*Совин Ярослав Романович*

*Лаби: Стахів Марта Юріївна*

Розраха – 10, лаби – 30 (13, 17), практичні – 10 (5, 5)

2МК(25,25)

Ви 4-й курс, на лекції у вас вільне відвідування. Не перевіряю, але записую)))

На практичні обов’язково.

Змістяра

[Тема: мікропроцесори та основні поняття. Класифікація мікропроцесорів. 3](#_Toc254688519)

[Класифікація мікропроцесорів 3](#_Toc254688520)

[>>>> за призначенням: 3](#_Toc254688521)

[>>>> за типом архітектури: 5](#_Toc254688522)

[>>>> за типом системи команд розрізняють: 5](#_Toc254688523)

Лекція №1

Список літератури:

1. Мікропроцесорна техніка – Ю.І.Якименко
2. Евстифеев А.В. Микроконтроллеры

Для практичних

1. Лебедев М.Б. CodeVisionAVR – пособие для начинаючих
2. Хартов В.Я. – Прикроконтроллери

# Тема: мікропроцесори та основні поняття. Класифікація мікропроцесорів.

Перший МП i4004 створений фірмою Intel у 1971р. Дядьки Тед Хофф та Фредерік Феггін.

**Мікропроцесор** (МП) – це пристрій, який здійснює прийом, обробку і видачу інформації. Конструктивно МП містить одну або декілька інтегральних схем і виконує дії за програмою, записаною в пам’яті.

**Мікропроцесорна схема** (МПС) – обчислювальна, контрольно-вимірювальна або керувальна система, в якій основним пристроєм обробки інформації є МП.

**Мультипроцесорна система** – система, яка утворюється об’єднанням деякої кількості універсальних або спеціалізованих МП, завдяки чому забезпечується паралельна обробка інформації і розподілене керування

**Мікропроцесорний комплект** (МПК) – сукупність інтегральних схем, сумісних за електричними, інформаційними та конструктивними параметрами і призначених для побудови електронно-обчислювальної техніки та мікропроцесорних систем керування. Chipset – в англійській мові. До складу МПК входять:

1. Мікропроцесор;
2. Оперативні запам’ятовуючі пристрої (ОЗП;RAM);
3. ПЗП (ROM);
4. Інтерфейси або котроллери зовнішніх пристроїв;
5. Службові (тактовий генератор, регістри, шинні формувачі, контроллери шин, арбітри шин).

**Мікроконтроллер** –пристрій, який виконаний в одному корпусі інтегральної схеми і містить основні складові МПК: процесор, пам’ять даних, пам’ять програм, програмовані інтерфейси.

**Інтерфейс** – визначає правила взаємодії компонентів системи, а також правила побудови технічних засобів призначених для реалізації цієї взаємодії.

## Класифікація мікропроцесорів

Є 3 параметри: призначення, тип архітектури, тип системних команд.

### >>>> за призначенням:

1. Універсальні – найпоширеніші є Intel і AMD. Займають до 90% всього ринку. 10% - Motorolla, Sun microsystems. На сьогодні мають розрядність від 32 до 64 біт, деякі 128.
2. Спеціалізовані
	1. Мікроконтроллери
		1. 8-розрядні
		2. 16-розрядні
		3. 32-розрядні
	2. Мультимедійні процесори
	3. Трансп’ютери
	4. Цифрові сигнальні процесори (DSP – Digital Signal Processors) – призначені для цифрової обробки сигналів в реальному масштабі часу. Є два основні алгоритми обробки сигналів: *цифрова фільтрація* і *спетральний аналіз*. Основною операцією цифрової обробки сигналів є MAC – перемноження з додаванням
		1. З фіксованою комою (16,24 біт)
		2. З плаваючою (32 біт)

Гордон Мур – один з засновників Intel. Має свій емпіричний закон: кожні 2 роки кількість транзисторів на 1-му кристалі подвоюється. Це правило до сих пір виконується.

Основні вимоги до ЦСП:

* апаратна підтримка операцій MAC(за 1 такт)
* наявність високошвидкісних інтерфейсів вводу-виводу даних

#### Мікроконтроллери

Призначені для застосування у вбудованих системах і орієнтовані на виконання функцій керування і регулювання. Перший був розроблений Intel в 1986р. Найбільшими є сімейство AVR.

