується струмова петля 10 мА (можливо і 5 мА) з гальванічною (оптронною) розв'язкою вхідного кола. Ця розв'язка виключає зв'язок "схемних земель" пристроїв, які з'єднуються через інтерфейсний кабель, що усуває завади (фон), вкрай небажані для звукової техніки.

Інтерфейс ірвового адаптера (GAME-порт) дозволяє вводити

значення дискретних (4 біти) і аналогових сигналів. Спочатку порт був призначений для підключення джойстиків і інших ігрових пристроїв, але він може використовуватись і для підключення інших датчиків. Метод вимірювання опору оснований на програмному визначенні тривалості імпульсу, яка пропорційна величині опору. Звичайно, точність і лінійність перетворення невисока, перетворення виконується не швидко (до 1,12 мс) і сильно завантажує процесор. Однак, на відміну від "справжніх" аналого-цифрових перетворювачів, цей перетворювач дістається задарма – ігровий адаптер входить до складу практично всіх комбінованих плат послідовних і паралельних портів і звукових плат.

Контрольні запитання і задачі

1. Що називають введенням і виведенням інформації в обчислювальних системах?
2. Які пристрої введення і виведення Ви знаєте? Приведіть їх характеристики.
3. Назвіть основні режими введення-виведення.
4. Охарактеризуйте програмне введення-виведення.
5. Яка послідовність реакції процесора на сигнал запиту переривання?
6. Що таке багаторівнева система переривань?
7. Які переривання називають вкладеними?
8. Що таке вектор переривання?
9. Назвіть основні характеристики систем переривання.
10. Що таке прямий доступ до пам'яті?
11. Які функції виконує контролер ПДП?
12. Який режим введення-виведення забезпечує найбільш високу швидкодію?
13. Які режими обміну даними Ви знаєте? Охарактеризуйте їх.
14. Охарактеризуйте синхронний та асинхронний методи передачі інформації.
15. Що називають інтерфейсом ЕОМ? Які бувають інтерфейси?
16. В чому полягають основні функції інтерфейсу?
17. Які принципи враховують при побудові інтерфейсів?
18. Що таке шини інтерфейсу? Які типи шин інтерфейсу Ви знаєте?
19. В чому полягає адресація та ідентифікація пристроїв?
20. Що таке пропускна здатність інтерфейсу?
21. Які шини розширення найбільш поширені в архітектурі РС? Охарактеризуйте їх.
22. Які типи зовнішніх інтерфейсів найбільш поширені в архітектурі сучасних ЕОМ?
23. Як здійснюється асинхронна передача через послідовний порт?
24. Охарактеризуйте варіант послідовного інтерфейсу типу "струмова петля".

7. ПАРАЛЕЛЬНІ ОБЧИСЛЮВАЛЬНІ СИСТЕМИ

Основні принципи організації паралелізму

Одним з ефективних напрямів підвищення продуктивності ЕОМ є розпаралелювання процесів обробки, введення та виведення інформації. При цьому використовуються різні принципи організації паралелізму: суміщення в часі різних етапів розв'язання декількох задач, одночасна обробка інформації багатьма пристроями, конвеєрна обробка інформації та інші.

Суміщення в часі різних етапів розв'язання декількох задач досягається використанням режиму мультипрограмної обробки інформації, який можливий і в однопроцесорній ЕОМ. В наш час такий режим широко застосовується в сучасних комп'ютерах. При цьому використовуються переривання. В потужних і супер-ЕОМ для організації введення-виведення використовуються спеціальні периферійні процесори – канали. В персональних ЕОМ часто діє принцип інтегрованих каналів, коли частину функцій каналу бере на себе процесор. На час реалізації цих функцій він зупиняє обробку інформації. Для забезпечення суміщення операцій застосовують:

- випереджальний перегляд команд і даних з відповідною вибіркою із пам'яті і попередньою підготовкою до виконання операцій;
- організацію асинхронної роботи пристроїв ЕОМ завдяки наявності в них місцевих пристроїв керування;
- конвеєрні принципи організації функціонування ЕОМ;
- реалізацію суміщених звертань до пам'яті за допомогою багатоканальної ОІ;
- суміщення в часі звертань до пам'яті на основі розшарування звертань.

При розшаруванні звертань користуються методом чередування адрес, відповідно до якого суміжні дані знаходяться в різних блоках. Зменшення середнього часу доступу досягається завдяки одночасній роботі кількох блоків пам'яті.

В однопроцесорній ЕОМ при мультипрограмному режимі роботи припускається, що одночасно в ОП присутні декілька програм (або їх фрагментів), які можуть знаходитись в активному стані чи стані готовності до обробки або чекання деякої події, наприклад завершення операції введення-виведення або звільнення потрібного ресурсу. Програма знаходиться в стані готовності, якщо їй виділені всі необхідні ресурси, окрім часу процесора.

Мультипрограмування призначене для підвищення пропускної здатності обчислювального пристрою шляхом більш рівномірного і більш щільного завантаження всього його обладнання, в першу чергу процесора. При цьому швидкість роботи самого процесора і номінальна продуктивність ЕОМ не залежать від мультипрограмування. Суттєвою характеристикою для користувача є пропускна здатність, яка оцінюється

середнім обсягом обчислень, що виконує система в одиницю часу при вирішенні наборів практичних задач. Оскільки програми, які беруть участь в мультипрограμній обробці, конкурують за отримання часу процесора, доступу до пам'яті і інших пристроїв, між ними повинні бути встановлені пріоритетні співвідношення. Відмітимо, що переключення програм із стану готовності в активний стан пов'язано з деякими додатковими витратами часу на роботу керуючої програми. Практично за рахунок мультипрограμмування вдається підвищити пропускну здатність в 4–5 раз.

Одночасну обробку інформації можна забезпечити при наявності багатьох пристроїв. При цьому розрізняють натуральний паралелізм незалежних задач і паралелізм незалежних гілок. Суть натурального паралелізму незалежних задач полягає в тому, що в систему надходить потік незв'язаних між собою задач, коли розв'язання однієї задачі не залежить від інших. Паралелізм незалежних гілок використовується при розв'язанні великих задач, у яких виділяються незалежні частини – гілки програми, що при наявності кількох пристроїв обробки інформації можуть оброблятися паралельно і незалежно одна від одної. Умови незалежності двох гілок:

- між гілками програми відсутній зв'язок, у тому числі вихідні величини однієї гілки програми не є вхідними для іншої;
- в кожній гілці програми використовується своя область пам'яті;
- незалежність керування в кожній з гілок програми, тобто умова виконання однієї гілки не залежить від результатів або ознак, отриманих при реалізації іншої гілки;
- обидві гілки повинні мати програмну незалежність, тобто повинні виконуватись в відповідності з різними блоками програми.

Паралельна обробка незалежних задач або незалежних гілок реалізується в багатопроеесорних системах, або в системах з багатьма EOM.

Багатопроеесорні системи – основний шлях побудови обчислювальних систем надвисокої продуктивності. При їх створенні виникає багато складних проблем, до яких в першу чергу слід віднести розпаралелювання обчислювального процесу для ефективного завантаження процесорів системи, подолання конфліктів при спробі декількох процесорів використати один і той же ресурс. На основі багатопроеесорності і модульного цринципу побудови інших пристроїв системи можливо створення відмовостійких систем або систем підвищеної живучості. За поширеною термінологією багатопроеесорні системи – це масивні паралельні процесори (massively parallel processors, MPP). MPP є найбільш потужними комп'ютерами у світі. Ці машини об'єднують декілька тисяч ЦП і сотні Гігабайт пам'яті в один комп'ютер і є суперкомп'ютерами. Суперкомп'ютери мають величезні обчислювальні потужності (на рівні TFLOPS) і використовуються для вирішення складних задач, наприклад: моделювання погоди, нових ліків та ін. Для більш

реалістичного моделювання швидко ростуть потреби в обчислювальній потужності.

В системах з багатьма ЕОМ останнім часом є розповсюдженими технології розподілених обчислень (distributed computing). При цьому використовується множина комп'ютерів, пов'язаних за допомогою мережі в одну віртуальну машину для вирішення однієї великої задачі. В зв'язку з тим, що все більше організацій мають швидкодіючі обчислювальні мережі, які зв'язують робочі станції загального призначення, комбінована обчислювальна потужність легко може перевищити продуктивність одного суперкомп'ютера. Важлива перевага розподілених обчислень - їх низька вартість. Великий суперкомп'ютер дорого коштує (більше \$10 млн.), а при використанні розподілених обчислень прямі витрати для користувача дуже малі. Звичайно, коли комп'ютери і мережа вже є, розподілені обчислення дозволяють використовувати більш раціонально устаткування, що вже існує в системі.

Конвеєрний принцип обробки інформації

Термін "конвеєр" використовується в розумінні суміщення операцій в обчислювальній системі шляхом використання в системі деякої базової функції, яка багатократно повторюється, і розподілі цієї функції на декілька підфункцій з такими властивостями:

- виконання базової функції еквівалентно послідовному виконанню цих підфункцій;
- всі вхідні дані для однієї підфункції поступають з виходів попередніх підфункцій в послідовному ланцюжку обробки;
- між підфункціями відсутні інші взаємозв'язки, окрім обміну інформацією між виходами і входами;
- для реалізації кожної підфункції є відповідні апаратні блоки;
- час, потрібний цим блокам на виконання їх індивідуальних обчислень, як правило, приблизно однаковий.

Апаратні блоки для реалізації підфункцій називають каскадами. Між каскадами розміщують елементи пам'яті, щоб застерегти від пробігу даних і їх спотворенню в наступному каскаді до того, як закінчився поточний такт роботи.

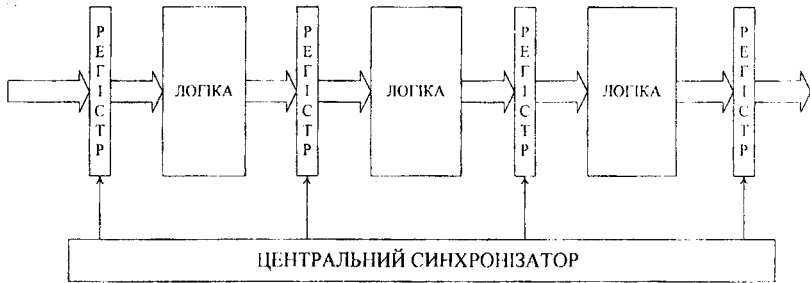
Розрізняють конвеєрні структури таких типів:

- однофункціональний конвеєр – виконує тільки одну базисну функцію;
- багатфункціональний конвеєр – виконує декілька різних функцій;
- конвеєр зі статичною конфігурацією – здатний на відносно рідкі зміни типів функцій;
- векторний процесор – зміна конфігурації здійснюється безпосередньо під програмним керуванням;
- конвеєр з динамічним настроюванням – допускає більш часті зміни функцій аж до змін функції для кожного набору.

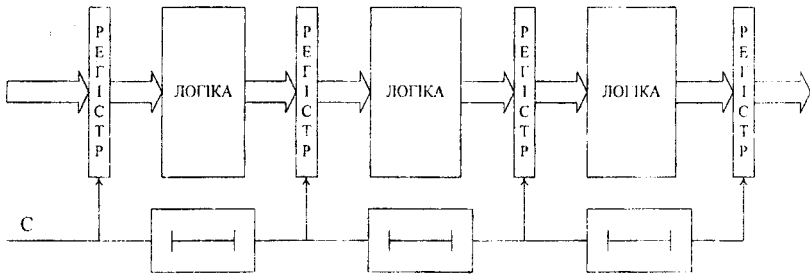
В конвеєрах спостерігаються явища змагань двох типів: структурні і ті, що залежать від даних. Структурними називають змагання, коли дві

різних порції даних намагаються використати один і той же каскад в один і той же час – відбувається зіткнення. Змагання, що залежать від даних, спостерігаються тоді, коли те, що відбувається в якомусь каскаді, визначає чи можуть дані проходити через решту каскадів (наприклад, коли два різні каскади поділяють спільну пам'ять).

Конвеєрні системи можуть бути асинхронними і синхронними. При асинхронній конвеєрній обробці використовується змінна тривалість виконання команд, а при синхронній тривалість окремого етапу постійна і вибирається з врахуванням самого тривалого етапу обробки команд. На практиці синхронізація здійснюється з застосуванням або центрального синхронізатора (рис. 7.1а), або "синхронізуючого ланцюжка" (рис. 7.1б).



а)



б)

Рис. 7.1. Синхронізація конвеєра: а) – з центральним синхронізатором; б) – за допомогою "синхронізуючого ланцюжка"

В сучасних комп'ютерах широкого призначення конвеєрний принцип використовується при організації читання команд із пам'яті, організації кеш-пам'яті, перетворення операційної частини команди. В спеціалізованих ЕОМ він використовується в випадках, коли потрібні швидкодіючі багатоетапні перетворення потоків даних.

Однорідні системи і середовища

В основу побудови однорідних систем і середовищ покладено принципи паралельності операцій, змінності логічних структур та однорідності елементів і зв'язків між ними.

Принцип паралельності базується на паралельності задач і алгоритмів. Кожна задача подається у вигляді зв'язаних простих підзадач, які можуть виконуватись одночасно. Виконання кожної з них повинне бути забезпечене відповідним паралельним алгоритмом.

Принцип змінності логічних структур передбачає встановлення зв'язку між складовими системи програмним шляхом. Змінність забезпечує універсальність, гнучкість і живучість системи.

Принцип однорідності полягає в використанні однорідних елементів і зв'язків, що спрощує їх реалізацію.

