**3. Основні характеристики сімейства мікроконтролерів АVR**

Однокристальні AVR-мікроконтролери представляють 8-розрядні високопродуктивні RISC-контролери загального призначення.

Особливістю AVR — контролерів є їх широка номенклатура, яка дозволяє користувачеві вибрати мікроконтролер з мінімальною апаратною надмірністю і, отже найменшою вартістю.

Так, наприклад, в номенклатуру групи AT90S входять прилади з ПЗП об'ємом від 1 до 8 Кбайт, з самими різними наборами периферії і в корпусах з кількістю виводів від 8 до 48. Зараз в серійному виробництві знаходяться три сімейства AVR — Tiny, Classic і Mega. Мікроконтролери Tiny — найбільш дешеві і прості, Mega — найбільш потужні, а Classic займають проміжне місце між ними[2,351].

**3.1 Основні характеристики AVR-контролерів**

**Classic** – основна лінія контролерів з продуктивністю до 16 MIPS, пам'ять програм FASH ROM 2-8Кбайт, пам'ять даних EEPROM 64-512 байт, пам'ять даних SRAM 128-512 байт.

**Tiny** – мікроконтролери низької вартості у 8-вивідному виконанні, мають вбудовану систему контролю напруги живлення.

**Mega** – продуктивність 4-6 MIPS, пам'ять програм FASH ROM 64-128Кбайт, пам'ять даних EEPROM 64-4096 байт, пам'ять даних SRAM 1-4 Kбайт, вбудований 10-розрядний 8-канальний АЦП, апаратний помножувач.

В принципі, всі мікроконтролери побудовані за однією схемою.

Система управління, що складається з лічильника команд і схеми декодування, бере на себе зчитування та декодування команд з пам'яті програм, а операційний пристрій відповідає за виконання арифметичних і логічних операцій; інтерфейс вводу / виводу дозволяє обмінюватися даними з періферійними пристроями, і, нарешті, необхідно мати запам'ятовуючий пристрій для зберігання програм і даних (рис.3.1).

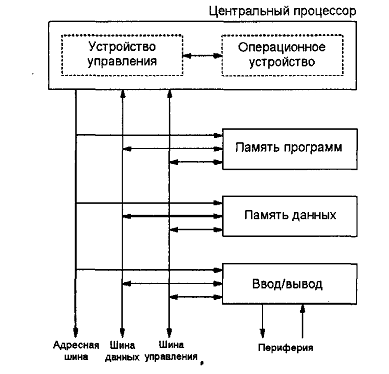


Рис.3.1 Базова структура мікроконтролера

Операційний пристрій, як правило, складається з арифметико-логічного пристрою (АЛП), нагромаджуючого суматора і декількох допоміжних регістрів. У класичній програмі майже половина всіх команд - це команди пересилання (move) для передачі даних від допоміжних регістрів до накопичення сумматору і назад. У сімействі мікроконтролерів AVR накопичувальний суматор, що представляє собою "тонке місце", стає не настільки критично важливим, завдяки застосуванню 32-х робочих регістрів, пов'язаних з блоком АЛП. В результаті арифметичні і логічні операції можуть бути виконані протягом єдиного такту[7,159].

**3.2 Арифметико-логічний пристрій**

Операції АЛП можна розділити на три основні категорії: арифметичні, логічні і порозрядного. Кожній з цих категорій відповідають потужні команди. Деякі (нові) мікроконтролери сімейства AVR мають теж апаратні помножувачі в арифметичному блоці АЛП.

**3.3 Структура команд**

Незважаючи на RISC-архітектуру, запас команд не так вже і обмежений: як ні як, AT90S8515, базової серії, має 118 команд, у вищій мірі оптимізованих з точки зору доцільності та ефективності, а самий "слабкий" представник сімейства - AT90S1200 - чи сильно "відстає від лідера" зі своїми 89 командами. У мікроконтролерах AVR базової серії всі команди мають однакову ширину слова 16 біт (тобто, 2 байти). Виключенням є тільки дві команди для прямої адресації статичної пам’яті даних Ids і sts, що складаються з двох слів і, відповідно, з 4 байт.

За небагатьма винятками (команда переходу, операції з безпосередньою адресацією і пересилання в пам'ять / з пам'яті) всі команди обробляються протягом одного такту системної синхронізації.

**3.4 Виконання команди**

На рис.3.2. показано зіставлення кількості необхідних тактових імпульсів осцилятора для виконання двох послідовних команд процесорами 68НС05, 80С51, а також сімействами мікроконтролерів PIC та AYR.