Для мікроконтроллерів характерні:

1. Шифрокі функціональні можливості
2. Низька ціна
3. Високі технічні характеристики

Мікроконтроллер складається:

1. 8,16,32бітного процесора
2. Внутрішньої пам’яті даних і програм
3. Набір інтерфейсних або переферійних модулів (порти вводу-виводу, таймери, АЦП, ЦАП)

8-бітні – найбільш чисельна група. Мають низьку продуктивність і малу вартість. Тобто орієнтована на масову продукцію, де кожен цент впливає на сумарну ціну. Для них об’єм пам’яті програм становить від 1-100кБ, об’єм пам’яті даних – від 10 байт до кількох кБ. Вони підтримують обмежений набір команд (50-100 і малу кількість способів адресацій)

16-бітні – переважно покращені версії 8-бітних контроллерів. Мають більший об’єм пам’яті, розширені систему команд і способи адресацій. Інколи є програмно сумісними з 8-бітними мікроконтроллерами.

32-бітні – по своїм можливостям відповідають молодшим моделям універсальних мікропроцесорів. Серед 32-бітних мікроконтроллерів домінують мікроконтроллери з архітектурою **ARM**(Advanced RiSC-Machine).

#### Мультимедійні мікропроцесори

Призначені для обробки аудіо- та відеоінформації.

* Універсальні процесори, в які добавлені спеціальні команди – MMX, 3Dnow!, SSE
* Суто мультимедійні процесори

#### Трансп’ютери

Утворений від «Транзистор+комп’ютер», призначені для роботи у системах з масовими паралельними обчисленнями. Тобто це мікрокомп’ютер(процесор) з власною внутрішньою пам’яттю та каналами (links) для з’єднання з іншими трансп’ютерами. Вони можуть паралельно вести обчислення і обмін інформацією.

### >>>> за типом архітектури:

1. МП з прінстонською (фоннейманівською) архітектурою
2. МП з гарвардською архітектурою

### >>>> за типом системи команд розрізняють:

1. CISC(*Complete Instruction Set Computing*)-процесори – процесори з повним набором команд. В них швидкодія визначається тактовою частотою. Якщо потрібно збільшити продуктивність – піднімаємо частоти;
2. RISC(*Reduced -//-*)-процесори – процесори зі скороченим набором команд. Висока продуктивність досягається за рахунок технології опрацювання даних;
3. VLIW(Very Large Instruction Word)-процесори – процесори з довгим командним словом. В них одна команда (до 128 біт) викликає паралельне виконання кількох операцій у різних блоках мікропроцесора.

2010-03-02

Лекція №2

# Тема: Мікропроцесорний пристрій

Схемка з нікона(МПП):

До складу МПП входять: ЦП (CPU). ОЗП, ПЗП, ПРВВ (пристрої вводу/виводу), Інтерфейси I/O.

Розглянемо їх детальніше.

**Шина** – це інформаційний канал, який об’єднує блоки МП або МПП і забезпечує обмін даними між ними. Тобто шина може бути внутрішня і зовнішня. Складається з *n* провідників. У паралельній шині інформація передається по кожному провіднику одночасно.

Усі блоки МПП з’єднюються паралельною шиною, яка називається **системною шиною**(System Bus). Вона складається з: шини адреси (ША), шини даних (ШД), шини керування (ШК).

1. Шина адреси (Address Bus) – призначена для передавання адреси комірки пам’яті або ПРВВ, з якими мікропроцесор буде обмінюватися інформацією. Шина адреси є однонаправленою, напрям передачі – від мікропроцесора, до пам’яті або зовнішнього пристрою.



1. Шина даних – призначена передачі даних між блоками мікропроцесорного пристрою. Вона є двонаправленою і позначається:



Розрядність шини даних визначає розрядність мікропроцесора і може бути 8,16,32,64

1. Шина керування – призначена для передачі керуючих сигналів, які узгоджують роботу мікропроцесора з пам’яттю і ПРВВ.

ПЗП і ОЗП складають систему пам’яті, призначену для зберігання, видачі та запису інформації. Тобто в ПЗП зберігаються дані, які не повинні знищуватися при зникненні живлення – програми. ОЗП – проміжні результати обчислення.

Пам’ять організована у вигляді масиву комірок, кожна з яких має свою адресу і містить *M* біт інформації. Переважно M=8 (байт), або M=16 (слово). Якщо мікросхема має NA адресних входів, то загальна кількість доступних комірок пам’яті $N\_{M}=2^{N\_{A} }$.

Наприклад, NA=8,M=8, тоді $N\_{M}=2^{8}=256$. Записуємо як 256x8.

Для позначення об’єму пам’яті використовують такі одиниці:

$2^{10}$ - Кіло, $2^{20}$ - Мега, $2^{30}$ - Гіга, $2^{40}$ - Тера.

