

## Лекція 06

### Загальна інформація про мікроконтролери

## Лекція 6

Судаков О.О, Радченко С.П.

«Сучасна мікропроцесорна техніка»

[6.2]

Мікроконтролер (Micro Control Unit, MCU,  $\mu\text{C}$ ,  $\mu\text{C}$ )

- **Мікроконтролер** – аналого-цифровий однокристальний комп'ютер призначений для керування пристроями
- Містить
  - Процесор**
  - Енергозалежну пам'ять
  - Енергонезалежну пам'ять
  - Периферію**
- Периферія
  - Генератори тактових сигналів
  - Таймери (лічильники, вартові тощо)
  - Компаратори
  - Підсилювачі
  - Цифрові порти вводу-виводу (GPIO)
  - АЦП
  - ЦАП
  - Датчики (температури, тиску тощо)
  - Контролери ПДП (DMA)
  - Послідовні шини (UART, SPI, I2C тощо)
  - Контролер індикатора ...

[6.3]

Застосування MCU

- Вбудовані системи (embedded systems)
- Керування пристроями
  - Пральна машина (програма прання)
  - Мікрохвильова пічка (програма приготування)
  - Двигун автомобіля (керування розподілом запалювання, інжекторами)
  - Підвіска і трансмісія автомобіля (автоблокування диференціала, авторозподіл моменту на передню та задню осі ...)
  - Медичні портативні пристрої (вимірювачі тиску, температури, рівня глюкози, серцеві стимулятори ...)
  - Дозиметри
  - Вимірювальні пристрої (мільтиметри, осцилографи)
  - Лабораторне обладнання (спектрометри, терези, ...)
  - Томографи (керування периферією)
  - ...
- Яким чином керує
  - Процесор вмикає різні периферійні пристрої
  - Периферійні пристрої записують сигнали з датчиків
  - Процесор "аналізує" сигнали з датчиків
  - Процесор змінює режим роботи периферійних пристроїв, щоб реагувати на зміну вхідних даних

[6.4]

Типи та характеристики MCU

- **Архітектура** (ядро)
  - Своя система команд
  - Свої особливості керування периферією
- **Набір периферії** (**призначення**)
  - Широкий діапазон (GPIO, ADC, DAC, ...)
- **Діапазон тактових частот**

- Тактового генератора (0...)
- Процесора (0-300 МГц)
- Периферії (0-...)
- Продуктивність (MFlops, MIPS)
- **Розрядність**
  - Процесора (8-16 біт)
  - Периферії (8-32 біт)
- **Напруга живлення**
  - 5,3.3,1.8,1.2,0.9 В
- **Потужність**
  - Вт ( $10^{-7}$  -  $10^0$ ) для потужних Вт/МГц (до  $10^0$ )

[6.5]