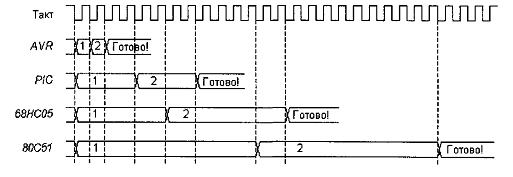


Рис.3.2. Порівняння часу виконання команд різними процесами

У запам'ятовуючих пристроях, відповідних класичної концепції фон Неймана, дані і команди зберігаються в одній пам'яті. На противагу цьому, пам'ять по Гарвардській архітектурі, використовуваної в мікроконтролерах АVR складається з декількох компонентів. В даному випадку розділені пам'ять команд і пам'ять даних, тобто, звернення до команд здійснюється незалежно від доступу до даних [14,98].

**У мікроконтролерах AVR окремі сегменти пам'яті влаштовані по-іншому також і фізично.**

• Пам'ять програм реалізована на основі програмованої і електрично зтирається флеш-технології. У всіх мікроконтролерах AVR пам'ять 16 - розрядна (двухбайтовая), і завжди знаходиться "на кристалі". Розширення пам'яті програм за допомогою блоків EPROM або флеш неможливо.

• Внутрішня пам'ять для енергозалежних даних (тобто, даних, які будуть втрачені після відключення робочої напруги), являє собою статичну пам'ять RAM (SRAM). Перевага цього полягає:

• в тому, що відпадає всяка потреба у внутрішній регенерації як у випадку з деякими іншими процесорами, які використовують динамічну пам'ять. З цієї причини мікроконтролери AVR можуть працювати з тактами аж до 0 Гц. На додаток до цього, деякі мікроконтролери AVR для збільшення обсягу оброблюваних даних можуть працювати з підключеною зовнішньою пам'яттю SRAM. Для цього, однак, доводиться жертвувати дорогоцінними контактами введення / виводу портів А і С.

• Для даних, які повинні зберегтися після відключення робочої напруги, в розпорядженні є мікросхема EEPROM (Electrically EPROM - електрично стирані ППЗУ). В пам'ять EEPROM можна записувати дані під час нормального виконання програми. Для області EEPROM також немає обов'язкової необхідності в програмуючих пристроїв[6,258].

**3.5 Область вводу / виводу**

У мікроконтролерах AVR область введення / ведення від $ 20 до $ 5F. У мікроконтролерах AVR функції вводу / виводу для взаємодії зі зовнішнім світом застосовні не тільки для портів (від А до D), але також і для всіх регістрів стану і управління таких "вбудованих в кристал" периферійних функцій як таймер, пристрій UART, сторожовий таймер, доступ на запис і читання до пам'яті EEPROM, інтерфейс SPI і т.д.

**3.6 Переривання і підпрограми**

Техніка підпрограм і переривань у мікроконтролерах AVR може бути застосована звичайним чином. Перехід до частин програми, які в процесі її виконання обробляються багаторазово або повинні міститися окремо по причині структурування, здійснюється за допомогою команди call. Це - команда переходу, яка поміщає в стек наступну команду нормального виконання програми в якості адреси повернення. Після виконання підпрограми адресу вилучають із стека за допомогою команди ret, і програма продовжується.

Стек для адрес повернення перебуває в пам'яті SRAM. Виняток становить тільки мікроконтролер AT90S1200, який застосовує для цієї мети апаратний стек за принципом LIFO (Last In, First Out - "останнім увійшов, першим вийшов").

У прикладі на рис. 3.3 показник стека ініціалізується, починаючи з мітки Start, після чого слід мітка Саll1 першого виклику підпрограми UPI. В якості адреси повернення в стек міститься адреса наступної команди, тобто, $ 78А. Сама підпрограма UP1 в представленому прикладі не виконує ніяких дій, і відбувається негайне повернення до головної програми. Для цього зі стека витягується адресу, і виконання програми продовжується з команди Са112. З мітки Cal12 відбувається повторний виклик підпрограми UP1. У цьому випадку в стек міститься адреса повернення $ 78D.

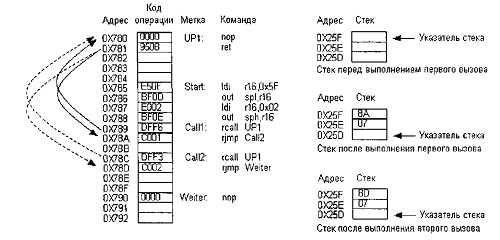


Рис 3.3 Два виклику підпрограми

У правій частині рис. 3.3 показаний фрагмент статичної пам'яті БІАМ, відведеної для стека. Зазначено вміст стека перед виконанням першого і після виконання першого і другого викликів. У правій частині рис. 3.3 показаний фрагмент статичної пам'яті БІАМ, відведеної для стека. Зазначено вміст стека перед виконанням першого і після виконання першого і другого викликів. Підпрограми також можуть бути вкладеними (тобто, з підпрограми можна викликати іншу програму) [4,147].