В однорідних системах всі три розглянуті принципи використовуються разом. Однорідні середовища можуть бути подані у вигляді ґратної структури, кожна клітинка якої з'єднана з чотирма сусідніми. Між сусідніми клітинками проходять інформаційні канали та канали настроювання. Елементи такої структури виконують як логічні, так і комутаційні функції.

Матричні пристрої можна розділити на 4 класи. До першого відносять комбінаційні схеми (ітеративні мережі). Якщо комбінаційну матрицю сумістити з матрицею тригерів і додати необхідні з'єднання, можна отримати другий і третій класи пристроїв – систолічні і асоціативні матриці. Слово "систолічний" пов'язано з поняттям "пульсуючий процес", яке внесено в обчислювальну техніку з фізіології. Для систолічних матриць характерні такі ознаки, як конвеєрність, матрична структура, регулярність зв'язків, введення-виведення через крайні елементи матриці, однократність вибірки даних. Систолічні пристрої можна розділити на декілька рівнів складності, найпростішим з яких є рівень однорозрядних елементів. При обробці векторів систолічні матриці забезпечують високу продуктивність завдяки використанню обох видів паралелізму: просторового і часового (конвеєрності). Невисока структурна складність полегшує автоматизоване проектування систолічних ВІС та НВІС. Функції асоціативної матриці полягають в забезпеченні приймання та передачі ознак, які характеризують зміст даних, причому в процесі проведення пошукових операцій дані нерухомі. Четвертий клас – це асоціативно-систолічна матриця, здатна в залежності від настроювання виконувати операції конвеєрної арифметики та асоціативного пошуку. Приклади побудови пристроїв на основі розглянутих принципів приведені в [2].

Класифікація архітектур систем паралельної обробки

Однією з найбільш поширених різновидів класифікації архітектур обчислювальних систем паралельної обробки є класифікація Фліна, відповідно до якої, в залежності від взаємодії потоку команд і потоку даних (операндів) розрізняють чотири основних класи систем:

- ОКОД (SISD– Single Instruction, Single Data) – системи з одиничним потоком команд і одиничним потоком даних;

- МКОД (MISD) – системи з множинними потоками команд і одиничним потоком даних;
- ОКМД (SIMD – Single Instruction, Multiple Data) – системи з одиничним потоком команд і множинними потоками даних;
- МКМД (MIMD) – системи з множинними потоками команд і множинними потоками даних.

Структури систем згідно з такою класифікацією приведені на рис. 7.2.

В системах класу ОКОД є тільки один пристрій обробки і при виконанні однієї машинної команди формується поодинокий результат. В сучасних системах класу ОКОД використовується паралельна обробка за рахунок суміщення в часі різних етапів розв'язання декількох задач.

В системах класу ОКМД команда, виділена пристроєм керування, одночасно передається множині процесорів з однаковою структурою. Всі процесори одночасно виконують одну і ту ж операцію, але над різними операндами. ПК на основі інформації про стан системи дозволяє або забороняє виконання операції. Виконання дозволяється тільки тим процесорам, у яких наявні певні умови. ПК контролює взаємозв'язок між процесорами і керує звертаннями до загальної пам'яті. При цьому частина процесорів може пропускати конкретні команди, які визначаються за допомогою маскування. На цій основі будуються асоціативні процесори, спеціальні процесори для обробки зображень та ін.

Системи класу МКОД відносять до класу систем з конвеєрною обробкою. Тут на поодинокий потік даних діють кілька потоків команд. Потік даних послідовно проходить через усі або частину спеціалізованих АЛП. Іноді такі системи називають системами з магістральною обробкою інформації. Обчислювальні процеси в системах класу МКОД розбиваються на кілька етапів, кожному з яких відповідає свій процесор. Продуктивність таких систем залежить від можливості і тривалості заповнення магістралі.

Обчислювальні системи класу МКМД є найскладнішими засобами комп'ютерної техніки. Один з варіантів реалізації архітектури МКМД – мультипроцесорні системи, де обробка розпаралелена за рахунок взаємодії множини процесорів. За структурою шинних зв'язків між процесорними елементами МКМД розділяються на системи з однією шиною, з круговою шиною, з матричною комутацією, з багатоступінчатою комутацією та ін.

В залежності від принципу керування обчислювальним процесом розрізняють три групи систем паралельної обробки:

- системи, які керуються командами;
- системи, які керуються даними;
- системи, які керуються запитами.

В системах, які керуються командами, команди виконуються в тому порядку, в якому вони записані в програмі. В системах, які керуються даними, повинні виконуватись всі команди (незалежно від їх місця в програмі), які можуть бути виконані, тобто для яких є в наявності операнди. Іншими словами керування обчислювальним процесом переходить до потоків даних, тому системи з такою архітектурою отримали назву поточкових.

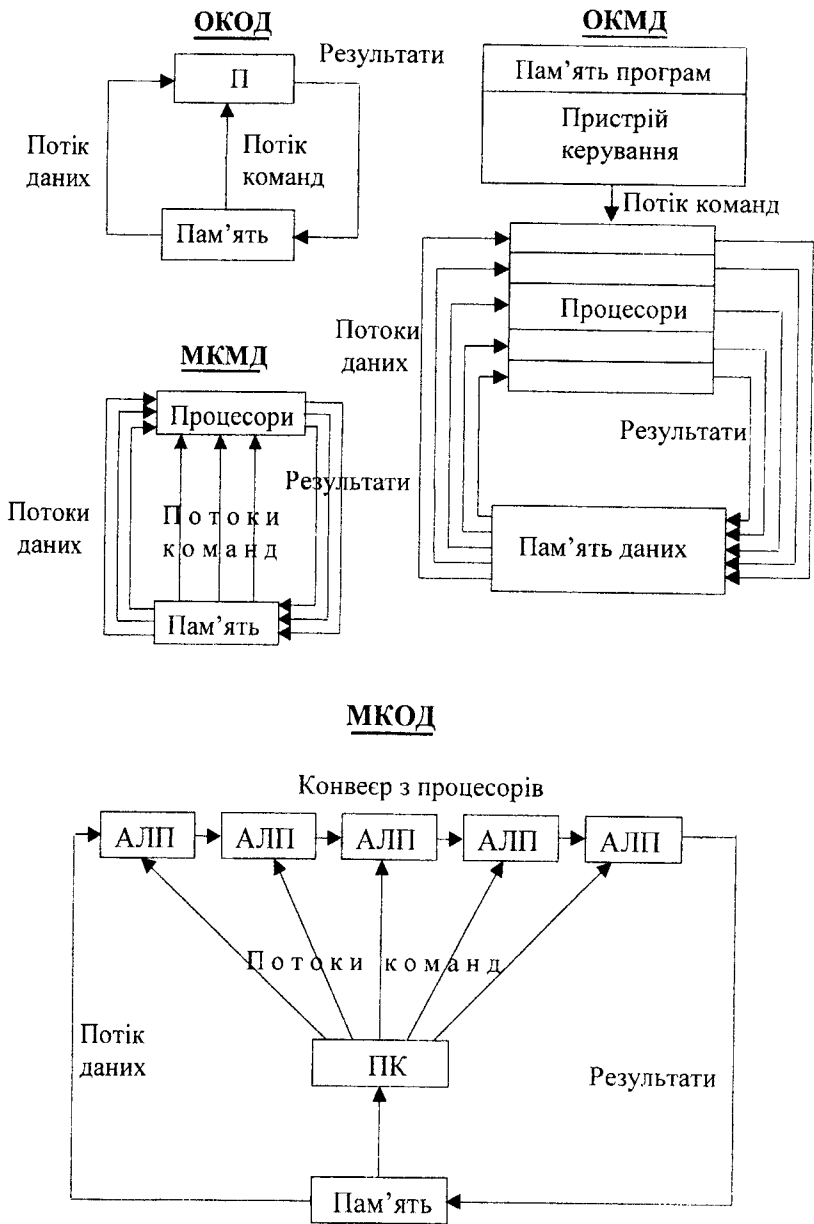


Рис. 7.2. Структури систем паралельної обробки за класифікацією Фліна

В системах, які керуються запитами, виконання будь-якої операції визначається нагальною потребою в її результатах. Такі системи призначені для безпосереднього виконання операцій, записаних на функціональній мові, в зв'язку з чим вони отримали назву редуційних (від англійського reduction - скорочення), оскільки працюють за принципом скорочення виразів тексту програми на функціональній мові.

Ідея потокової машини полягає в створенні ЕОМ, яка здатна одночасно виконувати всі команди, що мають операнди. Для цього вона повинна мати багато процесорних елементів. Важливим принципом архітектури поточкових машин є можливість динамічної реконфігурації, бо в кожний момент часу може бути готовою до виконання різна кількість команд. Реконфігурація може здійснюватись як апаратним способом (так звані ІМ-машини), так і за допомогою програмних засобів – шляхом використання спеціального формату команд (так звані SM-машини). За рахунок можливості реконфігурації потокова ЕОМ, яка є універсальною машиною, може в принципі розв'язувати задачі з тією ж продуктивністю, що й спеціалізовані системи. Найпростішими системами, які керуються даними, є систолічні матриці.

До інших класифікацій систем паралельної обробки відносять класифікації Шора, Хоббсона, Мурте і Бідлсома [13, 17].

Комунікаційні середовища

Конкретно архітектури розповсюджених сучасних комутаційних середовищ розглядаються в [9]. Зупинимося на основних особливостях і властивостях таких середовищ.

При реалізації комунікаційних середовищ використовуються рішення, які стали з часом міжнародними стандартами. Одним з таких стандартів є когерентний інтерфейс, що дозволяє змінювати масштаб, **SCI (Scalable Coherent Interface)**. Він передбачає реалізацію когерентності шляхом використання стандартно організованої кеш-пам'яті, яка розміщується в вузлі SCI. Ця кеш-пам'ять розміщується в інтерфейсі між обчислювальним модулем (ОМ) і вузлом. Якщо в попередніх рівнях ієрархії пам'яті ОМ виявляється відсутність необхідних даних, то здійснюється пошук цих даних в кеш-пам'яті вузла. Якщо дані знайдені, або модифікуються, то по закінченні модифікації вони розміщуються всередині ієрархії пам'яті ОМ. При відсутності даних формується сигнал промаху, який активізує адаптер інтерфейсу на дії по доставці даних від віддалених блоків пам'яті.

При організації комутаційних середовищ на базі SCI кожен вузол має вхідний і вихідний канали. Вузли пов'язані однонаправленими каналами “точка-точка” або з сусіднім вузлом, або під'єднані до комутатора. При об'єднанні вузлів обов'язково повинна формуватися циклічна магістраль (кільце) із вузлів, з'єднаних каналами “точка-точка”, між вхідним і вихідним каналами кожного вузла. Один вузол у кільці, який називається scrubber, виконує функції ініціалізації вузлів кільця зі встановленням адрес, керування таймерами, знищення пакетів, які не знайшли адресата. В кільці може бути тільки один scrubber. Можлива

побудова багатокільцевих систем, зв'язаних через агентів. Агент-комутатор або агент-міст забезпечує з'єднання між різними кільцями або між кільцем і комутаційним середовищем з іншим протоколом. Однонаправленість передач у SCI має принциповий характер завдяки тому, що вона виключає перемикання виводів мікросхем з передачі на прийом і навпаки, що створює електричні завади. SCI вузли захищені від механічних дій і електромагнітних випромінювань та можуть вставлятись і вийматись без відключення живлення.

В SCI прийнята така система позначень каналів: тип<кількість сигналів>-<рід сигналів> - <частота передачі в МГц/сек>. Наприклад, 18-E-500 передає 18 біт паралельно з використанням диференціальних електричних сигналів в передавачах і приймачах і передачею інформації по передньому і задньому фронту на частоті 250 МГц.

SCI використовує 16-бітову магістраль даних, а також лінії для передачі тактового сигналу і ознакового (флагового) біту. Ці зв'язки реалізуються як 18 пар провідників, які несуть диференціальні ЕПЛ-сигнали з пропускнуою здатністю 1 Гбайт/с. Оптиволоконний канал 1-FO-1250 використовується для побітової передачі з пропускнуою здатністю 1,25 Гбіт/с.

Стандарт SCI забезпечує побудову легкого в реалізації та ефективного комутаційного середовища для об'єднання процесорів і блоків пам'яті, для створення розподіленої мережі робочих станцій або для організації введення-виведення супер-ЕОМ, високопродуктивних серверів і робочих станцій на базі сучасних мікропроцесорів.

Комутаційне середовище **MYRINET** стандартизує формат пакета, спосіб адресації ОМ, набір керуючих символів протоколу передачі пакетів. Комутаційне середовище утворюється адаптерами "шина комп'ютера - лінк мережі" і комутаторами лінків мережі. Ніяких вимог до адаптера, окрім власне реалізації протоколу, середовище MYRINET не висуває. Кожен лінк містить пару однонаправлених каналів, утворюючи дуплексний канал. Лінки передають пакети заданої структури, які складаються з заголовка, даних пакета і кінцівки. Мережа MYRINET не повинна блокуватися при окремих неполадках або при наявності пристроїв, до яких не підведена напруга. Якщо лінк підключений до непрацездатного адаптера, то передача не блокується, а пакети, що передаються, втрачаються. Плата адаптера вмонтовується в шину ОМ і під'єднується до живлення комп'ютера. Допускається підключення довільної кількості адаптерів до шини в межах її комунікаційних можливостей. Адаптер має 128 Кбайт пам'яті, яка використовується як для збереження пакетів, так і керуючої програми адаптера, що повинна бути завантажена в процесор адаптера для переведення його в робочий стан.