Деякі архітектури MCU

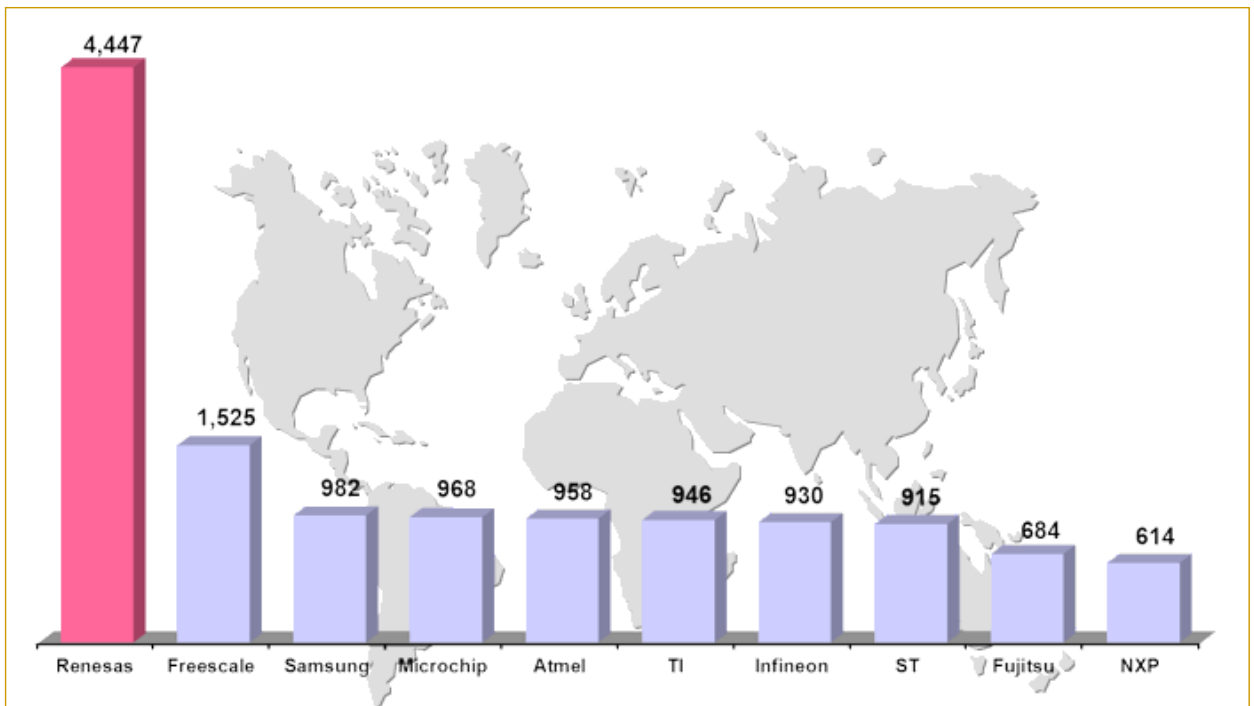
(Як правило стосується процесора)

- **ARM** – (Acorn RISC Machine)
  - Розробник архітектури ARM Holdings
  - Багато виробників процесорів
  - Як правило мікроконтролери з потужним процесором
- **MIPS** (Microprocessor without Interlocked Pipeline Stages)
  - Потужні процесори
  - Декілька виробників
- Intel **8051**
  - Класична архітектура мікроконтролерів, декілька виробників
- Atmel **AVR** (Alf-Egil Bogen and Vegard Wollan RISC)
  - Перші використали Flash пам'ять
- Texas Instruments **MSP430** (Mixed Signal Processor)
  - Наднизька потужність споживання
- **PowerPC** (Performance Optimization With Enhanced RISC – Performance Computing)
- Parallax Propeller (для відеоігор)
- **V850** (Renesas)
- Motorola **6800** і **68000** (M68k)

[6.6]