Наприклад, NA=16,M=8,$N\_{M}=2^{N\_{A}}\*8біт=2^{16}\*8біт=2^{10}⋅2^{6}\*8біт=64Кбайт$.

**Структура мікросхеми пам’яті**

****

Адресні входи підключаються до шини адреси. Для звернення до певної комірки пам’яті мікропроцесор повинен виставити її адресу на шину адресів. Через двонаправлені входи-виходи даних, які підключені до шини даних здійснюється запис або читання інформації з/в задану комірку пам’яті. Входи управління задають: тип операції(R/W), режим роботи(активний/неактивний/енергозберігаючий)

|  |
| --- |
| Сигнали управління мікросхем пам’яті |
| $$\overbar{CS}/\overbar{CE}$$ | $$\overbar{OE}$$ | $$\overbar{WE}$$ | Стан виводів D7-D0/IO7-IO0 | Режим роботи |
| 1 | x | x | Високоімпедансовий стан | Неактивний |
| 0 | 0 | 1 | Виходи | Зчитування |
| 0 | 1 | 0 | Входи | Запис |
| 0 | 1 | 1 | Високоімпедансовий стан | Неактивний |



CS – Chip Select, CE – Chip Enabled

*Основою ЦП* є мікропроцесор але також можуть входити *тактовий генератор, буфери шин* і т.д.

## Функціонування МПП

Робота МПП полягає в обробці даних і в обміні інформації з ПРВВ.

Модуль ЦП обробляє дані і керує всіма компонентами системи. Протягом часу виконання команди(командний цикл), ЦП виконує такі дії:

* Виставляє адресу команти на ША;
* Отримує від команти з пам’яті та дешифрує його;
* При потреб обчислює адреси операндів та зчитує їх;
* Виконує операцію, яка задана командою;
* Сприймає зовнішні керувальні сигнали (наприклад, запити переривань);
* Генерує сигнали стану і керування, потрібні для роботи пам’яті та ПРВВ.

## Основні характеристики ЦП

1. *Система команд*, показує скільки різних команд може виконувати мікропроцесор.
2. *Розрядність мікропроцесора*, показує скільки біт інформації передається або обробляється за одну команду.
3. *Об’єм адресованої пам’яті(адресний простір)*, показує до скількох комірок пам’яті може звертатися мікропроцесор і визначається адресною шиною. Якщо є 16 розрядна шина, то 64 Кб може адресувати процесор
4. *Частота,* показує скільки елементарних операцій виконується за одну секунду

## Інтерфейси I/O

Виконують ф-ю узгодження сигналів пристроїв вводу/виведення із сигналами системної шини. Їх ще називають контроллерами.

## Архітектурні особливості мікропроцесорів

Архітектура МП – це комплексна характеристика, яка включає: структурну схему мікропроцесора, систему команд МП, призначення вхідних та вихідних сигналів, формати даних і команд, способи адресації до даних.

### Прінстонська архітектура

Характеризується використанням спільної пам’яті для зберігання програм і даних. Для звертання до цієї пам’яті використовується спільна системна шина.

**Переваги:**

* Оскільки є спільна пам’ять, то в залежності від задачі ми можемо *оперативно розприділяти область* між програмами і даними.
* Використовують одну спільну системну шину, є менше з’єднань, більш ***висока надійність***, простота відладки, тестування

**Недолік:**

* Необхідність послідовної вибірки команд і даних по спільній шині, при цьому шина стає вузьким місцем, що обмежує продуктивність системи.

### Гарвардська архітектура

Характеризується фізичним розділеннямпам’яті програм і пам’яті даних. Кожна пам’ять з’єднюється з мікропроцесором окремою шиною. Це дає змогу одночасно проводити вибірку команди і зчитування даних. Тобто якщо такий процесор на протязі одного такту виконує більше дій.

**Переваги:**

* Зростання швидкодії

**Недоліки:**

* Більша кількість шин, важча відладка
* Не можемо розприділяти пам’ять під потрібні задачі, у нас фіксований розмір пам’яті для програм і даних

Ця архітектура переважає в цифрових сигнальних процесорах та мікроконтроллерах, а Прінстронська переважає в універсальних мікропроцесорах.