Комутатори випускаються з 4, 8, 12, та 16 портами, які іменуються 0, 1, 2, ..., k-1. Пакети, які адресуються портам, що не входять в цей діапазон, пропадають. В комутаторі використовується передача пакетів шляхом

встановлення з'єднання на час передачі. З'єднання виконується після прийому заголовка і визначення порту, через який пакет повинен вийти з комутатора. Якщо цей порт вільний, то спочатку по з'єднанню передається заголовок, потім байт за байтом тіла пакета, потім кінцівка, передача якої розриває встановлене в комутаторі з'єднання. Якщо потрібний вихідний порт зайнятий передачею іншого пакета, прийом пакета блокується вхідним портом. Комутатори побудовані на основі двох замовних НВІС: власне НВІС комутатора та інтерфейсної НВІС. Комутатор має власне живлення і може функціонувати автономно.

В середовищі MYRINET передавання символів виконується по 9 провідниках: 8 біт інтерпретуються в залежності від значення дев'ятого біта (1 – байт даних, 0 – керуючий символ). Двійкова "1" кодується зміною сигналу, а "0" – відсутністю зміни. Приймач функціонує асинхронно і може прийняти символ, коли той з'явиться. Тактова частота на сигнальній лінії складає 40 МГц, що відповідає пропускній здатності 80 Мбайт/с.

Комутаційне середовище **Raceway**, створене фірмою Cypress, забезпечує пропускну здатність на рівні 1 Гбайт/с. При організації комутаційного середовища використовуються кристали комутатора Cypress CY7C965 Raceway Crossbar і відповідні адаптери "порт комутатора – шина OM". Адаптери включають інтерфейсні схеми, канал прямого доступу до пам'яті, таймери та інші блоки, необхідні для керування комутатором. Кристал комутатора розрахований на 6 портів з пропускну здатністю кожного порту 160 Мбайт/с. Кожний порт використовує 32 сигнальні лінії даних, 5 сигнальних керуючих ліній і 3 сигнальних керуючих лінії, які є спільними для всіх портів. Тактова частота передачі даних складає 40 МГц. Комутатор може одночасно комутувати три вхідних порти з трьома різними вихідними портами, що забезпечує загальну пропускну здатність 480 Мбайт/с.

Комутаційне середовище на базі Raceway Crossbar при ініціації транзакції попередньо встановлює з'єднання, а потім передає заголовок з двома керуючими 64-розрядними словами, після яких передаються дані 64-розрядними словами (дві посилки по 32 розряди). Слово адреси, яке передається в заголовку, адресує 64-розрядні слова. Таким чином, початок і кінець блока даних, які передаються, повинні бути кратні 8 байтам. Кожен кристал Raceway Crossbar при встановленні з'єднання здійснює затримку передач приблизно на 125 нс. При проходженні сукупності комутаторів затримки додаються. Після того, як шлях передачі встановлений, передача здійснюється з пропускну здатністю пам'яті, але не більшою 160 Мбайт/с. Адаптер розробляється для шини конкретного OM. Він узгоджує формати даних і формує керуючі сигнали, необхідні для передачі даних із пам'яті OM в порт CY7C965 Raceway Crossbar.

Конектор шин PCI **SRC 3266 DE (Sebring Ring Connection for PCI)** призначений для роботи з 32-розрядною шиною PCI на частотах до 66 МГц. Кількість кристалів SRC 3266, які можуть об'єднуватися, досягає

256. Лінки, які об'єднують SRC кристали, мають ширину 16 біт і тактову частоту 266 МГц. Таким чином пікова пропускна здатність лінка складає 532 Мбайт/с. SRC використовує низьковольтні диференційні здатності. Підключені до шин PCI пристрої (ОМ) прозоро обмінюються пакетами по двох транспортних кільцях.

Кожному SRC 3266 при ініціалізації виділяється частина адресного простору 2^{32} шини PCI, яка називається його вікном. Можуть бути сформовані вікна розміром 1, 2, 4, 8, 16 Мбайт для числа шин PCI до 256. Розміри вікон, які більші за вищевказані, накладають обмеження на максимальну кількість шин PCI, які можуть об'єднуватися. Розмір вікна 32 Мбайта може бути при максимальній кількості шин 128, 64 Мбайта – при кількості шин до 64, і 128 Мбайт – при кількості шин до 32.

В SRC 3266 реалізована повністю технологія “Plug and Play” і автоконфігурація засобами BIOS.

Комутаційне середовище **Memory Channel (MC)** розвивається фірмою DEC для побудови кластерних систем. Кожен модуль кластера має MC адаптер, який під'єднується до комутатора. Комутатор реалізує парні з'єднання “точка-точка” або трансляційне з'єднання “точка – всім точкам”. Передачі через MC програмуються як доступ до пам'яті, а не як доступ до зовнішніх пристроїв. Перш ніж пара модулів зможе взаємодіяти через MC, в цих модулях повинен визначитися адресний простір в одному модулі як вхідний, в іншому – як вихідний. Коли модуль виконує ініціалізацію MC, встановлюються порції його локальної пам'яті, які розділяються з іншими модулями. Ці порції мають ту ж гранульованість, що і сторінки пам'яті. Модуль, який створює вхідну сторінку, виділяє сторінку фізичної пам'яті і робить її доступною для розділення з іншими модулями. Модуль, який створює вихідну сторінку, заповнює елемент таблиці в MC, задаючи в кожному елементі адресу модуля, за якою будуть пересилатись дані. Після створення вхідних і вихідних сторінок доступ до віртуальної пам'яті здійснюється командами “запис” і “читання”.

MC складається з адаптера, який вставляється в роз'єм PCI, модуля і комутатора, до якого під'єднуються лінки адаптерів всіх модулів. Максимальна пропускна здатність MC2 складає 132 Мбайта/с. В MC2 використовуються дуплексні лінії довжиною до 10 метрів, по яких передаються низьковольтні диференційні сигнали. В одному напрямі передаються паралельно 16 біт даних, тактовий сигнал і сигнал початку пакета.

Комунікаційні середовища реалізуються також на базі **трансп'ютероподібних мікропроцесорів**. В трансп'ютерах на одному кристалі реалізовані процесор, блок пам'яті і комутаційні канали (лінки) для об'єднання трансп'ютерів в паралельну систему. Лінки трансп'ютерів (OS-лінки) передають дані в кожному напрямі по одному проводу з пропускною здатністю 10 Мбіт/с. На основі трансп'ютерних лінків

розроблений стандарт IEEE P1355, який передбачає передачу даних з швидкостями в діапазоні 10 Мбіт/с – 1 Гбіт/с.

Вибір комутаційного середовища визначається:

- пропускну здатністю шини, до якої під'єднується адаптер;
- пропускну здатністю і вартістю адаптера;
- пропускну здатністю і вартістю мережі комутаторів;
- затримками при встановленні з'єднань;
- комерційною доступністю середовища;
- практичною апробацією середовища в існуючих системах;
- наявністю розвинутого програмного забезпечення адаптера і програмних середовищ, в яких працює адаптери.

Комутатори обчислювальних систем

Комутатори бувають прості і складені, які компонується з простих. Прості комутатори мають невелику затримку, але в силу фізичних обмежень можуть бути реалізовані тільки для систем з невеликою кількістю обчислювачів. В сучасних системах комутатори реалізуються в вигляді ВІС. Наприклад, в системі SP-2 фірми IBM використовується мікросхема, яка реалізує комутатор 4x4 з пропускну здатністю кожного напрямку 80 Мбайт/с. В системі Gray T3D для створення одного комутатора 6x6 на 6 напрямів використовується три однакових спеціально розроблених ВІС, кожна з яких реалізує комутатор 2x2 на 2 напрямки. Один напрям при цьому має пропускну здатність 300 Мбайт/с.

Прості комутатори бувають з часовим і просторовим розділенням. Комутатори з часовим розділенням називаються також шинами. Їх особливість полягає в використанні загальної інформаційної магістралі для передачі даних між пристроями, які підключені до шини. Шини, як правило, містять тільки пасивні елементи, а керування передачами здійснюється пристроями, які підключені до шини. Для вирішення конфліктів при одночасному запиті на передачу від декількох пристроїв використовуються такі прийоми: призначення кожному пристрою унікального пріоритету, постановка запитів в чергу з дисципліною обслуговування FIFO, з'єднання пристроїв в кільце з фіксацією їх взаємної пріоритетності.

Шинні структури мають цілий ряд обмежень, які суттєво впливають на пропускну здатність шини: кількість активних пристроїв, алгоритм арбітражу, централізоване або розподілене керування, ширина шини, механізм обробки збоїв і відмов, спосіб синхронізації та ін. Якою б не була високою швидкість передачі по шині, потрібен час на виконання арбітражу і виростає складність схеми арбітра. Час арбітражу збільшується при збільшенні кількості пристроїв, підключених до шини. Коли шина часто виділяється різним пристроям, час арбітражу суттєво обмежує пропускну здатність. В кожен момент часу шина використовується для передачі тільки одним пристроєм, що створює вузьке місце в системі з багатьма

пристроями. Пропускна здатність шини обмежується її шириною і тактовою частотою роботи шини. Фізичні обмеження, пов'язані з відбиттям сигналів і зміною навантаження при зміні кількості пристроїв, призводять до серйозних проблем.

Прості комутатори з просторовим розділенням здатні з'єднувати будь-який вхід з будь-яким одним виходом (ординарні) або підмножиною виходів (неординарні). Такі комутатори являють собою сукупність мультиплексорів, кількість яких дорівнює кількості виходів комутатора. Кожен вхід повинен поступати на всі мультиплексори. Такі комутатори характеризуються мінімальною затримкою, повною доступністю, але є складними і мають обмеження по кількості виходів комутатора, що пов'язано з можливостями по навантажувальній здатності вхідних сигналів.

Складені комутатори будуються із сукупності простих комутаторів з меншою кількістю входів і виходів шляхом об'єднання цих комутаторів лініями "точка-точка", що знижує вимоги до навантажувальної здатності вхідних сигналів. Апаратні витрати у складеного комутатора менші, ніж у прямокутного комутатора з тією ж кількістю входів і виходів, але вони мають затримку, яка пропорційна кількості каскадів. Найчастіше складені комутатори будуються з прямокутних комутаторів 2×2 з двома входами і двома виходами. Ці комутатори мають два стани: пряма передача з відповідних входів на виходи і перехресна передача. Зі складених комутаторів найбільш поширеними є комутатор Клоза, Баньян-мережі та розподілені складені комутатори [9].

Питання паралельного програмування для MPP систем, сучасні масово-паралельні обчислювальні системи різних виробників, реалізація асоціативних і нейромережових алгоритмів в MPP, питання надійності і продуктивності паралельних систем детально розглянуті в [9].

Контрольні запитання і задачі

2. Які принципи організації паралелізму Ви знаєте?
3. В чому полягає конвеєрний принцип обробки інформації?
4. Назвіть і охарактеризуйте найбільш поширені структури конвеєрів.
5. Що таке однорідні системи і середовища? Охарактеризуйте їх.
6. Охарактеризуйте структури обчислювальних систем паралельної обробки за класифікацією Фліна.
7. Що таке системи, які керуються запитаннями?
8. Що таке системи, які керуються даними?
9. Що таке системи, які керуються командами?
10. Приведіть характеристику сучасних комутаційних середовищ.
11. Для чого призначені комутатори обчислювальних систем? Які вони бувають?
12. В чому полягають обмеження шинних структур?
13. Чим визначається вибір комутаційного середовища?

8. МІКРОПРОЦЕСОРНІ СИСТЕМИ

Перший чіп Intel 4004, створений фірмою Intel в 1971 році, працював на частоті 750 Кгц, містив 2300 транзисторів і коштував біля \$200. Продуктивність його оцінювалася в 60 тис. операцій у секунду. На сьогоднішній день рекордні показники належать мікропроцесорам Alpha 21264 фірми DEC і складають: 600 Мгц, 15,2 млн. транзисторів, 2 млрд. операцій у секунду і біля \$300. У мікропроцесорах – найбільш складних мікроелектронних пристроях – втілені найпередовіші досягнення інженерної думки. В умовах властивої даній галузі виробництва конкуренції і величезних капіталовкладень випуск нової моделі мікропроцесора так чи інакше пов'язаний з черговим науковим, конструкторським, технологічним проривом.

Архітектура сучасних мікропроцесорів

Універсальні мікропроцесори призначаються для застосування в персональних ЕОМ, робочих станціях, а останнім часом і у масово-паралельних супер-ЕОМ. Основною їхньою характеристикою є наявність розвинутих пристроїв для ефективної реалізації операцій над числами з плаваючою комою, над 64 розрядними і більш довгими операндами. Призначаються в основному для проведення науково-технічних розрахунків.

Цифрові сигнальні процесори розраховані на обробку в реальному масштабі часу цифрових потоків, утворених шляхом оцифрування аналогових сигналів. Це обумовлює їх порівняно малу розрядність і переважно цілочисельну обробку. Однак сучасні сигнальні процесори здатні проводити обчислення з плаваючою комою над 32–40–розрядними операндами. Крім того, з'явився клас медійних процесорів, що являють собою закінчені системи для обробки аудіо – і відеоінформації.

Найбільшу спеціалізацію мають **мікроконтролери**, які використовуються у вбудованих системах керування, в тому числі в побутових приладах. Кількість їх типів з різними системами команд перевищує 500.