Виробники мікроконтролерів

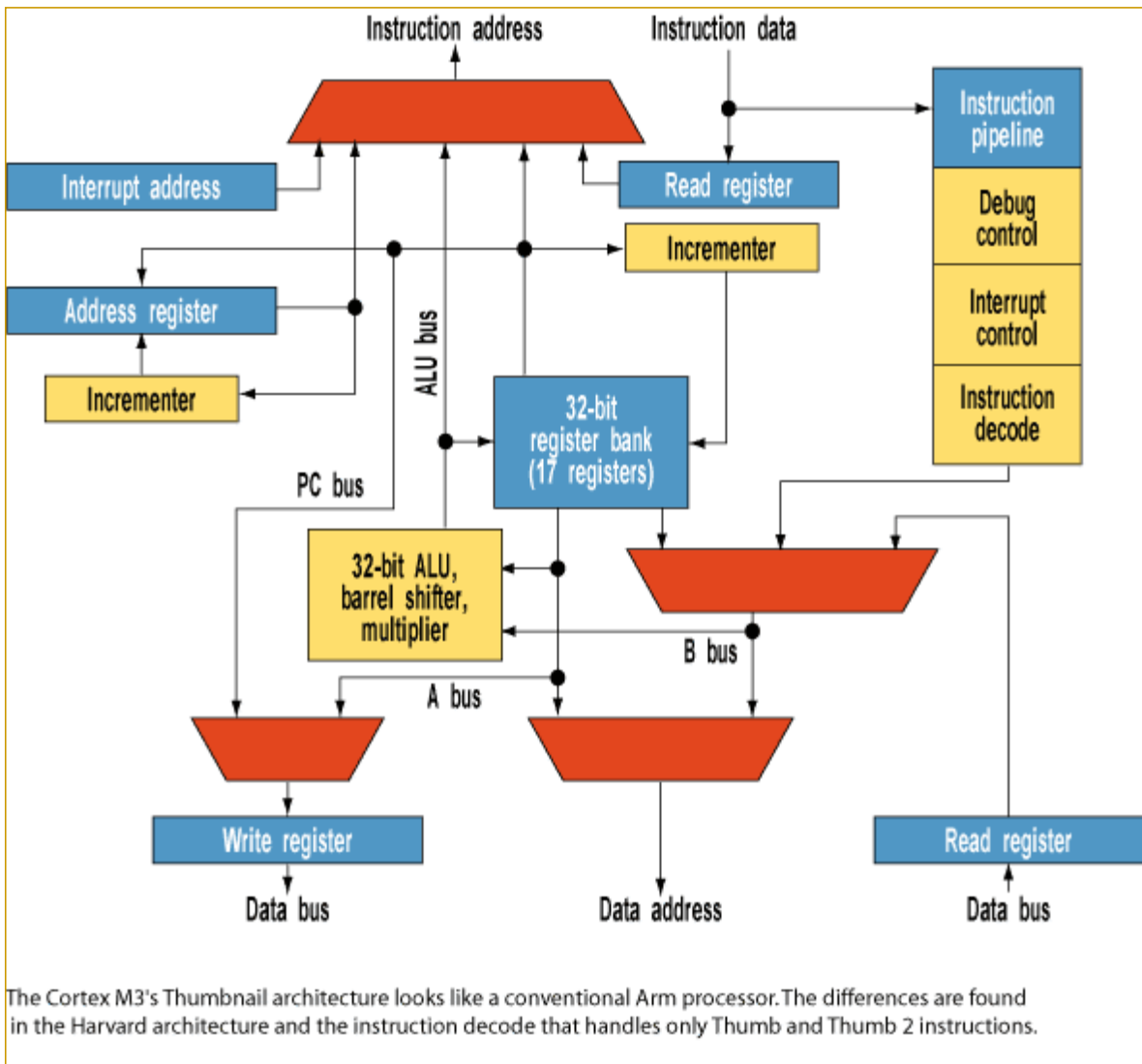
- Renesas
  - 32bit : RX, V850, SuperH, R32C, H8SX
  - 16bit: RL78, H8, M16C, R8C, 78K0R
  - 8bit : 78K0
- Freescale
  - 8,16,32 біт Motorola 6800 і 68000
- Samsung
- Microchip
  - PIC 8,12,13,16,24,32 біт
- Atmel
  - AVR 8 і 32 біт
- Texas Instruments
  - ARM TMS320 C2000 32 біт
  - MSP430 16 біт



[6.7]

ARM-високопродуктивні мікроконтролери для широкого кола задач <http://infocenter.arm.com>