Особливою формою підпрограм є підпрограми обробки переривань, за допомогою яких реалізовані заздалегідь не заплановані звернення з програм. Іншими словами, програма, що виконується асинхронно по відношенню до події, зухвалому переривання. Тут, наприклад, мова може йти про роботу таймера, прийомі байта через прийомопередавач і АVR або запиті на переривання від зовнішнього пристрою.

Залежно від побудови мікроконтролера, у розпорядженні базової серії АVR є до 12 векторів переривання (адрес входу).

**Периферійні функції.**

З точки зору апаратної частини, у мікроконтролерах АVR реалізовано безліч периферійних функцій. Так, наприклад, АТ9088515 володіє наступними особливостями:

* 32 програмувальні лінії введення /виведення;
* програмований пристрій;
* синхронний інтерфейс SPI;
* 8-розрядний таймер / лічильник;
* 16-розрядний таймер / лічильник з функціями порівняння та захоплення та можливістю реалізації двох виходів ШИМ (широтно-імпульсний модулятор);
* "Інтегрований сторожовий таймер;
* аналоговий компаратор.

**3.7 Блок-схема мікроконтролерів AT90S1200 і AT90S8515**

Архітектура сімейства мікроконтролерів AVR показана на прикладі її найбільш простого в даний час представника AT90S1200 (рис. 3.4) і самого продуктивного мікроконтролера AT90S8515 (рис. 3.5).

Два інших представники сімейства - мікроконтролери AT90S2313 і AT90S4414 - оптимізовані з точки зору розміру елементів і / або ціни і надають у розпорядження ту ж кількість функцій і модулів, що і мікроконтроллер AT90S8515.

**3.8 Конструктивне виконання корпусів і розташування виводів**

Розташування висновків чотирьох представників базової серії мікроконтролерів AVR показано на рис. 3.6 і рис. 3.7. Розглянемо призначення висновків докладніше.

**VCC**

Підведення напруги живлення

**GND**

заземлення

**Port А (РА0. .. РА7)**

Порт А являє собою двонаправлений порт вводу / виводу з пропускною здатністю 8 біт. Буфер виводу порту А в режимі прийому даних у стані приймати струм силою до 20 мА і, завдяки цьому, безпосередньо живити, наприклад, світлодіоди. Кожен висновок порту може бути налаштований індивідуально як вхід або вихід, а при виконанні функції введення до нього, при бажанні, можна підключати підтягує опір[10,123].

В якості особливої функції через порт А працює демультіплексірованная шина передачі даних і адрес, якщо до мікроконтролера AVR повинна бути підключена зовнішня пам'ять RAM.

**Port В (РВ0. .. РВ7)**

Порт В являє собою двонаправлений порт вводу / виводу (I / O) з пропускною здатністю 8 біт. Буфер виводу порту В у режимі прийому даних у стані приймати струм силою до 20 мА і, завдяки цьому, безпосередньо живити, наприклад, світлодіоди. Кожен висновок порту може бути налаштований індивідуально як вхід або вихід, а при виконанні функції введення до нього, при бажанні, можна підключати підтягує опір.

Альтернативно, через порт В можуть виконуватися також різні спеціальні функції (таймер, підключення входів аналогового компаратора, інтерфейс 8Р1).

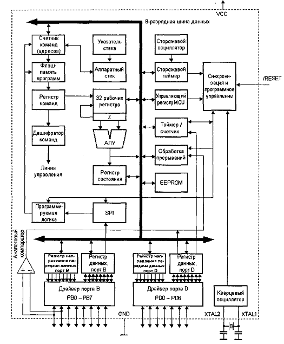


Рис.3.4 Блок-схема мікроконтролерів AT90S1200

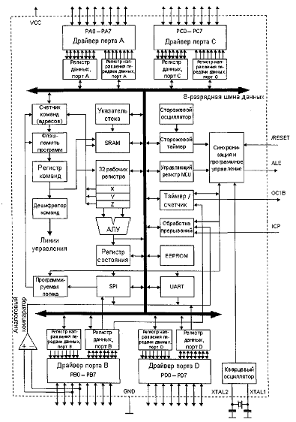


Рис. 3.5 Блок-схема мікроконтролерів AT90S8515

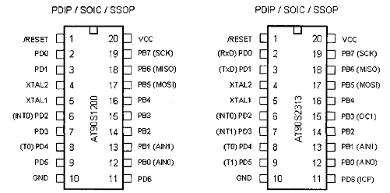


Рис. 3.6 Розміщення виводів мікроконтролерів

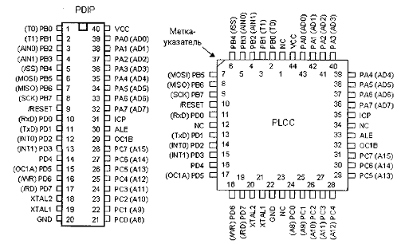


Рис.3.7 Конструктивне виконання корпусів і розташування виводів і AT90S4414 і AT90S8515