ПЛІС (програмовані логічні інтегральні схеми) – це матричні великі інтегральні схеми, що дозволяють програмно скомпонувати в одному корпусі електронну схему, еквівалентну схемі, що містить від декількох десятків до декількох сотень інтегральних схем стандартної логіки. В порівнянні з іншими мікроселектронними технологіями, технологія ПЛІС забезпечує рекордно короткий проектно-технологічний цикл (від декількох годин до декількох днів), мінімальні витрати на проектування, максимальну гнучкість при необхідності модифікації апаратури.

В цьому розділі основна увага приділяється універсальним мікропроцесорам у зв'язку з тим, що цифрові сигнальні процесори і мікроконтролери є предметом розгляду дисциплін "Обробка сигналів та зображень", "Методи та засоби комп'ютерних інформаційних технологій",

"Мікропроцесори", а ПЛІС розглядаються в дисципліні "САІР засобів обчислювальної техніки".

В ході еволюційного розвитку архітектур процесорів до складу системи команд вводилися і закріплювалися складні команди, що, на думку розроблювачів, відповідали потрібним задачам. Команди бувають різних типів: "регістр, регістр - регістр", "пам'ять, пам'ять - пам'ять", "регістр - пам'ять" і ін. Складні команди модифікують стан груп регістрів і комірок пам'яті і для їхньої реалізації при прийнятних витратах устаткування, як правило, застосовується мікропрограмування.

Команди називаються скалярними, якщо вхідні операнди і результат є числами (скалярами). Команди називаються векторними, якщо вхідні операнди і, можливо, результат є вектором (масивом) чисел, а для перетворення даних масиву використовується одна векторна команда.

Підвищення продуктивності мікропроцесорів досягається за рахунок збільшення тактової частоти, удосконалювання паралельної і конвеєрної обробки даних, а також зменшення часу доступу до пам'яті. Сучасні мікропроцесори містять десять і більше операційних пристроїв, кожний з яких являє собою конвеєр. Ефективне завантаження конвеєрів, що функціонують паралельно, забезпечується або апаратурою процесора, або компілятором, на вхід якого надходять програми на традиційній послідовній мові програмування, або спільно апаратурою і компілятором.

У компіляторах використовується витончена техніка виягування паралелізму з послідовних програм. Апаратура мікропроцесорів орієнтована на виділення більш простих форм паралелізму, у тому числі природного. Прагнення використовувати властивий більшості програм природний паралелізм обчислення цілочисельних адресних виразів і власне обробки даних у форматі плаваючою комою, привело до появи рознесених архітектур (decoupled architecture). У першому наближенні мікропроцесор з рознесеною архітектурою складається з двох зв'язаних підпроцесорів, кожний з яких керується власним потоком команд.

Важливим системним аспектом рознесеної архітектури є інтерфейс між процесором і пам'яттю за допомогою транзакцій читання і запису. Це дозволяє розташувати між процесором і пам'яттю довільне комутаційне середовище, що підводить логічний базис під концепцію побудови багатопроцесорних систем.

Розвиток мікропроцесорів відбувається при постійному прагненні збереження послідовності програмного забезпечення (ПЗ) і підвищення продуктивності за рахунок удосконалювання архітектури і збільшення тактової частоти. Збереження послідовності ПЗ і підвищення продуктивності суперечать один одному. Процесори з системою команд x86, що відносяться до класу CISC-процесорів, мають більш низькі тактові частоти в порівнянні з мікропроцесорами провідних компаній-виготовлювачів RISC-процесорів. Існують задачі, на яких продуктивність x86 мікропроцесорів значно нижча, ніж у RISC-процесорів, реалізованих

на тій же елементній базі. Однак можливість використання сумісного ПЗ для різних поколінь x86 процесорів, випущених протягом останнього десятиліття, забезпечує їм стійке домінуюче положення на ринку.

В наш час на основі піонерських розробок компаній NexGen і AMD, підхоплених компанією Intel, почата спроба вирішити проблему підвищення продуктивності в рамках архітектури x86. Ці компанії в останніх розробках, зберігаючи послідовність за системою команд із CISC-мікропроцесорами сімейства x86, створюють нові пристрої з використанням елементів RISC-архітектури. Прикладом такого підходу можуть служити мікропроцесори Nx586 (NexGen), K5, K6 (AMD), Pentium Pro, Pentium II (Intel), що використовують концепцію розділеної (decoupled) архітектури і RISC-ядра. У мікропроцесор вбудовується апаратний транслятор, що перетворює команди x86, у команди RISC-процесора. При цьому одна команда x86 може породжувати до чотирьох команд RISC-процесора. Виконання команд відбувається так, як у розвинутому суперскалярному процесорі. Компанія Intel використовувала цей підхід у своєму мікропроцесорі Pentium Pro, що дуже зміцнило її позиції на фоні досягнень RISC-архітектур.

Архітектура суперскалярних і мультискалярних процесорів

Є два крайніх підходи до відображення властивого мікропроцесору внутрішнього паралелізму обробки даних на архітектурному рівні в системі команд. Перший підхід більш консервативний і полягає в тому, що ніякої вказівки на паралельну обробку всередині процесора система команд не містить. Такі процесори відносяться до класу суперскалярних. Другий підхід цілком відкриває всі можливості паралельної обробки. У спеціально відведених полях команди кожному з АЛП, що працюють паралельно, пропонується дія, яку пристрій повинен виконати. Такі процесори називаються процесорами з довгим командним словом (VLIW). Передбачається, що існують компілятори з мов високого рівня, що готують програми для завантаження їх у мікропроцесори.

Основна ідея, що визначає розвиток суперскалярних мікропроцесорів, полягає в побудові можливо більшої кількості паралельних структур при збереженні традиційних послідовних програм. Це означає, що компілятори й апаратура мікропроцесора самі, без втручання програміста, забезпечують завантаження функціональних пристроїв, які працюють паралельно. Відповідно до моделі послідовного програмування програми пишуться в припущенні, що команди будуть виконані в тому ж порядку, в якому вони представлені в програмі. Однак з метою досягнення більшої ефективності сучасні процесори намагаються виконувати кілька команд одночасно і в деяких випадках у порядку, відмінному від їхньої послідовності в програмі. Це перевпорядкування може бути виконане в трансляторі і (або) в апаратних засобах під час виконання. Суперскалярні і VLIW-процесори належать до класу архітектур, що використовують паралельність рівня команди.

Підвищити ступінь паралелізму програми можна шляхом зміни відповідним чином її статичної чи динамічної структури, тобто множиною послідовностей інструкцій у порядку їхнього виконання. Припустимі межі перетворення динамічної структури програми задають відносини, які існують на множині інструкцій: залежність по керуванню і залежність за даними. При опису архітектур суперскалярних процесорів часто використовується модель вікна виконання. При виконанні програми мікропроцесор як би просуває за статичною структурою програми вікно виконання. Команди у вікні можуть виконуватись паралельно, якщо між ними немає залежності.

Для усунення залежностей, викликаних командами переходів, використовується метод передбачення, що дозволяє вибирати й умовно виконувати команди передбаченого переходу. Якщо пізніше виявляється, що передбачення було зроблено вірно, то результати умовно виконаних команд приймаються. Якщо передбачення було помилковим, стан процесора відновлюється на момент ухвалення рішення про виконання переходу.

Команди, розміщені у вікні виконання, можуть бути залежні за даними. Ці залежності обумовлені використанням тих самих ресурсів пам'яті (регістрів, комірок пам'яті) у різних командах. Тому для правильного виконання програми необхідне використання цих ресурсів у порядку, що визначається програмою.

В сучасних мікропроцесорах широко використовується принцип конвеєрного виконання окремих елементарних операцій. Конвеєризація внутрішніх процесів дозволяє виконувати команду за кожен процесорний цикл. Подальші впровадження принципів конвеєризації привели до появи класу суперскалярних мікропроцесорів. Їхньою відмінною рисою є можливість виконання декількох команд за один процесорний цикл. Такий режим виконання програми став можливим завдяки наявності в процесорах декількох виконавчих пристроїв.

В число основних блоків суперскалярного мікропроцесора входять: блок вибірки команд і передбачення переходів, блок декодування команд, аналізу залежностей між командами, перейменування і диспетчеризації, блоки регістрів і обробних пристроїв з фіксованою і плаваючою комою, блок керування пам'яттю, а також блок впорядкування виконаних команд.

Принаймні дві обставини обмежують ефективність використання суперскалярних архітектур. По-перше, є обмеження на ступінь паралелізму на рівні команд, навіть якщо застосовується найдосконаліша техніка суперскалярних обчислень. Перше обмеження виникає з умовних переходів. Інше впливає з того, що розмір вікна виконання (число активних команд, що можуть виконуватись паралельно) обмежує можливий властивий програмі паралелізм, тому що не розглядається паралельне виконання команд, які знаходяться на відстані, що перевищує розмір вікна. По-друге, складність суперскалярного процесора зростає з кількістю команд, що виконуються паралельно, і навіть швидше. Ймовірніше всього, що межею розпаралелювання при суперскалярній обробці є запуск одночасно на виконання в кожному такті 7-8 команд.

Альтернатива суперскалярній обробці - довге командне слово (VLIW). Використання цього методу припускає завдання в командному слові сукупності команд, що виконуються паралельно. Підготовкою таких програм займається компілятор. Достоїнства VLIW полягають в тому, що, по-перше, компілятор може більш ефективно досліджувати залежності між командами і вибирати команди, що паралельно виконуються, ніж це робить апаратура суперскалярного процесора, обмежена розміром вікна виконання. По-друге, VLIW-процесор має більш простий пристрій керування і потенційно може мати більш високу тактову частоту. Однак у VLIW процесорів є серйозний фактор, що знижує їхню продуктивність. Це команди розгалуження, що залежать від даних, значення яких стають відомі тільки в динаміці обчислень. Вікно виконання VLIW-процесора не може бути дуже великим через відсутність в компілятора інформації про залежності, що формуються динамічно в процесі виконання.

Подальше підвищення продуктивності мікропроцесорів пов'язується в наш час зі статичним і динамічним аналізом кода з метою виявлення резервів паралелізму рівня окремих команд і програмних сегментів з використанням інформації, наданої компілятором мови високого рівня. Дослідження в даному напрямку привели до розробки мультискалярної архітектури процесорів, яка є подальшим розвитком суперскалярної архітектури.

В наш час роботи в цьому напрямку перебувають у стадії теоретичного дослідження й імітаційного моделювання, однак, мабуть, вже незабаром варто очікувати появи перших мікропроцесорів, що повною мірою реалізують всі переваги мультискалярної архітектури.

В мультискалярних процесорах програма розбивається на сукупність задач за допомогою програмних і апаратних засобів. Задача - частина програми, виконання якої відповідає безперервній області динамічної послідовності команд (наприклад, частина базисного блока, базисний блок, множина базисних блоків, одинична ітерація циклу, повний цикл, звертання до функції, і т.д.). Задачі програми статично розмежовуються анотаціями. Залежності між операторами програми з керування представляються як граф керуючих залежностей (ГУЗ), у якому вершинами є задачі, а дугами задається порядок їх виконання. Динаміка виконання програми може розглядатися як обхід ГУЗ програми. На кожному кроці обходу мультискалярний процесор призначає одну задачу на один із процесорних елементів для виконання, без обліку фактичного змісту задачі, і продовжує обхід ГУЗ від розглянутої вершини до наступної. Задача призначається для виконання деякому процесорному елементу, передачею йому початкового значення програмного лічильника. Множина ініційованих у такий спосіб задач виконується паралельно на процесорних елементах, результатом чого є виконання множини команд за один процесорний такт.

Універсальні мікропроцесори

З універсальних найбільш поширеними в наш час є такі високопродуктивні мікропроцесори: архітектура x86, компанія Intel - лінія

Pentium (P5, P6), компанія AMD (NexGen) - K5, K6, компанія Cyrix - M1, M2; архітектура Power PC, компанія Motorola - Power PC 603, 604, 620; архітектура PA, компанія HP - PA-8000; архітектура Alpha, компанія DEC - лінія Alpha (21064, 21164, 21164A); архітектура SPARC, компанія SUN - лінія SPARC; архітектура MIPS, компанія Silicon Graphics - лінія MIPS R-x (R 10000).

Між виробниками йде гостра конкурентна боротьба, що особливо загострилася в зв'язку з тим, що велика кількість транзисторів у цих мікропроцесорах дозволяє вичерпно реалізувати в них структурно-функціональні рішення, характерні для векторно-конвексних супер-EOM з одним потоком команд, для потужних робочих станцій і великих EOM із багатокристальними процесорами.

Історично мікропроцесори з архітектурою x86 домінували в персональних EOM, а RISC-процесори в сукупності з графічними процесорами й потужними системами введення-виведення утворювали процесори робочих станцій. В наш час процесори з архітектурою x86 просунулися на ринок робочих станцій, і деякі виробники робочих станцій, такі як SUN та DEC, намагаються вийти зі своїми процесорами на ринок персональних EOM. В рамках програми "Прискореної стратегічної комп'ютерної ініціативи" (ASCI) у США реалізуються проекти побудови суперкомп'ютерів із продуктивністю на рівні 10 трильйонів оп/с на базі мікропроцесорів Pentium Pro, Power PC, Alpha, R-10000.

Архітектура мікропроцесорів розвивається в двох магістральних напрямках: перший характеризується прагненням до досягнення високої продуктивності, головним чином за рахунок високої тактової частоти, а другий пов'язаний з досягненням високої продуктивності за рахунок ускладнення логіки планування обчислень і внутрішньої структури процесора.