## CISC- та RISC-мікропроцесори

**CISC-процесори** – процесори з повним набором команд, виконують великий набір різноформатних команд з використанням різних способів адресації. Чим більше команд, тим простіше. З одної сторони це полегшує роботу, але з іншої строни складніший стає пристрій керування. *Якщо Pentium – понад 2200 різних команд, 11 способів адресації. Тут пристрій керування займає 60%.*

Компілятори з усієї системи команд використовують обмежений набір простих команд, типу *регістр-регітр*, *регітр-пам’ять*.

**RISC-процесори** – процесори з скороченм набором команд фіксованого розміру, при цьому значно скорочується число способів адресації.

**Переваги:**

* Спрощується структура процесора
* Зменшується розмір і вартість кристалу
* Зростає продуктивність

**Недоліки:**

* Немає команд деяких і потрібно використовувати інші команди для реалізації необхідних

В RISC процесорах для зменшення звертань до пам’яті збільшена кількість внутрішніх регістрів(від 32-100). В CISC від 8-16 регістрів.

### Конвеєр команд

В RISC-процесорах застосовують **конвеєр команд**. Кожна команда розбивається на деяку кількість кроків з однаковим часом виконання

F – вибірка команди

D – дешифратор команди

R – зчитування при потребі даних, які входять в команду

E – виконання самої команди

Ці кроки можуть запускатися паралельно із зсувом на 1 крок.

2010-03-09

Лекція №3

Програма мікропроцесора складається з послідовності команд, які зберігаються в пам’яті(у двійковій формі) і виконуються одна за одною.

**Пневмонічне позначення команд** використовується для:

1. Скороченого запису команди – тобто від кожної команди зазвичай беруть перші букви (для додавання – **ADD**ition)
2. Полегшення запам’ятовування
3. Ототожнення команди з їх фактичним призначенням

STS 0x0045, R30

STS – Store to SRAM

Пневмонічна форма запису використовується в мові асемблера. **Асемблером** називається:

1. Мова програмування низького рівня;
2. Спеціальна програма-транслятор, яка переводить пневмокоди у двійкові значення.

## Система команд

Кожен тип мікропроцесора має свій індивідуальний набір команд (система команд). *Команда* представляє собою двійкове слово, а структуру цього слова називають *форматом команди*. Команда може складатися з одного-двох-трьох машинних слів. *Машинне слово* має таку ж розрядність як і мікропроцесор.

Будь-яка команда складається з коду операції - ***КОП*** (Opcode – Operation Code) і ***адрес*** джерел і приймачів даних.

### Команди з трьохадресним форматом

Складаються з КОП; двох адрес джерел даних – *src1, src2*; однієї адреси приймача *dest*.

|  |  |  |  |
| --- | --- | --- | --- |
| Opcode | Src1 | Src2 | Dest |

Такий формат команд характерний для процесорів з архітектурою Register-Register (*aka* Load-Store), в яких будь-які операції, які передбачають їх завантаження в регістр.

### Команди з двоадресним форматом

|  |  |  |
| --- | --- | --- |
| Opcode | Dest | Src |

Такий формат характерний для процесорів з архітектурою *Register-Memory*, коли один операнд знаходиться в регістрі, а інший – в пам’яті.

### Команди з одноадресним форматом

КОП + адреса джерела даних

|  |  |
| --- | --- |
| Opcode | Src |

Такий формат характерний для процесорів з акумуляторною архітектурою, коли один з операндів завжди розсташовується в спеціальному регітрі-акумуляторі.

|  |
| --- |
| Псевдокод для виконання операції додавання |
| Register-Register | Register-Memory | Accumulator |
| Load Reg1, ALoad Reg2, BAdd Reg3, Reg1, Reg2Store C, Reg3 | Load Reg1, AAdd Reg1, BStore C, Reg1 | Load AAdd BStore C |

## Класифікація команд мікропроцесора за призначенням

Всі команди поділяються на 5 груп:

1. Переміщення даних
	1. Reg <-> Reg
	2. Reg <-> Memory
	3. Reg <-> I/O
2. Перетворення даних
3. Управління програмами – змінюють вміст внутрішнього регістра мікропроцесора, який називається програмний лічильник PC(Program Counter). Регістр PC вказує адресу наступної команди, яка буде виконуватися.
4. Вводу-виводу
5. Спеціальні – виконують специфічні функції, пов’язані з архітектурою мікропроцесора

Якщо у мові високого рівня

If (a == 0){

}else{

}

То в асемблері:



*Команди вводу/виводу* організовують обмін інформації із зовнішніми пристроями

i8080: IN 53 <- ввести дані в акумулятор із зовнішнього вводу в регістр пам’яті 53

