Лекція 14. МІКРОПРОЦЕСОРНА СИСТЕМА. ОСНОВИ ПРОГРАМУВАННЯ МІКРОПРОЦЕСОРА

14.1. Поняття про мікропроцесорну систему.

14.2. Основи програмування мікропроцесора.

14.3. Поняття про команди мікропроцесора.

14.4. Система команд і способи адресації мікропроцесора.

14.5. Мнемонічна форма запису команд.

14.6. Способи адресації команд. Класифікація команд.

14.7. Характеристика команд мікропроцесора.

14.1. Поняття про мікропроцесорну систему

Сукупність ВІС, побудована на принципі програмованої логіки, при якій функціонування цих схем забезпечується занесенням в пристрій пам'яті конкретної програми, що дозволяє вирішувати різноманітні задачі, утворює мікропроцесорний комплект (МПК). Пристрої, побудовані на основі МПК, забезпечили широке використання числових методів у різних сферах виробництва, наукових дослідженнях, побуті.

Мікропроцесорна система може мати яку завгодно конфігурацію обчислювальних засобів. Але МП функціонує тільки разом з складовими частинами, що утворюють МПК. МПК складається з ЦПЕ, пам'яті (П), пристроїв вводу-виводу (В/Вв) тощо, з`єднаних з допомогою шин.

Під шинами розуміють сукупність провідників, яким одночасно може передаватись інформація.

МП використовує адресну шину для вибору комірки пам'яті або пристрою вводу-виводу. Вибравши інформацію по даній адресі з допомогою адресної шини, МП пересилає її по шині даних.

Пам'ять з'єднана з МП шинами адресними (ША), даних (ШД), керування (ШК).

Дані, які поступають по адресних лініях, використовуються для ідентифікації області пам'яті і з якими МП необхідний зв`язок. Адресні лінії (шини) передають дані тільки з мікропроцесора, в зворотньому напрямку передача заборонена.

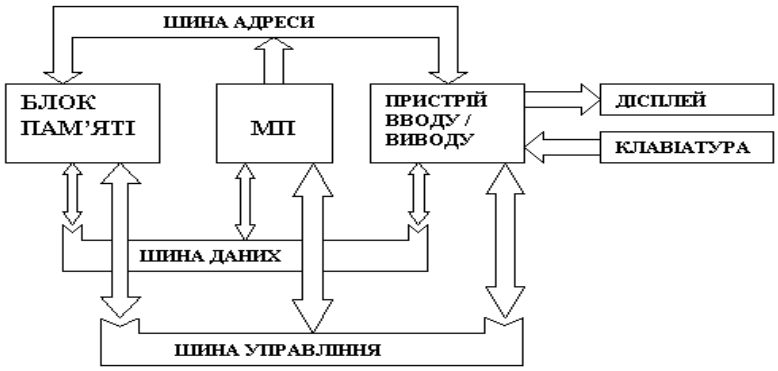

Рис. 14.1. Структурна схема мікропроцесорного комплекту (МПК)

По лініях (шинах) даних у визначений момент часу передається одне слово даних в МП, або в зворотньому напрямку. Шина даних є двонаправлена. По цих лініях передаються слова даних з внутрішньої шини МП в пам`ять або блок вводу-виводу, чи навпаки.

З допомогою ліній (шин) керування МП "каже" зовнішнім пристроям, коли подавати дані на шину даних, або коли одержати їх з шини.

Кожному слову даних пам`яті відповідає якийсь визначений номер - адреса його місцезнаходження. Адреса використовується всякий раз, коли необхідно записати слово даних в пам'ять, або прочитати слово з пам’яті.

Довжина слова даних, кількість адресованих слів пам'яті і швидкість виконання команд характеризують здатність мікропроцесора обробляти дані, яку називають *потужністю мікропроцесора*.

Довжина слова даних фіксована - 8, 16, 32 і більше біт (розрядів). Біт - це значення 0 чи 1 в двійковій системі числення. (Системи числення розглядають на практичному занятті).

Восьми бітове слово називають байтом.

Кожному слову даних присвоюється номер його місцезнаходження, тобто *адреса*. Чим більше значення максимальної адреси, тим більша обчислювальна потужність МП. Діапазон адресації 8-бітового слова дорівнює 256 словам адрес пам’яті, а 16-бітового слова - 65536 (64 КБ словам).

