**Структура процесорного ядра МК**

Основними характеристиками, які визначають продуктивність процесорного ядра МК, є:

* набір регістрів для збереження проміжних даних;
* система команд процесора;
* способи адресації операндів у просторі пам'яті;
* організація процесів вибірки і виконання команди.

З погляду системи команд і способів адресації операндів процесорне ядро сучасних 8-розрядних МК реалізує один із двох принципів побудови процесорів:

* процесори з CISC-архітектурою, які реалізують так звану повну систему команд (Complicated Instruction Set Computer);
* процесори з RISC-архітектурою, що реалізують скорочену систему команд (Reduced Instruction Set Computer).

**Система команд процесора МК**

Так само, як і в будь-якій мікропроцесорній системі, набір команд процесора МК містить у собі чотири основні групи команд:

* команди пересилання даних;
* арифметичні команди;
* логічні команди;
* команди переходів.

Для реалізації можливості незалежного управління розрядами портів (регістрів) у більшості сучасних МК передбачена також група команд бітового управління (булевий або бітовий процесор). Наявність команд бітового процесора дозволяє істотно скоротити об'єм коду управляючих програм і час їхнього виконання.

У ряді МК виділяють також групу команд управління ресурсами контролера, які використовуються для настроювання режимів роботи портів вводу/виводу, управління таймером і т.п. У більшості сучасних МК внутрішніх ресурсів контролера відображаються в пам'яті даних, тому для мети управління ресурсами використовуються команди пересилання даних.

Система команд МК у порівнянні із системою команд універсального МП має, як правило, менш розвинуті групи арифметичних і логічних команд, зате більш потужні групи команд пересилання даних і управління. Ця особливість пов'язана зі сферою застосування МК, яка вимагає, насамперед, контролю навколишнього оточення і формування управляючих впливів.

**4.2.3. Схема синхронізації МК**

Схема синхронізації МК забезпечує формування сигналів синхронізації, необхідних для виконання командних циклів центрального процесора, а також обміну інформацією з внутрішньою магістраллю. У залежності від виконання центрального процесора командний цикл може містити в собі від одного до декількох (4 - 6) тактів синхронізації. Схема синхронізації формує також мітки часу, необхідні для роботи таймерів МК. До складу схеми синхронізації входять дільники частоти, які формують необхідні послідовності синхросигналів.

### Пам'ять програм і даних МК

У МК використовується три основних типи пам'яті. Пам'ять програм являє собою постійну пам'ять (ПЗП), призначену для збереження програмного коду (команд) і констант. Її вміст у ході виконання програми не змінюється. Пам'ять даних призначена для збереження змінних у процесі виконання програми і являє собою ОЗП. Регістри МК - цей вид пам'яті містить у собі внутрішні регістри процесора і регістри, що служать для управління периферійними пристроями (регістри спеціальних функцій).

**Пам'ять програм**

Основною властивістю пам'яті програм є її енергонезалежність, тобто можливість збереження програми при відсутності живлення. З погляду користувачів МК варто розрізняти наступні типи енергонезалежної пам'яті програм:

* **ПЗП масочного типу - mask-ROM**. Вміст комірок ПЗП цього типу заноситься при її виготовленні за допомогою масок і не може бути з часом заміненим або допрограмованим. Тому МК із таким типом пам'яті програм варто використовувати тільки після досить тривалої дослідної експлуатації. Основним недоліком даної пам'яті є необхідність значних витрат на створення нового комплекту фотошаблонів і їхнє впровадження у виробництво. Зазвичай такий процес займає 2-3 місяці і є економічно вигідним тільки при випуску десятків тисяч приладів. ПЗП масочного типу забезпечують високу надійність збереження інформації з причини програмування в заводських умовах з наступним контролем результату.
* **ПЗП, програмовані користувачем, з ультрафіолетовим стиранням - EPROM (Erasable Programmable ROM)**. ПЗП даного типу програмуються електричними сигналами і стираються за допомогою ультрафіолетового опромінення. Комірка пам'яті EPROM являє собою МОН-транзистор з "плаваючим" затвором, заряд на який переноситься з керуючого затвора при подачі відповідних електричних сигналів. Для стирання вмісту комірки вона опромінюється ультрафіолетовим світлом, яке надає зарядові на затворі транзистора енергію, яка достатня для подолання потенційного бар'єра і стікання на підкладку. Цей процес може займати від декількох секунд до декількох хвилин. МК із EPROM допускають багаторазове програмування і випускаються в керамічному корпусі з кварцовим віконцем для доступу ультрафіолетового світла. Такий корпус коштує досить дорого, що значно збільшує вартість МК. Для зменшення вартості МК із EPROM його укладають у корпус без віконця (версія EPROM з однократним програмуванням).
* **ПЗП, однократно програмовані користувачем, - OTPROM (One-Time Programmable ROM)**. Являють собою версію EPROM, виконану в корпусі без віконця для зменшення вартості МК на його основі. Скорочення вартості при використанні таких корпусів настільки значне, що останнім часом ці версії EPROM часто використовують замість масочних ПЗП.
* **ПЗП, програмовані користувачем, з електричним стиранням - EEPROM (Electrically Erasable Programmable ROM)**. ПЗП даного типу можна вважати новим поколінням EPROM, у яких стирання комірок пам'яті відбувається також електричними сигналами за рахунок використання тунельних механізмів. Застосування EEPROM дозволяє стирати і програмувати МК, не знімаючи його з плати. Таким чином можна проводити налагодження і модернізацію програмного забезпечення. Це дає величезний виграш на початкових стадіях розробки мікроконтролерних систем або в процесі їх вивчення, коли багато часу іде на пошук причин непрацездатності системи і виконання циклів стирання-програмування пам'яті програм. За ціною EEPROM займають середнє місце між OTPROM і EPROM. Технологія програмування пам'яті EEPROM допускає побайтове стирання і програмування комірок. Незважаючи на очевидні переваги EEPROM, тільки в рідких моделях МК така пам'ять використовується для збереження програм. Це пов'язано з тим, що, по-перше, EEPROM мають обмежений об'єм пам'яті. По-друге, майже одночасно з EEPROM з'явилися Flash-ПЗП, що при подібних споживчих характеристиках мають меншу вартість;
* **ПЗП з електричним стиранням типу Flash - Flash-ROM**. Функціонально Flash-пам'ять мало відрізняється від EEPROM. Основна відмінність полягає в способі стирання записаної інформації. У пам'яті EEPROM стирання відбувається окремо для кожного комірки, а в Flash-пам'яті стирати можна тільки цілими блоками. Якщо необхідно змінити вміст однієї комірки Flash-пам'яті, буде потрібно перепрограмувати весь блок. Спрощення декодувальних схем у порівнянні з EEPROM привело до того, що МК із Flash-пам'яттю стають конкурентноздатними по відношенню не тільки до МК з однократно програмованими ПЗП, але і з масочними ПЗП також.

**4.3.2. Пам'ять даних**

Пам'ять даних МК виконується, як правило, на основі статичного ОЗП. Термін "статичне" означає, що вміст комірок ОЗП зберігається при зниженні тактової частоти МК до як завгодно малих значень (з метою зниження енергоспоживання). Більшість МК мають такий параметр, як "напруга збереження інформації" - USTANDBY. При зниженні напруги живлення нижче мінімально допустимого рівня UDDMIN, але вище рівня USTANDBY робота програми МК виконуватися не буде, але інформація в ОЗП зберігається. При відновленні напруги живлення можна буде проініціалізувати МК і продовжити виконання програми без утрати даних. Рівень напруги збереження складає зазвичай близько 1 В, що дозволяє в разі потреби перевести МК на живлення від автономного джерела (батареї) і зберегти в цьому режимі дані ОЗП.