AT9052313: OUT PORT B, R18 <- Тобто вміст регістра R18 виводиться в порт B

i8080: EI (Enable Interrupt)

AT9052313: SLEEP (перейти в режим енергозбереження)

## Типова архітектура універсального мікропроцесора



1. АЛП – призначений для виконання і обробки даних (БР – буферний регістр, цифри знизу – розрядність). Містить 2 буферних регістра і регістр-акумулятор. Один з операндів завжди розсташовується в акумуляторі, результат виконання операції так само розсташовується в акумуляторі.
2. Множина регістрів
3. Пристрій керування
4. Внутрішня шина даних

РЗП: D,C,D,E,H,L

РСП: F,A БА, БД, SP, PC/IP

F – *регістр прапорців (регістр стану)*, служить для зберігання результатів деяких перевірок, які здійснюються в процесі виконання програми.

### Формат F регістра

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| F | S | Z | x | AC | x | P | x | C |

1. Розряд перенесення C(Carry) – встановлюється в 1, якщо результат виходить за межі 8 біт
2. Прапорець нуля Z(Zero) – встановлюється в 1, якщо результат виконання операції є нульовий
3. Знаковий розряд S(Sign) – вказує знак результату. Якщо S=0, то додатній, S=1, то від’ємний.
4. Розряд парності P(Parity) – встановлюється в 1, якщо кількість результатів є парною.
5. Розряд додаткового переносу AC(Auxilary Carry) – встановлюється в 1, якщо є позика перенесення з 3го розряду в 4й.

Приклад.

MVI A, 26; A=2610=000110102

MVI D, 42; D=4210=001010102

ADD D; A <- A+D

MVI – заносить число, вказане в команді в адресу.



C=0, Z=0, S=0, P=1, AC=1

Пр.2.

MVI A, 11; A=1110=000010112

MVI D,-5; D=-510= 

ADD D; A <- A+D

Пр.3

MVI A, 7; A=710=000001112

MVI D, 15; D=1510=000011112

SUB D;



C=1; Z=0; S=1; P=0; AC=1

### Лічильник команд

Зберігає адресу команди, що буде виконуватися. Розрядність лічильника відповідає розрядності шини адреси. По сигналу початкової установки *Reset* в *Program Counter* заноситься наперед визначена адреса, для даного проца, 0x0000 і виконання програми починається з цієї адреси.

Після читання першої команди PC автоматично отримує приріст на одиницю і переходить на виконання наступних команд.

*Румпуруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруруррурурур [чорт, збився].*

Вміст програмного лічильника можна змінювати за допомогою команд умовного і безумовного переходів або виклику підпрограм

**Буферний регістр адрес (БА)**

Вихід буфера адреси утворює шину адреси. Вміст буфера завантажується з програмного лічильника C.

**СІД – схема інкременту/декременту**

Вона дозволяє без участі АЛП збільшити або зменшити на одиницю вміст РЗП(регістрів загального програчення) PC(Program Counter) або SP(Stack Pointer).

**М** – мультиплексор, з’єднює один з РЗП з внутрішньою шиною даних.

**SP** – вказівник стеку, вказує адресу верхньої комірки пам’яті, яка є початком області, відведеної під стек. Вміст SP автоматично зменшується на 1 при занесені даних в стек, і автоматично збільшується на 1 при читанні даних в стек.

PUSH, POP – команди роботи зі стечком

**PK** – регістр команд, зберігає код команди, яка виконується. З регістра код вказує на дешифратор команд, який задає дії процесора для виконання команд. Розрядність регістра команд відпвідає розрядності мікропроцесора.

**ПРК** – пристрій керування, відповідно до дешифрованого коду команди генерує внутрішні і зовнішні керувальні сигнали.

ВШД – внутрішня шина даних, вона забезпечує передачу даних всередині мікропроцесора. Кожен блок мікпроцесора підключений до ВШД, але може скористатися нею тільки після отримання відповідного сигналу від *ПРК*.

**Програмною моделлю мікропроцесора** називається сукупність регістрів, які можна змінити/прочитать за допомогою команд.



## Основні способи адресації у мікпропроцесорах

Спосіб адресації визначає яким чином відбувається звертання до даних в команді.

Неявна адресація – при такій адресації місцезнаходження операнду і результату фіксовано, переважно на якийсь спеціалізований регістр МП.



Регістрова адресація – операнди знаходяться в регістрах загального призначення мікропроцесора.



***Безпосередня адресація*** – дані задаються всередині команди. Для позначення часто використовується буква I (*Immediate* – безпосередньо).