Швидкодія мікропроцесора оцінюється тривалістю в часі циклу “вибірки-виконання” для одного кроку програми. Час циклу “вибірка-виконання” залежить від тактової частоти, що визначається генератором тактових імпульсів. Тактова частота сучасних мікропроцесорів досягає до 200 і більше МГц, що дозволяє виконувати команди за декілька сотих долей мікросекунди.

Сучасні мікропроцесорні системи можуть мати різну конфігурацію. Часом в одній мікросхемі розміщено центральний процесорний елемент, частину пам’яті або інші елементи. Проте при виготовленні будь-якої системи всі елементи цієї системи повинні відлагоджуватись в комплекті мікросхем.

Блок пам`яті складається з ОЗП, ПЗП.

ОЗП - оперативно-запам’ятовуючий пристрій, використовується для зберігання програми користувача і для визначення робочої області записаних на виконання оперативних програм.

ПЗП - постійно-запам`ятовуючий пристрій; містить закодовану в машинних кодах програму монітора, яка керує всіма елементами системи і визначає порядок роботи системи.

Для забезпечення зв’язку оператора з приладом, останній може мати пристрій індикації, який відображує інформацію про біжучу адресу і дані в шістнадцятковому або десятковому коді на індикаторі дисплеї, а також периферійний пристрій – клавіатуру, яка служить для вводу інформації в ОЗП і для керування операціями.

Клавіатура з’єднується з системою через блок паралельно-послідовного інтерфейсу (ППІ). Через інтерфейс периферійного обладнання здійснюється зв`язок з зовнішніми пристроями, до яких можуть належати накопичувачі інформації на магнітних носіях тощо.

Всіма операціями системи керує програма монітор, записана в ПЗП. Всі функції монітору визначають з допомогою простих операцій клавіатури. Використовуючи набір команд конкретного МП, користувач може записати свою програму і дані в ОЗП і виконати її, попередньо проводячи наладку з допомогою відповідного режиму.

Міжмодульні зв’язки і обмін інформацією між модулями здійснюється з допомогою колективних шин (магістралей) адреси, даних і керування.

Периферійне обладнання (пристрої вводу-виводу, дисплей, клавіатура або зовнішня пам’ять) підключаються до шин мікропроцесора не безпосередньо, а через відповідні інтерфейсні схеми, мікросхеми контролера прямого доступу до пам’яті КПДП, програмований паралельний інтерфейс ППІ, програмований послідовний інтерфейс зв’язку ППІЗ, програмований контролер пріоритетних переривань ПКПП, програмований таймер ПТ тощо.

В МП обробляються машинні слова, представлені у вигляді даних і команд. Машинні слова можуть бути представлені в двійковому чи двійково-десятковому коді. Командне слово завжди складається з коду операцій та операндів. Код операції - це закодована назва операції, яку повинен виконати МП. Сукупність кодів операції утворює набір команд конкретного типу МП. Операнд - адреса інформаційного слова, або саме інформаційне слово чи адреса - місцезнаходження машинного слова в пам’яті.

14.2. Основи програмування мікропроцесора

Програмування МП має багато спільного з програмуванням мікро (ЕОМ). *Програмування* – опис послідовності дій, які МП повинен виконати для вирішення поставленої задачі, причому список команд повинен бути абсолютно точним по кількості і порядку виконання.

Вибір послідовності операцій, які забезпечують вирішення поставленої задачі, здійснює програміст.

Програмне забезпечення – це сукупність програм і документації, яка використовується процесором для вирішення прикладних задач.

Алгоритм – це результат опису вирішення задачі. Алгоритм можна закодувати, тобто представити в виді послідовності машинних команд, які здійснюють рішення поставленої задачі. Для кожної машини одного типу є свій специфічний набір команд, тому програміст повинен писати програму, користуючись тільки такими командами.

Після написання програми, програму треба відлагодити, тобто потрібно цю програму закласти в машину і спробувати її виконати.

Скласти програму – це значить описати алгоритм на мові МП (ЕОМ), тобто шляхом застосування команд, які машина може виконати.

Запис алгоритму шляхом застосування команд називається *кодуванням*, а використовувані для цього команди – *вихідним кодом*.

Так, як для функціонування машини необхідні двійкові коди, вихідний код мусить бути трансльований (переведений) в *об’єктний код*, який є двійковим представленням вихідного коду.

14.3. Поняття про команди мікропроцесора

Так, як мікропроцесор це пристрій програмованої обробки даних, то здійснюється програмованою чи схемною логікою, то треба навчитись складати програми обробки даних.

Програма це послідовність команд, виконання яких процесором приводить до розв’язку задачі. Програма складається з команд і операндів з визначенням точних адрес, де вони знаходяться.