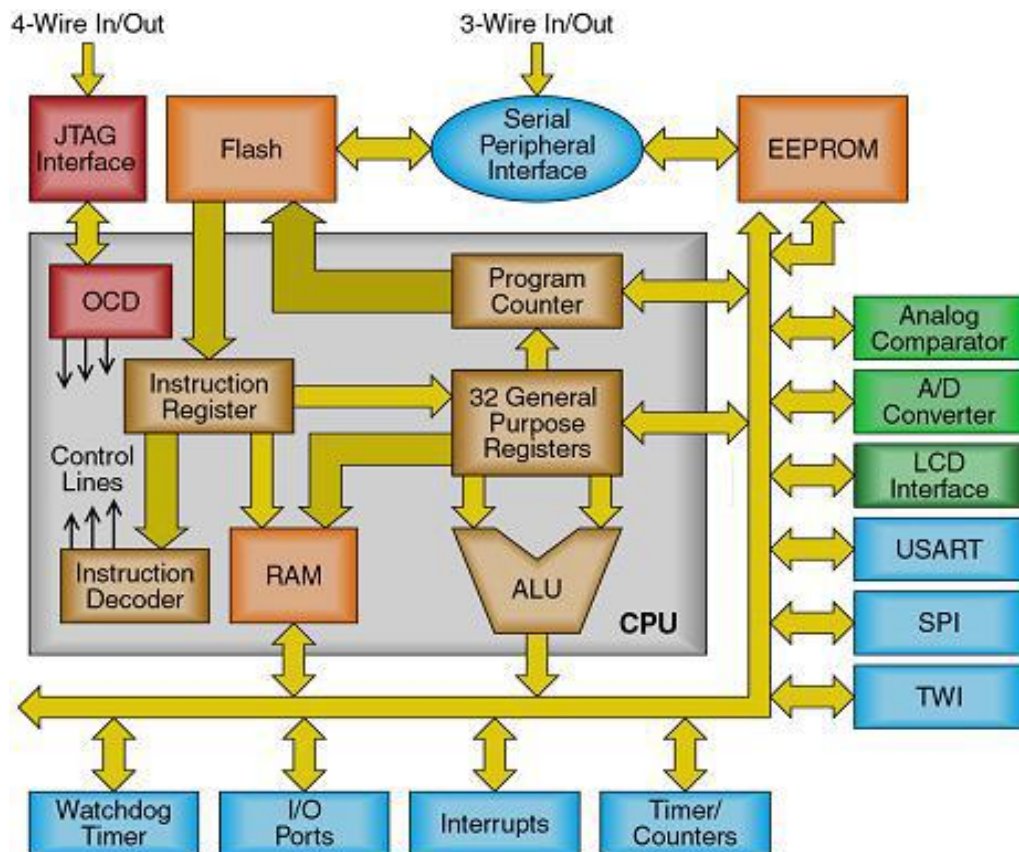
- Найбільше виробників
- RISC 32 біти
- 17 регістрів загального призначення
- Гарвардська архітектура
- Окремий контролер переривань



[6.8]

AVR-мікроконтролери для широкого кола задач <http://atmel.com>

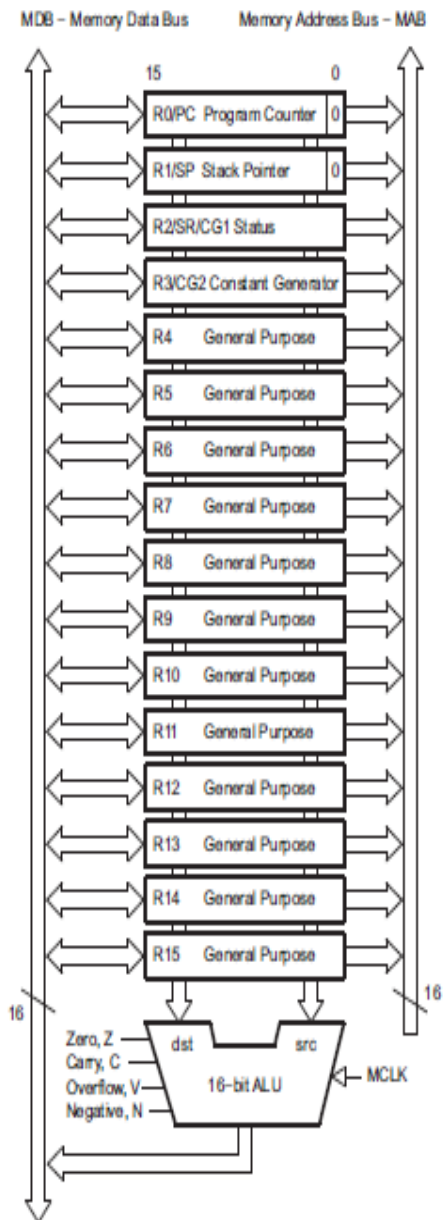
- Виробник ATMEL
  - ATmega
  - ATtiny
  - ..
- RISC процесор
  - 8 або 32 біти
- 16-32 регістра загального призначення
- Гарвардська архітектура



[6.9]

MSP430-мікроконтролери з наднизьким енергоспоживанням

- Виробник Texas Instruments
  - <http://ti.com>
- RISC процесор 16 біт
- 16 регістрів 16 біт
- Фоннейманівська архітектура
- Для мобільних пристроїв
  - Серцеві стимулятори
  - Дозиметри
  - Вимірювачі глюкози



[6.10]