MVI E, 5CH (занести в регістр E 5 в 16му коді)

При **прямій** у команди безпосередньо вказується адреса комірки пам’яті, у якій знаходяться дані



Непряма адресація – використовує регістрові бари (індексні регістри), вміст яких є адресою даних в пам’яті



Непряма адресація має декілька підвидів - з автодекрементом або автоінкрементом. Збільшення/зменшення індексного регістра може відбуватися перед виконанням команди (preінкремент/preдекремент) або після (-post)

Пр.

LD R17, Y+

Завантажити в R17 вміст комірки за адресою Y, після чого Y збільшується на 1.

2010-03-16

Лекція №4

### Непряма адресація зі зміщенням

Адреса операнда визначається як сума вмісту індексного регістра та зміщення вказаного в команді.



### Відносна адресація

У лічильник команд вноситься сума поточного значення Program Counter’а і зміщення вказаного в команді.



### Відносна регістрова адресація

В лічильник команд заноситься вміст регістру загального призначення



|  |  |  |
| --- | --- | --- |
| **Спосіб адресації** | **Команда** | **Опис** |
| Неявна | load | Заноситься вміст регістра спеіального призначення Reg |
| Регістрова | load | Завантажити в R1 вміст R2 |
| Безпосередня | load | Завантажити в R1 якусь const.Тобто R1 <- const |
| Пряма | load R1, (const)[[1]](#footnote-1) | R1 <- Mem(const) |
| Непряма | load R1, (R2) | R1 <- Mem(R2) |
| Непряма з постінкрементом | Load R1, (R2)+ | R1 <- Mem(R2); R2 <- R2+1 |
| Непряма з постдекрементом | Load R1, (R2)- | R1 <- Mem(R2); R2 <- R2-1 |
| Непряма з преінкрементом | Load R1, +(R2) | R2 <- R2+1; R1 <- Mem(R2) |
| Непряма з предекрементом | Load R1, -(R2) | R2 <- R2-1; R1 <- Mem(R2) |
| Непряма адресація зі зміщенням | Load R1, disp(R2) | R1 <- Mem(R2 + disp) |
| Зміщення | Branch disp | P <- P + disp |
| Відносна регістрова | Branch R1 | P <- R1 |

# Структура мікроконтроллерів загального призначення

Особливості мікроконтроллерів. Система команд орієнтована на задачі керування і регулювання. Відповідно у них менш розвинуті арифметичні і логічні команди, але більш розвинуті команди переміщення даних і управління програмою. Також в мікроконтроллерах присутні бітові команди.

Система і пристрої на базі Intel мають *низьку вартість* – більшість тих компонентів системи є реалізовані в межах одного кристалу (може містити АЦП, ЦАП, таймери, лічильники); *універсальність* і *можливість розширення* значно нижче, ніж в мікропроцесорах загального призначення; *модульна організація*, при якій на базі одного процесорного ядра проектується ряд(лінійка), які відрізняються об’ємом і типом пам’яті програм, об’ємом пам’яті даних, набору периферійних модулів і частотою синхронізації.

## Процесорне ядро



ВКМ – внутрішня контрольна магістраль

Основні характериситики процесорного ядра МК:

* Набір регістрів
* Система команд
* Способи адресації операндів
* Організація процесів вибірки і виконання команд

**CISC:** великий набір команд, час виконання від 3 до 12 тактів

**RISC:** набір команд скорочений до мінімуму; всі команди фіксованої довжини; виконуються за 1 такт

**Продуктивність** – вимірюють в MIPS (Million Instructions Per Second)

$$MIPS=\frac{1}{t\_{команд}}$$

## Схема синхронізації

Вона формує тактові сигнали, необхідні для виконання командних циклів центрального процесора, роботи таймерів, а також обміну інформацією з ВКМ

Схема управління режимами роботи мікроконтроллера:

1. Задає режими роботи МК з точки зору енергозбереження
2. Здійснює початковий запуск мікроконтроллера

Більшість мікроконтроллерів підтримує 3 основні режими роботи:

* Активний режим (Run mode) – мікроконтроллер виконує програму і використовує всі свої ресурси;
* Режим очікування (Wait mode, Idle mode, Halt mode) – припиняє роботу центральний процесор, але продовжують працювати периферійні модулі, потужність споживання зменшується в 3-4 рази;
* Режим зупину (Stop mode, Sleep mode, Power Down) – в цьому режимі припиняє роботу центральний процесор і більшість периферійних модулів, а також тактовий генератор. Вихід з цього режиму можливий або після скиду або зовнішнього переривання. Споживана потужність зменшується в 1000 разів.