Загальною особливістю більшості мікропроцесорів є високошвидкісна обробка 64-розрядних операндів з фіксованою і плаваючою комою. Побудова функціональних вузлів таких мікропроцесорів вимагає складної схемотехніки, що обумовлює використання великої кількості транзисторів (близько 2 млн. для логіки процесора) і великої кількості шарів металізації (чотири). Відповідно використовується складний технологічний цикл виробництва, що обумовлює високу ціну мікропроцесорів. Продуктивність універсальних мікропроцесорів сучасного третього покоління наближається до мільярда операцій у секунду. В той же час при створенні нових RISC-мікропроцесорів розроблювачі дозволяють собі відходити від принципів, що лежать в основі канонічної RISC-архітектури. Робиться це в першу чергу для подолання основного недоліку, властивого RISC-архітектурі - великого обсягу кода. Прикладом у даному випадку може служити процесор Power C/AS, в систему команд якого включені складені команди.

Мікропроцесори з архітектурою x86

На відміну від попередніх мікропроцесорів з системою команд x86, в процесорах сімейства Pentium є цілий ряд технічних нововведень, до числа яких відносяться: близька до суперскалярної архітектура; окремі кеш-пам'яті для команд і даних; передбачення переходів; високопродуктивні операції з плаваючою комою; удосконалена 64-розрядна шина даних; засоби забезпечення цілісності даних; SL-технологія з засобами керування енергоспоживанням; підтримка багатопроцесорності; моніторинг продуктивності різних розмірів сторінки пам'яті.

Два конвеєри процесора Pentium можуть виконувати дві команди одночасно. При цьому кілька команд можуть знаходитись на різних етапах виконання. Однак два конвеєри не є незалежними. При зупинці одного зупиняється й інший. Блок арифметики з плаваючою комою використовує блок арифметики з фіксованою комою. Отже, ці операції не можуть виконуватись паралельно. Це обмежує суперскалярність процесора.

У процесорі Pentium багато команд мікрокода, які застосовувались в попередніх поколіннях мікропроцесорів, замінені внутрішніми командами. Що стосується більш складних команд, то удосконалений мікрокод процесора Pentium збільшує продуктивність, застосовуючи для виконання цих команд два конвеєри для обробки цілих чисел.

Кожна кеш-пам'ять процесора Pentium має розмір 8 Кбайт. Кеш-пам'яті є частково-асоціативними. Пошук необхідної інформації виконується в стандартних 32-байтових рядках. Буфер трансляції адрес (TLB) перетворює адреси комірок зовнішньої пам'яті у відповідні адреси даних у кеш-пам'яті.

Кеш даних процесора Pentium використовує метод зворотного запису. Метод зворотного запису дозволяє модифікувати дані в кеші без звертання до ОП (дані записуються в ОП тільки при видаленні з кеша). У попередніх поколіннях використовувався кеш з наскрізним записом. При кожній модифікації даних у кеші дані передавалися у внутрішню пам'ять. Метод зворотного запису підвищує продуктивність, зменшуючи навантаження на шину інтерфейса з пам'яттю. В процесорі Pentium забезпечується когерентність (погодженість) даних у кешах процесорів і в основній пам'яті при роботі в мультипроцесорній системі.

Процесор Pentium виконує прогнозування переходів, використовуючи буфер ВТВ (Branch Target Buffer) і два буфери попередньої вибірки. Один буфер застосовується для попередньої вибірки команди, у припущенні, що переходу нема, інший виконує попередній вибір інструкцій у буфер, використовуючи вміст ВТВ. Алгоритм попередження переходів процесора Pentium не тільки прогнозує вибір простих гілок, а й підтримує більш складні прогнозування (наприклад, у вкладених циклах). Це робиться за допомогою збереження в буфері ВТВ декількох адрес переходів. ВТВ зберігає до 256 результатів переходів, що

дозволяє виконувати правильне прогнозування, ймовірність якого не менша 0,8.

У випущеному в січні 1997 р. мікропроцесорі Pentium MMX система команд розширена 57 додатковими командами, орієнтованими на ефективне виконання типових мультимедійних алгоритмів (до числа яких відносяться і багато алгоритмів, характерних для цифрової обробки сигналів - операції над векторами, згортка, перетворення Фур'є і т.п.). Це перша істотна зміна в системі команд мікропроцесорів сімейства x86, починаючи з виходу в світ мікропроцесора Intel80386 у 1985 році, що мав 220 команд. До числа задач, що ефективно вирішуються цим мікропроцесором, відносяться: синтез звуку і музики, розпізнавання мови, обробка відео- і графічної інформації, виконання комунікаційних функцій та ін.

Застосування MMX-команд дозволяє збільшити швидкість виконання мультимедійних задач на 60% в порівнянні зі звичайним процесором Pentium при однакових тактових частотах.

Мікропроцесори компанії AMD

Розроблені компанією AMD мікропроцесори із системою команд x86, що одержали назву "Криптон-5" (скорочено K5 і K6), успішно конкурують з процесорами сімейства Pentium.

K5 - це перший мікропроцесор AMD, при створенні якого не використовувалася ніяка інтелектуальна власність Intel (за винятком мікрокода), в той же час він має кращу в порівнянні з процесорами Intel продуктивність. За даними AMD реальні задачі (такі як Microsoft Excel, Word чи CorelDraw) працюють на процесорах серії K5 на 30% швидше, ніж на Pentium з тією ж тактовою частотою. Така продуктивність досягається в основному за рахунок збільшеного обсягу кеш-пам'яті і більш прогресивної суперскалярної архітектури. Замість парних конвейерів цілочисельних обчислень Pentium у K5 використовується суперскалярна архітектура зі зчетвереною подачею команд п'яти обробним блокам і RISC-ядром з унікальним декодером інструкцій x86.

Використовується динамічне перейменування регістрів, прогнозування переходів, зміна послідовності виконання й інших методів динамічного виконання. K5 реалізує гібридну CISC/RISC-технологію.

Новий мікропроцесор компанії AMD K6 є результатом спільних зусиль компанії AMD і фахівців з колишньої NexGen. Мікропроцесор K6 виготовляється за технологією КМОН/0,35 мкм із п'ятишаровою металізацією, містить 8,8 мільйонів транзисторів на кристалі площею 162 мм² і працює з тактовими частотами 166,200 і 233 Мгц. Процесор цілком сумісний по системі команд з сімейством x86. Крім того, K6 оснащений мультимедійним розширенням системи команд - MMX. За продуктивністю K6 порівнянний з Pentium Pro і істотно перевершує Pentium MMX. На відміну від Pentium Pro K6 однаково успішно працює як з 32-розрядними,

так і з 16-розрядними задачами.

Як і в K5, у K6 застосована суперскалярна архітектура RISC86 з розділним декодуванням/виконанням команд, що забезпечує узгодженість із системою команд x86 і досягнення високої продуктивності, властивій мікропроцесорам 6-го покоління.

Висока продуктивність процесора досягається завдяки ряду нових архітектурних і технологічних рішень: перекодування команд при їхній вибірці в кеш; наявність кеша першого рівня по 32 Кбайт для даних і команд; наявність високопродуктивного блока мультимедійних операцій стандарту MMX; множинне декодування x86 інструкцій в одноктактові RISC-операції; підтримка логіки передбачення розгалужень; наявність паралельних дешифраторів, централізованого планувальника операцій і семи виконавчих блоків, що забезпечують суперскалярне виконання x86 інструкцій у шестиступеневому конвеєрі.

Планувальник K6 містить у буфері 24 команди RISC86 і призначає інструкції на виконання семи виконавчим модулям: збереження, завантаження, переходу, цілочисельному АЛП, цілочисельному блокові мультимедійних команд, операцій з плаваючою комою. Наявність буфера дозволяє планувальнику в динаміці аналізувати й змінювати обрану для виконання послідовність команд, забезпечуючи максимальне завантаження виконавчих пристроїв.

Мікропроцесори компанії Cyrix

Іншим конкурентом компанії Intel, після AMD, за обсягом виробництва процесорів з архітектурою x86 є Cyrix.

Мікропроцесор Cyrix 5x86 - це перший представник серед процесорів компанії Cyrix з архітектурою x86 5-го покоління, запропонований як альтернатива Pentium. Для спрощення етапу переходу користувачів на новий процесор компанія забезпечила його сумісність по виводах з 486-м процесором. У 5x86 втілені архітектурні і структурні рішення, характерні для процесорів п'ятого покоління, такі як 64-розрядна внутрішня архітектура, передбачення переходів, попереднє посилання даних, виконання декількох операцій за один такт, 80-розрядний пристрій обчислень із плаваючою комою, 16-Кбайтовий загальний кеш даних і команд.

Особливістю мікропроцесора є високоефективна система керування енергоспоживанням процесора і зовнішніх пристроїв, що робить його кращим для використання в мобільних комп'ютерах, для яких критичне енергоспоживання і відведення тепла.

Архітектура 5x86 - результат компромісу між продуктивністю і простотою реалізації, завдяки якій удалося зменшити число транзисторів на кристалі і знизити енергоспоживання.

Мікропроцесор Cyrix 6x86 містить два незалежних семиетапних конвеєри, що дозволяють виконувати кілька команд за один такт. Процесор має два кеша: загальний кеш даних і команд розміром 16 Кбайт і

256-байтову кеш-пам'ять команд із прямим відображенням. Виділений кеш команд дозволяє уникнути частих конфліктів при звертанні до даних і команд у загальному кеші. В 6x86 реалізовані такі прогресивні архітектурні методи підвищення продуктивності, як перевпорядкування інструкцій, динамічне усунення залежності між командами, перейменування регістрів, прогнозування переходів.

Наступною розробкою компанії Сугіх є процесор з мультимедійними розширеннями 6x86MX, відомий також як M2. Сугіх представляє цей процесор як більш дешевий конкурент Pentium II. Подібно Pentium PRO, він оптимізований для роботи з 32-розрядними програмами, але, на відміну від останнього, він успішно працює з 16-розрядними програмами. Процесор виготовляється за технологією КМОН 0,35 мкм із 5-шаровою металізацією. Йому потрібно дві напруги живлення: 2,9 Вт для процесорного ядра і 3,3 Вт для підсистеми введення-виведення. Процесор може працювати з частотою шини до 75 МГц.

6x86MX - це поліпшений процесор 6x86, у який додана можливість виконання набору з 57 мультимедійних команд, сумісних із MMX розширеннями Intel, підвищена тактова частота, збільшений обсяг кеш-пам'яті, застосований дворівневий буфер трансляції адреси (TLB) і поліпшений кеш адреси переходів. Процесор базується на суперскалярному ядрі 6x86, що, завдяки наявності двох незалежних конвеєрів, здатне виконувати кілька команд за такт. У 6x86MX збільшена довжина конвеєрів, що дозволило підвищити тактову частоту процесора.

Мікропроцесори з архітектурою Alpha

На сьогоднішній день мікропроцесори Alpha є найбільш продуктивними однокристальними пристроями, що випускаються серійно. Мікропроцесори Alpha є гарною ілюстрацією концепції Speed Daemon - досягнення високої продуктивності за рахунок збільшення тактової частоти при відносно простій логіці функціонування.

Мікропроцесор Alpha 21064 має 64-розрядну суперскалярну RISC-архітектуру з двома виконавчими конвеєрами. Мікропроцесор може виконувати за один такт до двох команд з фіксованою і плаваючою комою, чи команду переходу. Блок прогнозування переходу забезпечує ймовірність правильного прогнозу 0,8.

Мікропроцесор містить на кристалі окремі кеш-пам'яті команд і даних, кожен ємністю 16 Кбайт, 32 регістри з плаваючою комою і 32 регістри з фіксованою комою. Кеш другого рівня - зовнішній, може складати від 128 Кбайт до 16 Мбайт. Тактова частота шини в системі може бути в 2,5 - 10 разів меншою, ніж частота роботи мікропроцесора. Ширина зовнішніх шин адреси і даних складає 43 і 128 біт відповідно.

У мікропроцесорах Alpha 21064 не використовується складна логіка перевпорядкування команд і перейменування регістрів. Забезпечення ефективного завантаження виконавчих пристроїв покладається на компілятор.

Наступним рекордним досягненням DEC став випуск у вересні 1994 року мікропроцесора нового покоління Alpha 21164 з тактовими частотами 266 і 300 МГц. Це був перший мікропроцесор, що мав продуктивність до 1,2 мільярди операцій в секунду. Застосування нової технології 0,5 мкм дозволило розмістити на кристалі площі 298 кв.мм 9,3 млн. транзисторів. Архітектура цього процесора максимально орієнтована на досягнення високої тактової частоти. У порівнянні з попереднім поколінням, у Alpha 21164 вдвічі збільшилося число виконавчих пристроїв, а число тактів, необхідних для виконання команд, зменшилося.

Чотири команди читаються одночасно з кеш-пам'яті команд обсягом 8 Кбайт і розміщуються в один із двох буферів команд, кожний з яких може зберігати чотири команди. Команди читаються з буфера в порядку, обумовленому програмою. Коли для команди готові операнди, команда передається на виконання у відповідний їй функціональний блок. Для забезпечення правильного виконання переривань процесор не змінює порядку виконання команд, а зміна значень регістрів відбувається в порядку, запропонованому програмою.

За один такт процесор може виконувати до 4-х команд: дві з фіксованою комою, дві – з плаваючою. Як і в більш ранніх мікропроцесорах Alpha, у Alpha 21164 інтенсивно використовується конвеєризація. Перші 4 етапи виконання команди є загальними для всіх операцій і виконуються в модулі інструкцій. Це попередня вибірка команди, перехід, визначення, перевірка/подача команди. Блок цілочисельних операцій додає до обробки команд ще 3 етапи, а блок операцій із плаваючою комою - ще 5. В мікропроцесорі Alpha 21164, на відміну від попередніх моделей, блок виконання обновляє регістри безпосередньо. Йому не потрібний складний механізм відстеження команд, що забезпечує відновлення регістрів у потрібному порядку. Щоб уникнути втрат продуктивності через чекання завершення команд, Alpha 21164 передбачає обхідні шляхи, завдяки яким операнди стають доступними ще до завершення команди. Це аналогічно методам посилення даних з попередженням, які застосовуються в інших мікропроцесорах.