Команда визначає операцію, яку має виконати МП над даними (операндами). Команда містить у явній, або неявній формі інформацію про те, як буде виконана операція, які частини процесора будуть задані, адресу де буде розміщено результат операції, адресу наступної команди.

Команда визначається форматом, який конкретно вказує поля команди і склад команди.

Формат команди містить операційну та адресну частину, тобто код операції (КОП).

КОП	Ознака адресації	Адреси операндів

Загальна кількість бітів у КОП називають довжиною формату. Довжина КОП $m > \log_2 M$, де M – кількість різних операцій довжина адреси одного операнда $n > \log_2 S$, де S – кількість адрес комірок пам’яті.

Довжина команди визначає швидкість виконання команди і залежить від способу адресації команди. Адресація – це спосіб визначення даних. Адресація буває: пряма, непряма, безпосередня (регістрова), уявна, автоінкрементарна (автодекрементарна), сторінкова (віртуальна), індексна (стекова) (Intel, 68010 Motorola), відносна.

Програмування мікропроцесорів здійснюється часто мовою Асемблер.

Програмування мовою Ассемблер, яка є алгоритмічною мовою програмування низького рівня, є менш ефективним, ніж мовами високого рівня, такими як Бейсік, Фортран, Паскаль, C^{+}, C^{++}. Алгоритмічні мови високого рівня максимально наближені до математичного запису задачі набором операторів, що полегшує програмування і не вимагає знань особливостей організації обчислювальних пристроїв, об'єму пам'яті і її адрес, систем числення та кодування, набору мнемокоманд тощо. Під час програмування мовою Ассемблер витрачається набагато більше часу, ніж аналогічної програми мовою високого рівня, наприклад, для обчислення добутку двох чисел мовою високого рівня вживається один оператор присвоєння, а мовою Ассемблер - це близько десяти команд. Крім того багато часу займає процес асемблерування - перетворення мнемокоманд у машинні коди двійкової чи шістнадцяткової систем числення, коли кожна мнемокоманда транслюється в одну машинну команду. Для полегшення цієї процедури можна використовувати спеціальну програму транслятор-наприклад, TASM-COM, яка програму в мнемокодах перетворює в програму машинних кодів. Але така програма є не у всіх мікропроцесорних обчислювальних пристроях, бо для її зберігання потрібна постійна пам'ять відповідного об'єму.

Під час програмування в машинних кодах необхідно стежити за розподілом пам'яті, доступної для записування-читання, а саме: який її об'єм, початкова і кінцева адреси, глибина стекової пам'яті, якщо така використовується. А це пов'язано з конкретним мікропроцесорним обчислювальним пристроєм, його технічними даними, тому програма в машинних кодах не є універсальною для всіх МПС. Якщо в програмному забезпеченні є службова програма асемблерування, то в ній передбачено отримання так званої об'єктної програми, що переміщується. Таку програму можна завантажити в будь-яку область пам'яті конкретного МПС, бо в ній всі адреси формуються відносно деякої базової адреси, а вже спеціальна системна команда налагоджує програму на конкретні адреси. Якщо такої службової програми немає, то зручно складати програму в мнемокодах з букво-цифровими символами, які мають мнемонічний зміст, замість адрес. Така програма буде універсальною для цього типу мікропроцесора, а для інших МПС символічні адреси необхідно замінити на дійсні.

14.4. Система команд і способи адресації мікропроцесора

Команда мікропроцесора – це таке двійкове слово, яке будучи “прочитаним” мікропроцесором, заставляє МП виконувати якісь конкретно визначені дії. Інші двійкові слова подібних дій в МП не викликають. Більшість команд здійснює пересилання або обробку даних, розташованих в пам'яті або в одному з регістрів МП.

Якщо говорять про набір команд процесора, то розуміють під цим всі його команди, включаючи допоміжні.

Довжина команди як двійкового слова співпадає з довжиною слова даних. Так довжина слова 8р. і 16 р. МП має відповідно 8 і 16 біт, може бути і 24 біт.

Для виконання команда пересилається в регістр команд, дешифратор і схеми керування МП, де вона ідентифікується, в результаті чого формуються сигнали, які направляються в інші частини мікропроцесора. З допомогою цих сигналів виконуються операції, які визначає команда.

Мікропроцесор завантажує команду в регістр команд на протязі циклу вибірки. Протягом наступного за ним циклу виконання МП декодує команду і створює сигнали керування процесом виконання операції цієї команди.