Останні 2 режими називають режимами:

1. пониженого енергоспоживання (LPM – Low Power Mode). МК можуть програмно відключати периферійні модулі, які не використовуються в даний момент.

*Ромчік бомчік, виліз на стовпчік, на сопілці грав, Ваню діставав*

Комплементарний Метал-Оксидний Напівпровідник – КМОН технологія (aka CMOS-MK).

Потужність споживання $P=C\*V\_{cc}^{2}\*F\_{T}+ V\_{cc}⋅I\_{Q}$

C – ємність переключення

$V\_{cc}$ – напруга живлення мікроконтроллера

$F\_{T}$ – тактова частота

$I\_{Q}$ – струм витоку в статичному режимі

Для зменшення потужності найкраще зменшити напругу живлення. Іншим варіантом є зменшення тактової частоти і продуктивності.

1. Початковий запуск



2010-03-30

# AVR-мікроконтроллери

## AVR

1. Розроблялась спільно з розробником-компіляторів з мов програмування високого рівня (IAR Systems Ltd.), тому структура AVR-MK максимально оптимізована для того щоб писати прорграми на мовах високого рівня.
2. Застосування конвеєра команд, завдяки чому команди виконуються за один такт

МК виконані за КМОН технологією і поділяються на 4 сімейства:

1. Tiny AVR – найбільш дешеві і прості МК у 8-ми вивідному корпусі. Мають найменшу серед AVR-МК ємність пам’яті та перефирійних пристроїв.
2. Classic AVR – базова лінія МК з середньою продуктивністю до 16 MIPS, з пам’яттю програм 2-8 Кб.
3. Mega AVR – з підвищеною продуктивністю, мають збільшені обсяги пам’яті програм (до 128 Кб) і пам’яті даних (до 4 Кб)
4. XMega AVR – найбільш продуктивні мікроконтроллери, продуктивність до 32 MIPS. Пам’ять програм до 1 Мб. Мають блоки шифрування AES, DES.

Вивчатимемо AT90523117, належить до Classic.

## Архітектура AVR-мікроконтроллера



Всі AVR мають однакове процесорне ядро, виконане по вдосконаленій RISC архітектурі. Вони є 8-ми бітні, команди мікроконтроллера – 16-ти бітні.

## Характеристика ядра AVR

1. Повністю статична архітектура, мінімальна частота – 0 Гц;
2. АЛП підключений безпосередньо до 32-х РЗП*(регістри)*;
3. Завдяки гарвардській RISC-архітектурі більшість команд виконується за 1 такт – продуктивність 1MIPS за МГц;
4. Максимальна тактова частота до 32 МГц;
5. Багаторівнева система переривань, підтримка вкладених переривань – забезпечує швидку реакцію на події;
6. Наявність програмного стеку.

## Характеристики підсистеми вводу-виводу

1. Програмне конфігурування;
2. Всі виводи мають буфери з тригером Шмідта – це збільшує завадостійкість;
3. Можливість підключення до виводів внутрішніх підтягуючих резисторів опором 35...120 кОм;
4. Навантажувальна здатність всіх виходів становить 20мА – це дозволяє керувати багальма зовнішніми пристроями без додаткових драйверів

## Характеристики AVR-МК AT90S2313 (аналог Attiny2313)

1. 2 Кб флеш-пам’яті програм (кількість перепрограмувань не менше 1000)
2. 128 EEPROM пам’яті даних (кількість циклів перезапису – до 100000)
3. 128 байт ОЗП (SRAM) даних;
4. Тактова частота до 10 МГц(ATtiny2313 до 20 МГц);
5. 15 ліній вводу-виводу загального призначення;
6. Напруга живлення від 1.8 до 5.5В
7. Два режими пониженого енергоспоживання
8. Підтримує зовнішні і внутрішні переривання;
9. Периферійні пристрої
	1. Два таймери/лічильники (T0 I T1) 8 і 16 розрядні;
	2. Універсальній асинхронний приймач-передавач UART;
	3. Сторожовий таймер (перезавантажує автоматом контроллер при підвисанні);
	4. Аналоговий компаратор;



## Організація пам’яті AVR-мікроконтроллерів

### АЛП та файл РЗП

Мають 32 РЗП.



Всі 32 безпосередньо підключені до АЛП. Кожен регістр може бути як джерелом даних, так і приймачем даних. Всі операції в АЛП виконуються за один такт. Він виконує арифметичні, логічні і бітові операції.