Подальший розвиток мікропроцесорів сімейства характеризується збільшенням тактової частоти (у липні 1996 випущений Alpha 21164 з 500 МГц, у березні 1997 - 600 МГц), різким зниженням (майже в 2 рази) ціни на процесори, а також пошуком шляхів адаптації архітектури Alpha для використання в Windows NT системах.

Одна з останніх розробок - мікропроцесор Alpha 21164PC (березень 1997р.), покликана забезпечити проникнення мікропроцесорів DEC у настільні системи, створюючи тим самим конкуренцію процесорам компанії Intel. Сумісність із системою команд x86 DEC забезпечує за допомогою програмного транслятора двійкового кода DIGITAL FX!32. Відмінностями Alpha 21164PC є зовнішній кеш 2-го рівня, збільшений до 16 Кбайт кеш команд, наявність блока, що забезпечує кодування/декодування в реальному часі даних у відповідності зі

стандартом MPEG-2.

У травні 1997 р. компанією DEC був представлений мікропроцесор нового покоління - Alpha 21264, що значно перевершує за продуктивністю попередніх представників сімейства. Мікропроцесор містить 15,2 млн. транзисторів на кристалі площею 310 кв. мм і виготовляється за КМОН технологією з 6 шарами металізації.

На відміну від попередніх мікропроцесорів сімейства, поряд з високою тактовою частотою, у Alpha 21264 використовується складний механізм динамічного виконання команд: динамічне планування зі зміною послідовності команд, перейменування регістрів, спекулятивне виконання команд.

Для динамічного виконання розглядаються відразу 80 команд - більше, ніж у будь-якого іншого процесора. Після декодування команда розміщується в одну з черг до відповідних пристроїв. Команди, що одержали всі операнди, конкурують за доступ до виконавчих пристроїв. Більший пріоритет мають команди, що довше знаходяться в черзі. Одночасно може виконуватись до 6 команд.

Мікропроцесори з архітектурою SPARC

Лінія процесорів з архітектурою SPARC включає 32-розрядні мікропроцесори MicroSPARC, SuperSPARC, HiperSPARC і 64-розрядний мікропроцесор UltraSPARC. Традиційно системи на основі SPARC мікропроцесорів - це високопродуктивні робочі станції для наукових розрахунків і Unix-сервери. З випуском нового покоління мікропроцесорів значно покращилися можливості SPARC-систем з обробки графіки і відеозображень. UltraSPARC являє собою один з перших процесорів загального призначення, в якому ці функції реалізовані апаратно. Він має спеціальний модуль для обробки відеоданих і графіки, представленої в RGB і альфа-форматі. Функції обробки відеозображень можуть працювати відразу з 8 елементами зображення. Обробка зображень виконується в блоці FPU, що звичайно виконує операції над 64-бітовими даними з плаваючою комою. У цьому блоці виконується логічне додавання і множення елементів зображення. В систему команд мікропроцесора включений набір команд, що дозволяє завантажувати й обробляти дані у вигляді 64-бітових блоків. Він нараховує 30 команд для ефективною обробки мультимедіа, графіки, зображень і інших алгоритмів, орієнтованих на цілочисельну обробку. За один такт може виконуватись до 8 цілочисельних операцій над байтами чи півсловами.

При реалізації алгоритму компресії-декомпресії MPEG більше всього часу вимагає та частина алгоритму, що виконує аналіз руху і порівняння кожної частини потокового кадру з попереднім. Завдяки наявності спеціальних команд у UltraSPARC ці дії виконуються як одна графічна операція. Спеціальна підсистема пам'яті автоматично завантажує елемент зображення в 8-байтові блоки. Для цього не потрібно виконання окремої команди. При конвеєризації цих спеціальних команд мікропроцесор буде виконувати одну операцію за такт. Використання таких спеціалізованих команд забезпечує 80-кратне підвищення швидкості

виконання обробки зображень у порівнянні з іншими SPARC-процесорами.

UltraSPARC має також кілька спеціальних засобів, що поліпшують його роботу з пам'яттю. Це підвищує продуктивність програм мультимедіа й інших системних задач. Найбільшою зміною є нова команда переміщення блока даних по шині "процесор-пам'ять" зі швидкістю 600 Мбайт/с. Це дозволяє основному процесору працювати як відеопроцесор, передаючи дані з екрана на екран. Переміщення блока можна з вигодою використовувати й в інших задачах, наприклад у мережному програмному забезпеченні для переміщення пакетів даних.

Процесори з архітектурою SPARC - це на сьогоднішній день єдині процесори, що використовують концепцію регістрових вікон. Замість 32 базових регістрів ці мікропроцесори пропонують 8 вікон регістрів, що перекриваються, по 24 регістра кожне. З початком нової процедури або гілки обробки вікно не потрібно записувати в пам'ять - новий контекст просто буде використовувати нове вікно регістрів. Однак, на практиці, число доступних вікон швидко вичерпується, і в цьому випадку приходиться записувати інформацію в пам'ять. Однозначної думки про достоїнства і недоліки подібної моделі поки не існує. В UltraSPARC-2 доданий додатковий механізм, завдяки якому при кожному перериванні стає доступним нове вікно з 8 регістрів. Таке рішення повинне значно збільшити швидкість виконання мультипроцесорного програмного кода.

Мікропроцесор PA-8000

Hewlett-Packard була однією з перших компаній, що вийшли на ринок RISC-процесорів. Її 32-розрядний мікропроцесор PA-RISC з архітектурою Precision Architecture був випущений у 1986 р. Послідовно розвиваючи принципи RISC-архітектури, HP випустила один із найпродуктивніших на сьогоднішній день мікропроцесорів PA-8000, у якому повною мірою втілені основні принципи динамічного виконання команд.

PA-8000 має 64-розрядну чотириконтверсну архітектуру з оригінальною схемою зміни послідовності виконуваних команд. Процесор містить 10 функціональних пристроїв: два цілочисельних АЛП, два цілочисельних пристрої зсуву/об'єднання, два пристрої множення/нагромадження з плаваючою крапкою, два пристрої ділення/добування кореня і два пристрої завантаження/збереження. Пристрої множення/нагромадження вимагають трьох тактів і цілком конвесризуються для обробки даних з одинарною точністю. Їхня продуктивність складає до 4 FLOPS за такт. Блоки ділення вимагають 17 тактів і не конвесризуються.

У PA-8000 використовується буфер перевпорядкування команд, що

переглядає наступні 56 команд у потоковій інструкції і визначає, які з них можна виконувати паралельно.

Мікропроцесор Power PC 620

Процесор PowerPC, створений зусиллями альянсу IBM, Apple і Motorola, має вже солідну історію. З 1991 р. регулярно з'являються нові версії цього процесора. У 1993 р. з'явилася PowerPC 601 - недорога 32-розрядна реалізація архітектури PowerPC. Слідом за ним у тому ж році був випущений мікропроцесор PowerPC 603 - менш могутня версія 32-розрядного процесора, призначена для портативних комп'ютерів. У 1995 р. вийшла перша 64-розрядна реалізація архітектури PowerPC - мікропроцесор PowerPC 620. Якщо попередні версії мікропроцесорів PowerPC були орієнтовані на персональні комп'ютери, то PowerPC 620 призначений для робочих станцій і високопродуктивних серверів.

Цей мікропроцесор має 6 незалежних виконавчих блоків, високошвидкісний шинний інтерфейс, внутрішній кеш 64Кбайт. Він забезпечує високу продуктивність, необхідну в задачах моделювання й обробки транзакцій. При виробництві процесора використовується 0,5-мікронна технологія КМОН з 4 шарами металізації. На кристалі площею 311 мм² розміщено 7 млн. транзисторів, що майже вдвічі більше, ніж у PowerPC 604. Як і PowerPC 603 і 604, PowerPC 620 має робочу напругу 3,3В. На частоті 133 МГц він споживає не більше 30Вт.

Як і в попередніх мікропроцесорах, у PowerPC 620 використовується гарвардська архітектура з окремими маршрутами кода і даних. Процесор має 2 вбудованих кеша першого рівня по 32Кбайт кожний. Кожен кеш має власний блок керування пам'яттю і функціонує незалежно від іншого.

PowerPC 620 має суперскалярне RISC-ядро. У процесорі застосовується чотириконвейерна суперскалярна архітектура із шістьма виконавчими модулями: трьома цілочисельними АЛП, блоком операцій із плаваючою комою, блоком завантаження/збереження і блоком переходів. За один такт процесор може виконувати до 4-х команд. Для підтримки ефективного завантаження виконавчих блоків у процесорі застосовується динамічне прогнозування переходів.

Мікропроцесор R10000

На сьогоднішній день сімейство мікропроцесорів Rxxxx є абсолютним лідером за обсягом продаж серед RISC-процесорів. Область застосування цих процесорів велика: від ігрових приставок компаній Sony і Nintendo, до високопродуктивних серверів і робочих станцій компаній Pyramid Technology, MIPS, Silicon Graphics Inc., Siemens Nixdorf Inc. Розроблений компанією Mips 64-розрядний мікропроцесор R10000 побудований на базі попередніх поколінь RISC-процесорів (R2000, R3000,

R4000 і R5000).

В основу цього мікропроцесора покладена суперскалярна RISC-технологія 5-го покоління, реалізована раніше в орієнтованому на супер-ЕОМ процесорі R8000. Однак на відміну від багатокристалного R8000, оптимізованого для високопродуктивних наукових розрахунків, R10000 являє собою однокристалний процесор загального призначення для настільних ПК, робочих станцій і серверів. Основними особливостями архітектури процесора R10000 є високий ступінь паралельності виконання мікрооперацій, ефективне прогнозування переходів і планування завантаження функціональних модулів. R10000 має 5 функціональних блоків: два блоки операцій із плаваючою комою, два блоки цілочисельних операцій і один блок завантаження/збереження.

R10000 містить високошвидкісний (до 1,6 Гбайт/с) внутрішньокристалний інтерфейс "процесор-шина", що дозволяє поєднувати в багатопроцесорній конфігурації до чотирьох процесорів без використання додаткових інтерфейсних схем.

Контрольні запитання

1. Які мікропроцесори називають універсальними?
2. Назвіть основні області застосування цифрових сигнальних процесорів та мікроконтролерів.
3. Що таке ЦЛІС?
4. Які команди називають скалярними, а які векторними?
5. Які шляхи підвищення продуктивності мікропроцесорів Ви знаєте?
6. Для чого потрібно зберігати послідовність програмного забезпечення?
7. Що таке суперскалярні мікропроцесори? Які властивості вони мають?
8. Які процесори називають процесорами з довгим командним словом?
9. Назвіть основні блоки, що входять до структури суперскалярного процесора.
10. Які обставини обмежують ефективне використання суперскалярних архітектур?
11. Що таке мультискалярна архітектура мікропроцесорів? Які властивості вона має?
12. Охарактеризуйте мікропроцесори з архітектурою x86.
13. В чому полягають основні особливості мікропроцесорів K5, K6?
14. Назвіть основні архітектурні особливості процесора Cugix 6x86MX.
15. Охарактеризуйте процесори Alpha 21064, 21164, 21264.
16. Назвіть основні архітектурні особливості процесора PA-8000.
17. Охарактеризуйте процесори з архітектурою SPARC.
18. Які властивості має мікропроцесор PowerPC 620?
19. Назвіть основні особливості архітектури мікропроцесора R10000.

ЛАБОРАТОРНИЙ ПРАКТИКУМ

В даному розділі розглядається тематика лабораторних робіт, відповідно до якої студенти попередньо готуються до виконання робіт, користуючись при цьому лекційним матеріалом, матеріалом практичних занять, літературою та відповідними вказівками, викладеними в комп'ютерних файлах.

Лабораторна робота №1

Знайомство з функціональними можливостями програми OrCAD for Windows та експериментальне їх дослідження

Мета роботи:

ознайомитися з проектуванням і моделюванням схем цифрових пристроїв з використанням бібліотеки символів в програмі OrCAD Express for Windows.

Порядок виконання роботи

1. Ознайомитись зі вказівками по роботі в OrCAD Express for Windows.
2. Побудувати схему, використавши елемент у відповідності із варіантом завдання (задається викладачем).
3. Розширити розрядність, скористатись для з'єднання елементів шиною.
4. Додати бібліотеку символів компонентів.
5. Здійснити пошук елемента по всіх існуючих бібліотеках символів компонентів, що знаходяться в каталозі Library.
6. Виконати моделювання схеми, побудованої в п.2.
7. При відсутності математичної моделі елемента, додати до проекту відповідний *.vhd файл.
8. Виконати редагування стимулів за допомогою команди Edit Interactive.
9. Скориставшись другим маркером визначити різницю моментів часу між їх положеннями.
10. Виконати перегляд значень сигналів у вікні схеми.
11. Роздрукувати схему в файл.
12. Зробити висновки по роботі. Що нового ви дізнались про роботу в OrCAD Express for Windows?

Методичні вказівки

Для виконання цієї роботи слід пригадати матеріал по роботі з програмою OrCAD Express for Windows, що вивчався в минулому році з дисципліни “Схемотехніка ЕОМ”, та скористатись методичними вказівками по роботі з програмою OrCAD, які викладені в комп'ютерному файлі.