Об'єм пам'яті даних МК, як правило, невеликий і складає переважно десятки і сотні байт. Цю обставину необхідно враховувати при розробці програм для МК. Так, при програмуванні МК константи, якщо можливо, не зберігаються як змінні, а заносяться в ПЗП програм. Максимально використовуються апаратні можливості МК, зокрема, таймери. Прикладні програми повинні орієнтуватися на роботу без використання великих масивів даних.

**4.3.3. Регістри МК**

Як і всі МШС, МК мають набір регістрів, що використовуються для управління його ресурсами. У число цих регістрів входять звичайно регістри процесора (акумулятор, регістри стану, індексні регістри), регістри управління (регістри управління перериваннями, таймером), регістри, що забезпечують ввід/вивід даних (регістри даних портів, регістри управління паралельним, послідовним або аналоговим вводом/виводом). Звертання до цих регістрів може відбуватися по-різному.

У МК із RISC-процесором усі регістри (часто й акумулятор) розташовуються за адресами, що явно задаються. Це забезпечує більш високу гнучкість при роботі процесора.

Одним з важливих питань є розміщення регістрів в адресному просторі МК. У деяких МК усі регістри і пам'ять даних розташовуються в одному адресному просторі. Це означає, що пам'ять даних сполучена з регістрами. Такий підхід називається "відображенням ресурсів МК на пам'ять".

В інших МК адресний простір пристроїв вводу/виводу відділено від загального простору пам'яті. Окремий простір вводу/виводу дає деяку перевагу процесорам з гарвардською архітектурою, забезпечуючи можливість зчитувати команди під час звертання до регістра вводу/виводу.

**4.3.4. Стек МК**

У мікроконтролерах ОЗП даних використовується також для організації виклику підпрограм і обробки переривань. При цих операціях уміст програмного лічильника й основних регістрів (акумулятор, регістр стану й інші) зберігається і потім відновлюється при поверненні до основної програми.

У фон-нейманівській архітектурі використовується єдина область пам'яті, у тому числі, і для реалізації стека. При цьому знижується продуктивність пристрою, тому що одночасний доступ до різних видів пам'яті неможливий. Зокрема, при виконанні команди виклику підпрограми наступна команда вибирається після того, як у стек буде поміщений уміст програмного лічильника.

У гарвардській архітектурі стекові операції відбуваються в спеціально виділеній для цієї мети пам'яті. Це означає, що при виконанні програми виклику підпрограм процесор з гарвардською архітектурою проводить кілька дій одночасно.

Необхідно пам'ятати, що МК обидвох архітектур мають обмежену ємність пам'яті для збереження даних. Якщо в процесорі існує окремий стек і об'єм записаних у нього даних перевищує його ємність, то відбувається циклічна зміна вмісту покажчика стека, і він починає посилатися на раніше заповнену комірку стеку. Це означає, що після занадто великої кількості викликів підпрограм у стеці виявиться неправильна адреса повернення. Якщо МК використовує загальну область пам'яті для розміщення даних і стека, то існує небезпека, що при переповненні стека відбудеться запис в область даних або буде зроблена спроба запису даних, що завантажуються в стек, в область ПЗП.

**4.3.5. Зовнішня пам'ять**

Незважаючи на існуючу тенденцію щодо переходу до закритої архітектури МК, у деяких випадках виникає необхідність підключення додаткової зовнішньої пам'яті (як пам'яті програм, так і даних).

Якщо МК містить спеціальні апаратні засоби для підключення зовнішньої пам'яті, то ця операція відбувається штатним способом (як для МП).

Другий, більш універсальний, спосіб полягає в тому, щоб використовувати порти вводу/виводу для підключення зовнішньої пам'яті і реалізувати звертання до пам'яті програмними засобами. Такий спосіб дозволяє задіяти прості пристрої вводу/виводу без реалізації складних шинних інтерфейсів, однак приводить до зниження швидкодії системи при звертанні до зовнішньої пам'яті.