Sub R5, R17

Or R18, R20

Sbr R26, 0b00000011

R16-R31 можуть використовуватися в командах з безпосередньою адресацією. Напр., команда ldi (завести константу в регістр) R17, 0x5F спрацює, а якщо R7, то ні, бо діапазон вже не той.

Пам’ять AVR мікроконтроллерів виконана по гарвардській архітектурі(окремо пам’ять програм і пам’ять даних).

Пам’ять даних складається з трьох областей:

1. Регістрова пам’ять (32 РЗП і 64 регістри вводу-виводу);
2. Статисного ОЗП – 128 байт;
3. Пам’яті EEPROM – 128 байт.

### Структура пам’яті AVR-мікроконтроллера



Для адресації пам’яті програм (тобто для звернення до пам’яті програм, так?), чить для адресації пам’яті програм використовується програмний лічильник.1223івфвфів.31337 (пін код).

Після скиду виконання програми починається з нульової адреси.

### Пам’ять даних

**Регістровий файл.** Можна звертатися як до регістрів, а можна як до комірок пам’яті.

Шість з 32 регістрів файлу – R26…R31 можна використовувати як 16-розрядні позначки адреси при непрямій адресації даних. Це так звані індексні регістри, позначаються як X,Y,Z.



**Регістри вводу/виводу**. Містять службові регістри і регістри управління периферійними пристроями.

Юхху, «*ну добре... сьогодні вже не встигнемо...* на сьогодні тоді все».

Лекція після модуля

2010-04-20

*Пісня в голові: And the world around us... just keep feeding me down [або шось таке]... A… a… aa..*

# Тема: EEPROM пам’ять

AVR мікроконтроллери мають таку пам’ять. AVR AT90S2313 має 128 байт EEPROM.

Час запису в пам’ять є від 1.5 – 4 мс, залежить від напруги живлення.

Для доступу до EEPROM пам’яті використовуються 3 регістри вводу/виводу (РВВ).

EEAR*(EEPROM Address Register)* – регістр адреси, задає адресу комірки пам’яті при r/w

EEDR(EEPROM Data Register) – регістр даних, містить байт, який читається або записується в EEPROM пам’ять

EECR(EEPROM Control Register) – регістр управління, керує доступом до пам’яті. Розглянемо його детальніше.



* EERE – 1 дозволяє читати. Читається з EEDR дані. Після читання біт скидається в 0. Операція читання займає 4 такти.
* EEWE – дозволяє запису. Після встановлення в 1 відбувається запис EEPROM пам’яті. Після завершення запису біт апаратно скидається в 0.
* EEMWE(Master Write Enable) – біт управління дозволом запису. Якщо EEWE і EEMWE = 1, то операція запису виконується. Після встановлення в 1 він апаратно скидається в 0 після 4х тактів.

Пр.

R17 – адреса комірки EEPROM

R18 – дані для EEPROM

**EEPROM\_Read**:

Sbic EECR, EEWE

Rjmp EEPROM\_Read

*Вийде з циклу вище якщо EEWE стане рівне 0*

Out EEAR, R17

Sbi EECR, EERE

in R18, EEDR

ret

ldi R17, 0x27

rcall EEPROM\_Read

**EEPROM\_Write**:

Sbic EECR, EEWE

Rjmp EEPROM\_Write

Out EEAR, R17

Out EEDR, R18

*Встановлюємо мастер біт:*

Sbi EECR, EEMWE

Sbi EECR, EEWE

Ret *– вихід з підпрограми*

Якщо хочемо записати байт F5, в адресу 0x27:

**Ldi R17, 0x27**

**Ldi R18, 0xF5**

**Rcall EEPROM\_Write**

# Способи адресації в AVR мікроконтроллерах

AVR підтримують такі способи адресації:

* Неявну – операнд знаходиться в якомусь регістрі спеціального призначення (регістр прапорців)
* Безпосередню – дані знаходяться в самому коді команди, *aka* and**i** R18, **98**
* Пряма регістрова з одним регістром – тут дані розташовуються в одному з РЗП(регістр загального призначення), *aka* clr R5
* Пряма регістрова з двома регістрами, – mov R5, R7
* Пряма регістрова регістру I/O, – in R17, PORTB; out SREG, R20
* Пряма
* Непряма
* Непряма відносна
* Відносна регістрова
* Відносна зі зміщенням
* *також вітайте:*
* Крива
* Напівкрива
* Перекособочена
1. Дужки означають вміст комірки пам’яті [↑](#footnote-ref-1)