Лабораторна робота №2

Дослідження послідовного підсумовувального пристрою

Мета роботи:

ознайомитися з принципом побудови послідовних пристроїв на прикладі операції арифметичного додавання та отримати навички з побудови схеми, її налагодження і моделювання за допомогою сучасних програмних засобів.

Порядок виконання роботи

1. Побудова і аналіз структурної схеми пристрою.
2. Побудова функціональної схеми пристрою в заданому базисі.
3. Визначення вхідних сигналів для перевірки схеми на працездатність при заданих умовах.
4. Машинне моделювання схеми послідовного підсумовувального пристрою.
5. Аналіз результатів і висновки.

Методичні вказівки

Для підсумовування чисел, представлених у послідовному коді, використовують одну однорозрядну підсумовувальну схему. Числа, які додаються, подають на вхід цієї схеми, починаючи з молодших розрядів. Сигнал переносу з виходу підсумовувальної схеми з затримкою на один такт додавання надходить на її вхід одночасно із сигналами наступних розрядів чисел, які підсумовуються. При моделюванні схеми необхідно підключити бібліотеку `ls.vhd`, яка містить математичну модель суматора (74ls183).

Для позитивних результатів при моделюванні схеми керування має відбуватися в такій послідовності: підготовчий етап – занесення значень операндів А і В в регістри, та початкового значення переносу Р в тригер; реалізація циклу додавання – зсув вмісту регістрів та керування тригером затримки. Для цього відповідно задають значення вхідних (інформаційних та керуючих) сигналів.

Лабораторна робота №3

Дослідження арифметико-логічного пристрою

Мета роботи:

знайомство з функціональними можливостями арифметико-логічного пристрою (АЛП) та його дослідження за допомогою сучасних засобів.

Порядок виконання роботи

1. Дослідити АЛП на виконання логічних операцій.
2. Дослідити АЛП на виконання арифметичних операцій.
3. Розширити розрядність АЛП до 16 розрядів.
4. Дослідити розширене АЛП на виконання арифметичних операцій та визначити затримку розповсюдження сигналу затримки.
5. Дослідити розширене АЛП на виконання арифметичних операцій при застосуванні схеми прискореного переносу, визначити затримку.
6. Зробити висновки по роботі.

Методичні вказівки

В лабораторній роботі досліджується АЛП К155ИПЗ (74181). Ця мікросхема призначена для операцій з двома 4-розрядними двійковими словами: $A=A_3A_2A_1A_0$ та $B=B_3B_2B_1B_0$. Конкретний вид операції, що виконується мікросхемою, задається 5-розрядним кодом на входах $MS_3S_2S_1S_0$. Взагалі таке АЛП здатне виконати $2^5=32$ операцій: 16 логічних та 16 арифметичних і арифметико-логічних. Старший розряд вибору операції (вхід М) визначає характер дій, які виконуються АЛП. Коли на цьому вході сигнал високого рівня, АЛП виконує логічні операції порозрядно над кожною парою біт слів А і В. Внутрішній перенос в цьому

режимі не діє. Арифметичні операції виконуються, коли на вході М встановлено низький потенціал, який також дозволяє перенос між розрядами. На виходах F_0 , F_1 , F_2 і F_3 формуються результати логічних перетворень та арифметичних дій. На виході переносу C_{N+4} утворюється сигнал старшого (п'ятого) розряду при виконанні арифметичних дій, він також є інверсним відносно сигналів на входах А та В. Для дослідження затримки в розширеному АЛП операнди А і В підбираються таким чином, щоб виникло перенесення з останнього розряду.

АЛП може працювати з високими або низькими активними логічними рівнями. При цьому змінюються знаки інверсії на входах та виходах, а також утворюються різні таблиці відповідності логічних та арифметичних функцій кодам вибору функції ($S_0 - S_3$). Мікросхема 74181 має активні високі рівні.

Лабораторні роботи № 4 – 5

Дослідження автоматів з жорсткою логікою

Мета роботи:

ознайомитися з принципом побудови пристроїв керування з жорсткою логікою, набути навички синтезу та аналізу схем керування, введення схеми, її налагодження та моделювання за допомогою сучасних засобів.

Порядок виконання роботи

1. У відповідності з варіантом накреслити граф-схему алгоритму роботи керуючого автомата.
2. Відмітити граф-схему станами автомата, закодувати їх та побудувати граф автомата.
3. Побудувати суміщену таблицю переходів і виходів.
4. Згідно з таблицею записати систему функцій збудження для тригерів та рівняння вихідних сигналів. Провести мінімізацію булевих функцій.
5. Побудувати схему керуючого автомата в заданому базисі.
6. Визначити всі можливі шляхи проходження граф-схеми алгоритму і створити набір вхідних сигналів для перевірки працездатності схеми на всіх виявлених шляхах.
7. Провести машинне моделювання схеми керуючого автомата. Визначити основні параметри.
8. Аналіз результатів та висновки. Порівняння параметрів отриманих автоматів Мілі і Мура.

Методичні вказівки

При побудові автомата доцільно скористатись дешифратором станів (74LS138 або 74LS139). В цьому випадку при моделюванні необхідно підключити бібліотеку `ls.vhd`, яка містить моделі вказаних елементів.

Тести для моделювання автоматів потрібно створювати таким чином, щоб показати всі можливі шляхи проходження від початкової вершини до кінцевої. При визначенні параметрів сигнали станів автомата слід згрупувати таким чином, щоб стани на часовій діаграмі було видно також в десятичному еквіваленті.

Лабораторна робота №6

Дослідження запам'ятовувальних пристроїв

Мета роботи:

ознайомитися з принципом роботи запам'ятовувальних пристроїв (ЗП) на прикладі статичного оперативного запам'ятовувального пристрою (ОЗП), побудовою блокового (розширеного) ЗП, провести дослідження ЗП за допомогою сучасних програмних засобів.

Порядок виконання роботи

1. Дослідження мікросхем пам'яті (у відповідності із варіантом завдання) на функціонування:
 - заповнити комірки за всіма адресами довільними даними;
 - прочитати дані з комірок пам'яті, адреси яких зазначені у вашому варіанті завдання;
 - визначити швидкодію ЗП (часу вибірки та часу циклу).
2. Побудувати і дослідити блоковий ЗП:
 - розширити ЗП по розрядності, по кількості слів або по розрядності і кількості слів одночасно (у відповідності з варіантом);
 - записати дані в комірки пам'яті, відповідно до заданих адрес;
 - прочитати записані дані з комірок пам'яті;
 - прочитати дані з порожньої комірки пам'яті.
3. Зробити аналіз результатів і висновки по роботі.

Лабораторна робота №7

Дослідження інтерфейсних ВІС

Мета роботи:

ознайомитись з існуючими типами уніфікованих інтерфейсних ВІС. Дослідити роботу буферного регістра, багаторежимного буферного регістра (ББР) та шинного формувача за допомогою сучасних програмних засобів.

Порядок виконання роботи

1. Дослідити режими роботи ББР (Intel 8212):
 - режим безпосереднього пропускання даних;
 - вхідний буфер;
 - вихідний буфер;
 - пристрій переривання;
 - двонаправлену передачу інформації.
2. Дослідити роботу буферного регістра.
3. Дослідити роботу шинного формувача.
4. Зробити висновки по роботі.

Методичні вказівки

Відповідно до варіанта завдання в лабораторній роботі досліджуються буферні регістри КР580ІР82 та КР580ІР83 (Intel 8282 та 8283), багаторежимний БР К589ІР12 (Intel 8212) та шинні формувачі К589АП16 та К589АП26 (Intel 8216 та 8226), КР580ВА86 та КР580ВА87 (Intel 8286 та 8287).

МЕТОДИЧНІ ВКАЗІВКИ ДО ВИКОНАННЯ КУРСОВОЇ РОБОТИ

Організація курсового проектування здійснюється відповідно до "Положення про виконання курсових проектів та робіт у ВДТУ", затвердженого Ученою радою ВДТУ 30.12.98 р. Матеріал цього розділу може бути використаний при виконанні курсової роботи з дисципліни «ЕОМ і мікропроцесорні системи» студентами денної та заочної форм навчання.

Курсова робота з дисципліни «ЕОМ і мікропроцесорні системи» виконується відповідно до індивідуальних завдань і є самостійною роботою студента, призначеною для закріплення, розширення, узагальнення і практичного використання знань, умінь і навичок, одержаних під час навчання. В процесі курсового проектування студент повинен: розширити і закріпити знання, набуті при вивченні курсу "ЕОМ і мікропроцесорні системи"; навчитися самостійно і творчо застосовувати отримані знання для рішення конкретної інженерної задачі; отримати навички в синтезі обчислювальних пристроїв; опанувати сучасні програмні засоби для моделювання схем; навчитись користуватись довідниковою науково-технічною літературою.

Відповідальність за правильність прийнятих рішень, обґрунтувань, розрахунків та якість оформлення роботи несе студент - автор роботи.

Відповідно до затвердженого графіка студент зобов'язаний своєчасно подавати керівникові результати роботи над курсовою роботою.

Завдання на курсову роботу видаються студентам викладачем на першому плановому практичному занятті. Варіанти завдань генеруються за допомогою спеціальної програми індивідуально для кожного студента і не повторюються.

Завдання можуть бути також спеціалізованими. Спеціалізовані курсові роботи незалежно від об'єкта проектування повинні передбачати схемотехнічне проектування та машинне моделювання спроектованого пристрою, завдання повинно бути узгодженим з керівником і консультантом курсової роботи і затвердженим на засіданні кафедри.

Робота над курсовою роботою

На першому плановому практичному занятті кожному студенту керівником видається індивідуальне завдання, яке студент оформлює на окремому аркуші за встановленою формою (приведена нижче). Після отримання завдання на курсову роботу студенти зобов'язані скласти календарний план роботи, який затверджується керівником. В плані вказуються зміст і строки виконання окремих етапів робіт. При складанні календарного плану необхідно врахувати, що частина матеріалу, необхідного для виконання курсової роботи, розглядається в тому ж семестрі, в якому виконується курсова робота.

Робота виконується студентом самостійно. Забороняється користуватись фрагментами схемотехнічних рішень, виконаних іншими студентами.

ЗАВДАННЯ

на курсову роботу з дисципліни " ЕОМ і мікропроцесорні системи " студента групи ІТТІ-99 Петрова П.П.

Тема роботи: Дослідження автоматів за принципом розподілення сигналів.

Початкові дані: розподільвач сигналів на основі лічильника і дешифратора, побудувати послідовності станів 1-4-9-5-15-3-7-11-15-12-6-2-7 і 10-3-9-7-13-4-11-8-14-2-1-4-6 на основі «І-НІ» логіки, виконати над ними операцію "F = (A + B) додати A" за допомогою арифметико-логічного пристрою; використати для моделювання лічильника сучасні програмні засоби (PCAD, OrCAD та ін.).

Основні розділи пояснювальної записки: титульний лист; зміст; вступ; завдання; аналіз завдання і вибір методу синтезу; синтез заданого вузла; машинне моделювання вузла; оцінка результатів проектування; перелік використаної літератури.

Завдання видано 7.09.2001 р.

Керівник _____ І.І.Іванов

Студент _____ С.С.Сидоров

Зміст пояснювальної записки до курсової роботи

При оформленні тексту пояснювальної записки до курсової роботи слід користуватися рекомендаціями стандарту ДСТУ 3008-95, де встановлені вимоги до оформлення звітної наукової документації.

Текст пояснювальної записки має бути викладений в лаконіч лаконічному обґрунтовальному стилі. Особливу увагу слід звернути на багатоваріантний аналіз на кожному етапі розробки чи дослідження. При викладенні матеріалу в пояснювальній записці даються обґрунтовані пояснення з посиланнями на таблиці, рисунки, діаграми, літературні джерела.

Пояснювальна записка повинна включати такі розділи: титульний лист; зміст; вступ; завдання; аналіз завдання і вибір методу синтезу; синтез заданого вузла; машинне моделювання вузла (з відповідними дослідженнями); оцінка результатів проектування; перелік використаної літератури.

Пояснювальна записка виконується на листах стандартного формату А4 (функціональні схеми вузла і результати моделювання можуть бути представлені на листах більшого формату) згідно з діючими стандартами. Орієнтовний обсяг пояснювальної записки 15-20 листів друкованого тексту. Зразок титульного листа приведений в додатку 1.

Захист курсових робіт

Захист робіт провперед комісією з двох-трьох викладачів при безпосередній участі керівника роботи в присутності студентів групи. В результаті захисту курсова робота оцінюється дванадцятибальною оцінкою і

відповідною їй модульною оцінкою за модульно-рейтинговою системою (для студентів денної форми навчання) в залежності від якості виконання роботи, якості її оформлення, повноти досліджень та рівня відповідей на запитання при захисті роботи. Курсові роботи, виконані не за своїм варіантом завдання, або не в повному обсязі чи з суттєвими помилками, виконані не самостійно (про що свідчить некомпетентність у рішеннях та матеріалах), до захисту не допускаються і направляються керівником роботи на доопрацювання. В цьому випадку у заліковій відомості робиться запис "не допущений", що еквівалентно одержанню оцінки "незадовільно", тобто свідчить про появу академзаборгованості, яка може бути ліквідована на загальних підставах.

Пояснювальні записки, виконані шляхом ксерокопіювання, до розгляду не приймаються.

Основні етапи проектування

Курсова робота полягає в побудові двох послідовностей кодів, над якими необхідно виконати задану арифметичну операцію за допомогою арифметико-логічного пристрою. Оскільки коди в послідовності можуть повторюватись, їх можна реалізувати за допомогою розподільвача сигналів. Узагальнена структура пристрою включає послідовне з'єднання розподільвача сигналів, комбінаційної схеми, АЛП та регістра.