### Основні компоненти мікроконтролерів

- **Генератори**
  - Генерація тактових сигналів різного призначення
- **Процесор**
  - RISC – прості одноступінні команди
  - Оптимізація для обробки переривань
  - Мінімальні затримки
  - Мала продуктивність
- **Пам'ять**
  - Оперативна (Static RAM)
  - Постійна (ROM)
  - Перепрограмована (Flash, EEPROM)
- **GPIO** - цифровий ввід-вивід
  - – виведення високого, або низького рівня на «ніжки» мікросхеми
  - Введення високого, або низького рівня з «ніжок» мікросхеми
- **Таймери**
  - Вимірювання часових інтервалів
- Керування **живленням**
  - Реакція на втрату живлення
- **Аналогові пристрої**
  - АЦП, компаратори, підчилювачі

[6.11]

#### Процесор мікроконтролера-вимоги

- Виконання **простих програм**
  - Основна функціональність лежить на периферії
  - Великого об'єму пам'яті не треба
  - Висока продуктивність не обов'язкова
- **Швидка реакція на події** від периферії
  - Малі затримки в обробці переривань
  - Велика кількість регістрів загального призначення
- **Простий набір команд**, які швидко виконуються
  - RISC – reduced instruction set
  - Команда вимагає мало тактів
- Найчастіше **гарвардська архітектура**
  - Команди і дані в різній пам'яті і передаються по різним шинам
  - Менше тактів на обробку команд з аргументами
  - Недоліки: важко перепрограмувати програму і дані**
- Швидка **зупинка і відновлення** роботи
  - Робота з різною тактовою частотою

[6.12]

#### Режими роботи процесора

##### Програмування периферії

- Запуск периферійних пристроїв на виконання

##### Обробка подій

- Опитування (polling)
- Переривання (interrupting)

##### Опитування (синхронний режим)

- Процесор постійно працює і перевіряє стан пристроїв
- Висока продуктивність
- Простота програмування
- Високі енергозатрати

##### Переривання (асинхронний режим)

- Процесор зупиняється, або виконує опитування
- Коли настає подія робота процесора переривається і викликається функція-обробник події
- Складніше програмування
- Низькі енергозатрати
- Висока швидкість реакції на події
- Нижча продуктивність

```
void setup()
```

```
{  
  pinMode(pin, OUTPUT);  
  attachInterrupt(0, blink..);  
}
```

```
void loop()
```

```
{  
  digitalWrite(pin, state);  
}
```

```
void blink()
```

```
{  
  state = !state;  
}
```

- **Програмування периферії**

- Запуск периферійних пристроїв на виконання

- **Обробка подій**

- Опитування (polling)
- Переривання (interrupting)

- **Опитування (синхронний режим)**

- Процесор постійно працює і перевіряє стан пристроїв
- Висока продуктивність
- Простота програмування
- Високі енергозатрати

- **Переривання (асинхронний режим)**

- Процесор зупиняється, або виконує опитування
- Коли настає подія робота процесора переривається і викликається функція-обробник події
- Складніше програмування
- Низькі енергозатрати
- Висока швидкість реакції на події
- Нижча продуктивність

```
void setup()
{
  pinMode(pin, OUTPUT);
  attachInterrupt(0, blink..);
}

void loop()
{
  digitalWrite(pin, state);
}

void blink()
{
  state = !state;
}
```

[6.13]

Керування периферією

- **Підключення/включення пристроїв**

- В середині мікросхеми
- До зовнішніх виводів
- Є спеціальні регістри 1-ввімкнено 0-вимкнено
- Виконується процесором записом в регістри

- **Програмування пристроїв**

- Вибір та зміна режиму роботи
- Запис векторів переривань
- Запуск пристроїв на виконання
- Опитування і обробка переривань

[6.14]

Приклад – підключення до виводів

- **Кожен вивід мікросхеми може бути підключений до різних виводів різних пристроїв**