**Port С (РС0. .. РС7)**

Порт С являє собою двонаправлений порт вводу / виводу (I / O) з пропускною здатністю 8 біт. Буфер виводу порту С в режимі прийому даних у стані приймати струм силою до 20 мА і, завдяки цьому, безпосередньо живити, наприклад, світлодіоди. Кожен висновок порту може бути налаштований індивідуально як вхід або вихід, а при виконанні функції введення до нього, при бажанні, можна підключати опір.

В якості особливої функції через порт С виводиться старший байт адресної шини, якщо до мікроконтролера AVR повинна бути підключена зовнішня пам'ять RAM.

**Port D (PDO. .. PD7)**

Порт D являє собою двонаправлений поріг введення / виведення з пропускною здатністю 8 біт. Буфер виводу порту D в режимі прийому даних у стані прийому струм силою до 20 мА І, завдякі цьому, безпосередньо живить, наприклад, світлодіоді. Коженов висновок порту може бути налаштований індівідуально Як вхід або вихід, а при виконанні функції введення до нього, при бажанні, можна підключати опір[15,148].

Альтернативно, через порт D можуть виконуватися також різні додаткові функції (наприклад, надходження запитів на переривання, передача вихідних даних таймера, інтерфейс з пристроєм UART).

**RESET**

Висновок для подачі сигналу скидання. Рівень балка 0 на цьому виводі на протязі мінімум двох циклів системного такту Ф при активному осцилляторі переводить мікроконтроллер в початковий стан.

**ICP**

Висновок функції "Захоплення" (Capture) інтегрованого таймера / лічильника Т / С 1.

**ОС1В**

Висновок функції "Порівняння" (Compare) інтегрованого таймера / лічильника Т/С1.

**ALE**

Висновок для подачі імпульсу при записі молодшого адресного байта з демультиплексованої шини даних / адреси через порт А в зовнішній фіксуючий регістр, коли до мікроконтролера AVR підключена зовнішня пам'ять RAM (див. рис. 3.6, рис. 3.7 і рис. 3.8). Байт даних передається на другому кроці звертання до пам'яті RAM через порт А.

**XTAL1**

Вхід інтегрованого осциллятора для вироблення такту системної синхронізації Ф і, рівним чином, вхід для зовнішнього тактового сигналу, якщо внутрішній осцилятор не застосовується[13,165].

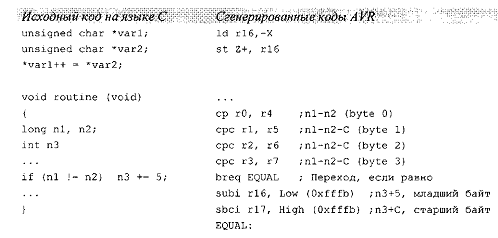
**XTAL2**

Вихід інтегрованого осциллятора для вироблення такту системної синхронізації Ф.

**3.9 Програмування для AVR на мові високого рівня Сi++**

У зв'язку з тим, що мікроконтролери AVR були повністю новою розробкою, і відпала необхідність враховувати сумісність із якими-небудь попередниками, з'явилася можливість скористатися найсучаснішими досягненнями. Так, наприклад, із самого початку в проектування і розробку структур мікроконтролерів AVR були залучені фахівці високого рівня - перш за все, на мові програмування Сi++. У результаті була отримана архітектура процесора, спеціально розрахована на складання програм на мові Сi++.

Нижче на прикладі двох коротких фрагментів програм показано, як ефективно елементи мови Сi++ можуть бути перетворені в коди асемблера для мікроконтролерів AVR.



У першому прикладі показана типова операція з покажчиком в тому вигляді, в якому вона дуже часто зустрічається в мові програмування Сi++.

У другому прикладі ni і п2 - цілі числа типу Long Integer, тобто, обидва мають довжину 4 байти. ПЗ - це цілочисельна змінна довжиною 2 байти. Значення ni, п2 і ПЗ визначені усередині підпрограми як локальні змінні.

У коді, створеному на асемблері, число ni розміщується в регістрах г 3-г 0, п2 - в регістрах г 7 - г 4, а ПЗ - в регістрах г 17-р 16.

Крім того, у другому прикладі показано, що відсутні в наборі AVR-команд команди addi (add immediate - "додати безпосереднє значення") і ici (add immediate with carry - "додати безпосереднє значення з урахуванням перенесення") для обчислення виразу ПЗ 5 можуть бути замінені відніманням другого компонента з 5[11,254].