Розподільвач сигналів формує розподілені в часі сигнали на необхідну тривалість циклу. При його побудові потрібно проаналізувати варіанти його реалізації з прямими та інверсними виходами і вибрати кращий за критеріями мінімуму апаратних витрат або мінімальної затримки в комбінаційній схемі, що формує послідовності кодів.

Використання в схемі мікросхем середньої степені інтеграції повинно супроводжуватись аналізом їх функцій.

Рекомендується моделювання схеми пристрою здійснювати поетапно: розподільвач сигналів – формувач послідовності кодів – фрагмент схеми з АЛП – весь пристрій. При цьому звернути увагу на властивості сигналів, що з'являються на виході АЛП.

При дослідженні слід експериментально визначити мінімальний період та максимальну тактову частоту пристрою та затримки на кожному етапі перетворення. Результати досліджень (часові діаграми) потрібно представляти в вигляді, зручному для сприйняття. Наприклад, при визначенні затримок показувати фрагменти часових діаграм в збільшеному масштабі для конкретних сигналів, а для аналізу кодів групувати відповідні сигнали та показувати їх в десятковому форматі. Множина сигналів, яка виноситься на часову діаграму, повинна забезпечувати як зручність сприйняття (бути простою), так і його повноту (включати необхідний мінімум сигналів). При аналізі результатів моделювання необхідно довести, що робота пристрою відповідає варіанту завдання і значення вихідних сигналів задовольняють задану функцію.

ЛИТЕРАТУРА

1. Баранов С.И. Синтез микропрограммных автоматов.- Л.: Энергия, 1979.- 216 с., ил.
2. Вишенчук И.М., Черкасский Н.В. Алгоритмические операционные устройства и супер-ЭВМ.- К.: Тэхника, 1990.- 197 с., ил.
3. Григорьев В.Л. Микропроцессор i486. Архитектура и программирование (в 4-х книгах). – М.: ГРАНАЛ, 1993. – 346 с., ил.
4. Гук М. Аппаратные средства IBM PC. Энциклопедия – СПб: «Издательство «Питер», 1999.- 816 с., ил.
5. Гук М. Процессоры Pentium II, Pentium Pro и просто Pentium.- СПб: Питер, 1999.- 288 с., ил.
6. Каган Б.М. Электронные вычислительные машины и системы: Учеб. пособие для вузов. - М.: Энергоатомиздат, 1985.- 552 с., ил.
7. Колесниченко О.В., Шишигин И.В. Аппаратные средства PC – 3-е изд., перераб. и доп.– СПб.: БХВ – Санкт-Петербург, 2000.– 800 с., ил.
8. Компьютеры на СБИС: В 2-х кн. Кн.1: Пер с япон./ Мотоока Т. и др.- М.: Мир, 1988.- 392 с., ил.
9. Корнеев В.В., Параллельные вычислительные системы.- М.: НОЛИДЖ, 1999.- 320 с., ил.
10. Корнеев В.В., Киселев А.В. Современные микропроцессоры.- М.: НОЛИДЖ, 2000.- 320 с., ил.
11. Кохонен Т. Ассоциативные запоминающие устройства: Пер. с англ.- М.: Мир, 1982.- 384 с., ил.
12. Кравец В.А., Шпильберг А.Я. Зарубежные ЭВМ. Оборудование и программное обеспечение.- Х.: Изд-во "Основа", 1991.- 216 с.
13. К.Дж. Тербер. Архитектура высокопроизводительных вычислительных систем./ Пер. с англ.- М.: Наука, 1985.- 272 с.
14. Майоров С.А., Новиков Г.И. Структура электронных вычислительных машин.-Л.: Машиностроение, 1979. - 384 с., ил.
15. Разевиг В.Д. Система проектирования цифровых устройств OrCAD.- М.: Солон-Р, 2000.- 160 с.
16. Самофалов К.Г. и др. Цифровые электронные вычислительные машины. - Киев: Вища школа, 1983.- 455 с.
17. Системы параллельной обработки./ Пер с англ. Под ред. Д.Ивенса. М.: Мир, 1985.- 416 с., ил.
18. Угрюмов Е.П. Цифровая схемотехника.- СПб.: БХВ – Санкт-Петербург, 2000.- 528 с., ил.
19. Фостер К. Ассоциативные параллельные процессоры: Пер. с англ.- М.: Энергоиздат, 1981. – 240 с., ил.
20. Хвощ С.Т. и др. Микропроцессоры и микроЭВМ в системах автоматического управления: Справочник.- Л.: Машиностроение, 1987.- 640 с., ил.

ДОДАТОК 1

Зразок титульного листа пояснювальної записки до курсової роботи

Міністерство освіти і науки України
Вінницький державний технічний університет

Факультет ІТКІ
Кафедра ПМОС

ДОСЛІДЖЕННЯ АВТОМАТІВ ЗА ПРИНЦИПОМ
РОЗПОДІЛЕННЯ СИГНАЛІВ

КУРСОВА РОБОТА

з дисципліни «ЕОМ і мікропроцесорні системи»

Керівник: доц., ктн. _____ 22.12.2001 р. І.І.Іванов
(Підпис)

Студент гр.ІПЗ-99 _____ 11.12.2001 р. С.С.Сидоров
(Підпис)

Вінниця - 2001

ДОДАТОК 2
Аналоги деяких інтегральних TTL-схем

Тип	Аналог	Функціональне призначення
АГЗ	123	Два одновібратори
АП5	244	Два двоканальних неінвертуючих формувача з трьома станами на виходах
АП6	245	Восьмиканальний двонаправлений формувач з трьома станами на виходах
ИДЗ	154	Дешифратор-демультиплексор на 16 виходів
ИД7	138	Двійковий дешифратор на 8 виходів
ИЕ2	90	Чотирирозрядний двійково-десятковий лічильник
ИЕ4	92	Чотирирозрядний лічильник-дільник на 12
ИЕ5	93	Чотирирозрядний двійковий лічильник
ИЕ6	192	Чотирирозрядний двійково-десятковий реверсивний лічильник
ИЕ7	193	Чотирирозрядний двійковий реверсивний лічильник
ИЕ8	97	Шестирозрядний дільник частоти зі змінним коефіцієнтом ділення
ИЕ9	160А	Чотирирозрядний двійково-десятковий лічильник з асинхронною установкою в нульовий стан
ИЕ10	161А	Чотирирозрядний двійковий лічильник з установкою в нульовий стан
ИЕ11	162А	Чотирирозрядний двійково-десятковий лічильник
ИЕ13	191	Чотирирозрядний двійковий реверсивний лічильник
ИЕ14	196	Двійково-десятковий лічильник з програмуваним коефіцієнтом ділення
ИЕ15	197	Чотирирозрядний лічильник з попередньою установкою
ИЕ16	168	Чотирирозрядний двійково-десятковий синхронний реверсивний лічильник
ИЕ17	169	Чотирирозрядний двійковий синхронний реверсивний лічильник
ИЕ18	163А	Чотирирозрядний двійковий лічильник з синхронною установкою в нульовий стан
ИЕ19	393	Два чотирирозрядних двійкових лічильника з індивідуальною синхронізацією і установкою в нульовий стан
ИК1	AM25505	Схема швидкого множення 2×4
ИК2	381	Чотирирозрядне АЛП– 8 функцій
ИМ2	82	Дворозрядний двійковий суматор
ИМ3	83	Чотирирозрядний двійковий суматор
ИМ5	183	Два однорозрядних повних суматори
ИМ6	283	Чотирирозрядний двійковий суматор з прискореним перенесенням
ИП2	180	Восьмирозрядна схема контролю парності і непарності
ИП3	181	Чотирирозрядне АЛП– 32 функції
ИП4	182	Схема прискореного перенесення

Тип	Аналог	Функціональне призначення
ИП5	280	Дев'ятирозрядна схема контролю парності і непарності
ИП6	242	Чотириканальний двонаправлений інвертуючий формувач з трьома станами на виходах
ИП7	243	Чотириканальний двонаправлений неінвертуючий формувач з трьома станами на виходах
ИП8	261	Двійковий помножувач 4×2 з регістром
ИП10	AM93648	Дванадцятирозрядна схема контролю парності і непарності
ИР1	95	Чотирирозрядний універсальний зсувний регістр з паралельним виходом і роздільною синхронізацією послідовного і паралельного введення інформації
ИР8	164	Восьмирозрядний послідовний зсувний регістр з паралельним введенням інформації
ИР9	165	Восьмирозрядний зсувний регістр з паралельним введенням інформації
ИР10	166	Восьмирозрядний зсувний регістр з послідовно-паралельним введенням інформації і установкою в 0
ИР12	195	Чотирирозрядний універсальний зсувний регістр з паралельним виходом, синхронним послідовно-паралельним введенням інформації і асинхронною установкою в нульовий стан
ИР13	198	Восьмирозрядний реверсивний зсувний регістр з паралельним виходом, синхронним послідовно-паралельним введенням інформації і асинхронною установкою в нульовий стан
ИР15	173	Чотирирозрядний регістр з трьома станами на виходах
ИР21	AM25509	Чотирирозрядний комбінаційний зсувний пристрій з трьома станами виходів
ИР22	373	Восьмирозрядний паралельний регістр з трьома станами на виходах і потенціальним входом синхронізації
ИР23	374	Восьмирозрядний паралельний регістр з трьома станами на виходах і динамічним входом синхронізації
ИР29	323	Восьмирозрядний реверсивний регістр з синхронним послідовно-паралельним введенням інформації, синхронною установкою нульового стану і двонаправленою шиною даних
ИР32	170	4×4 регістровий файл з відкритим колектором на виходах
ИР34	873	Два незв'язаних чотирирозрядних регістри з установкою нульового стану і трьома станами виходів
КП1	150	Селектор-мультиплексор на 16 каналів з інверсним виходом і стробуванням
КП2	153	Два селектори-мультиплексори 4-1
КП5	152	Селектор-мультиплексор восьми каналів з інверсним виходом
КП7	151	Селектор-мультиплексор восьми каналів з прямим і інверсним виходом і стробуванням
КП11	257	Чотири селектори-мультиплексори двох каналів з прямими виходами, спільним адресним входом і трьома станами на виходах

Тип	Аналог	Функціональне призначення
КП12	253	Подвоєний селектор-мультиплексор чотирьох каналів з прямими виходами, загальними адресними входами і роздільним керуванням виходами з трьома станами
ЛА1	20	Два 4І-НІ
ЛА2	30	8І-НІ
ЛА3	00	Чотири 2І-НІ
ЛА4	10	Три 3І-НІ
ЛД1	60	Два чотиривходових логічних розширювача по АБО
ЛД3		Восьмивходовий розширювач по АБО
ЛЕ1	02	Чотири 2АБО-НІ
ЛЕ3	25	Два 4АБО-НІ, що стробуються
ЛЕ4	66	Три 3АБО-НІ
ЛЕ5	28	Чотири буферних 2АБО-НІ
ЛЕ7	260	Два 5АБО-НІ
ЛИ1	08	Чотири 2І
ЛИ3	11	Три 3І
ЛИ6	21	Два 4І
ЛЛ1	32	Чотири 2АБО
ЛН1	04	Шість НІ
ЛН6	366	Шість буферних інверторів з трьома станами на виході
ЛП5	86	Чотири двовходових елементи нерівнозначності
ЛР1	50	Два 2-2І-2АБО-НІ, один розширюється по АБО
ЛР3	53	2-2-2-3І-4АБО-НІ, розширюється по АБО
ЛР4	55	4-4І-2АБО-НІ, розширюється по АБО
ЛР11	51	Два 2-2І-2АБО-НІ і 2-3І-2АБО-НІ
ЛР13	54	2-3-3-2І-4АБО-НІ
РП3	172	Шістнадцятирозрядний 8×2 регістровий ЗП с двома каналами доступу і трьома станами на виходах
РТ1	МС9001	Програмований ПЗП на 1024 біт
РУ1	81	ОЗП на 16 біт (16×1) зі схемами керування
РУ2	89	ОЗП на 64 біт з довільною вибіркою (16×4)
РУ11		ОЗП на 64 біт (16×4) з синхронним записом інформації і з трьома станами на виходах
СП1	85	Схема порівняння двох чотирирозрядних чисел
ТВ1	72	JK-тригер з логікою на 3І на входах JK з занесенням і установкою в нульовий стан
ТВ6	107	Два JK-тригери з установкою в нульовий стан
ТВ9	112	Два JK-тригери із занесенням і установкою в нульовий стан
ТМ2	74	Два D-тригери із занесенням і установкою в нульовий стан
ТМ8	175	Чотири D-тригери з прямими і інверсними виходами і входом установки в нульовий стан
ТМ9	174	Шість синхронних D-тригерів із входом установки в нульовий стан

Навчальне видання

Петух А.М., Обідник Д.Т.

ЕОМ І МІКРОПРОЦЕСОРНІ СИСТЕМИ

Навчальний посібник

Оригінал-макет підготовлено авторами

Редактор В.О.Дружиніна
Коректор З.В.Поліщук

Підписано до друку **7.12.2001р.**
Формат 29,7x42¹/₄ Гарнітура Times New Roman
Друк різнографічний Ум. друк. арк. **661**
Тираж **75** прим.
Зам. № **2001-241**

Віддруковано в комп'ютерному інформаційно-видавничому центрі
Вінницького державного технічного університету
21021, м.Вінниця, Хмельницьке шосе, 95, ВДТУ, ГНК, 9-й поверх
Тел. (0432) 44-01-59