Команда повинна нести інформацію: 1) що робити МП (додавання, очистка, зсув і т. д.); 2) вказувати адресу, тобто місцезнаходження даних (і які дані необхідно обробити). Наприклад – додавати до акумулятора якусь суму з комірки пам’яті, помістити її туди, очистити аккумулятор тощо.

Команда складається з 2-ох частин, - коду операції (КОП) і адреси даних, або самих даних. КОП – вказує МП, що робити. Адреса – вказує на місцезнаходження даних, які приймають участь в операції.

Якщо довжина команди складає 2, 3 слова, то обов’язково перше слово код операції; але команди, що складаються тільки з одного слова, не безадресні.

14.5. Мнемонічна форма запису команд

В зв’язку з тим, що двійкові слова тяжко запам’ятовуються, то для скорочення форми запису можна було б використовувати 8-му або 16-ту системи числення. Але знов таки є складність для програміста, що означає кожна команда, яка буде виражена в такій формі. Для того, щоб було зручно при написанні програм, використовують скорочену форму запису назви команд - *мнемонічну*, тобто три чи чотири букви латинського алфавіту, що відповідають скороченому запису назви дії команди на англійській мові, наприклад JMP - безумовний перехід “стрибок” вимагає вказівки адреси переходу до безумовного виконання.

Така форма запису і використовується в мові Асемблер.

На кожен тип МП є опис відповідних команд з відповідним набором команд і вказівкою кодів операцій як в мнемоніці, так і в числовій формі.

Система команд процесора КР580 має коди операції, які класифікуються за різними ознаками – довжині команди (числу байтів), функціональними ознаками (які операції виконує), способу адресації тощо.

14.6. Способи адресації команд. Класифікація команд

Під способом адресації, або режимом адресації розуміють спосіб визначення даних, які беруть участь в операціях, або тип звернення до даних (адресації). Спосіб адресації – спосіб визначення операндів. Для того, щоб скласти програму важливо знати особливості процедур, які дозволяють перетворювати інформацію про адреси команди і дані в фізичні адреси ділянок пам'яті машини.

Для МП КР580 існують 4 способи адресації – неявна або регістрова, безпосередня, пряма і непряма.

А) *Регістрова адресація* (*неявна*). *Неявна* – адреси джерела і приймача даних вказані неявно. Має таку особливість, що код команди і дані адреси знаходяться в 1 байті інформації.

Особливість. 1-байтні команди не адресуються до даних, які завантажують в регістр, регістрову пару, або даними, які зберігаються в області пам’яті, адреса якої знаходиться в регістровій парі.

Приклад. 1-байтова команда пересилання даних з регістру А в регістр В має двійковий код 10001111, де

1	0	0	0	1	1	1	1
КОП		адреса рег.В			адреса рег.А		

Адреси джерела даних А і приймача В задані неявно і говорять, що вони, адреси, вбудовані в команду неявно, тому і адресація неявна.

Так як МП виконує цикл вибірки і виконання, то в випадку 1-байтової команди виконуються тільки два мікроцикли. Тому цей тип операції є найбільш швидкодіючим.

Б) *Безпосередня адресація* – найбільш економічний спосіб зберігання і пошуку інформації. Необхідні дані після байту КОП має сама команда (слідом за КОП йдуть дані, які займають 1 чи 2 байти). Дані зберігаються в 2-му чи 2 і 3 му байтах команди. Тут не треба вказувати адресу пам'яті, необхідний тільки код операції і записати дані, які вказує програміст.

В) *Пряма адресація* – в 2-му і 3-му байтах команди після КОП записується повна 16 р. адреса пам'яті, а дані зберігаються в комірці пам'яті під такою адресою.

Г) *Непряма* (*регістрова*) *адресація* – при якій визначається регістрова пара, наприклад В, С, що записується за кодом команди. Це зберігаються не дані, а повна 16 р. адреса комірки пам'яті, в якій зберігаються ці дані. Старший байт записується в 1 регістр, молодший в другий регістр. Такі команди вимагають попередньої підготовки для запису в ці комірки даних. Наприклад в МПК КР580.

14.7. Характеристика команд мікропроцесора

До характеристики команди відносяться:

1. Призначення.
2. Використовувані способи адресації.
3. Мнемонічний запис на Асемблері.
4. Розміщення команди в пам'яті.
5. Представлення дії, які виконуються з допомогою логічних символів.
6. Пояснення дії, які виконуються командою.
7. Довжину команди.
8. Вплив дії результату виконання команди на регістр стану МП.

Необхідно мати точний опис всіх 8 пунктів для кожного типу процесора.