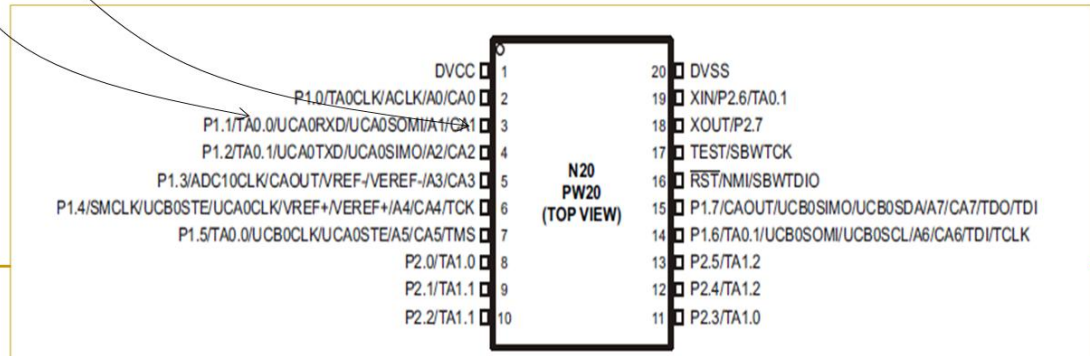
- Таймер
- Порт цифрового вводу-виводу
- Компаратор
- АЦП

- **Повна інформація – в документації на мікроконтролер**

- Datasheet
- User guide



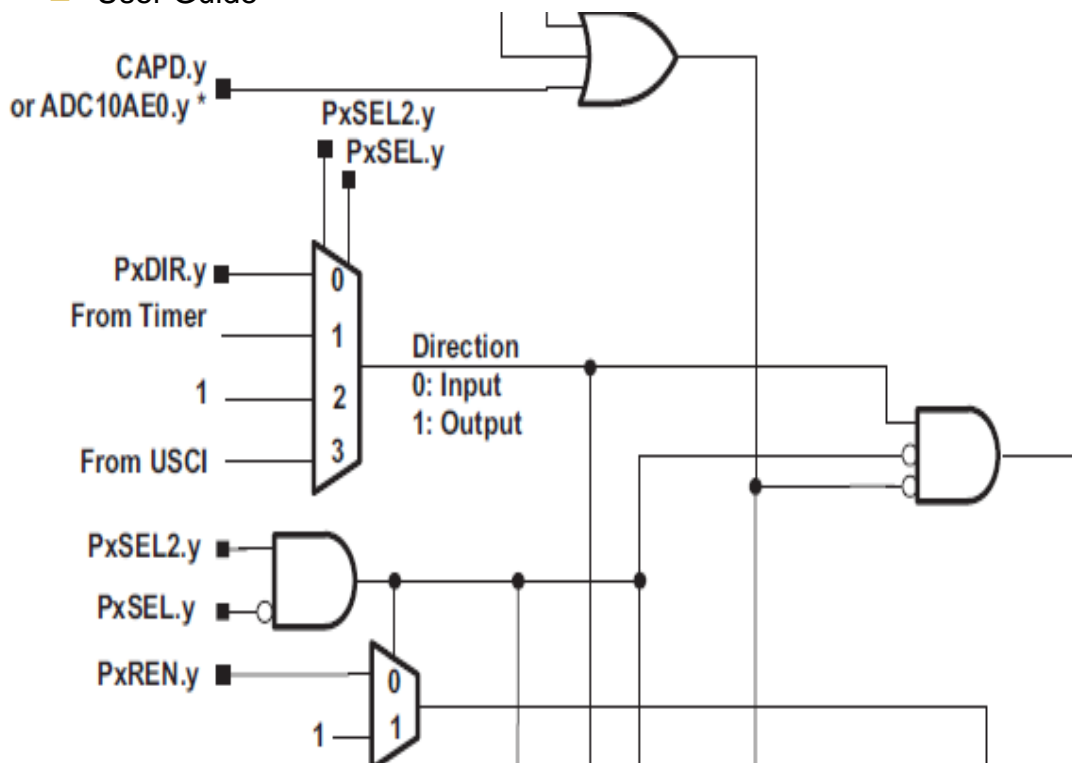
- Кожен вивід мікросхеми може бути підключений до різних виводів різних пристроїв
  - Таймер
  - Порт цифрового вводу-виводу
  - Компаратор
  - АЦП
- Повна інформація – в документації на мікроконтролер
  - Datasheet
  - User guide



[6.15]

Приклад – підключення в середині мікросхеми

- Різні пристрої можуть бути з'єднані між собою
- Повна інформація в документації на мікроконтролер
- Datasheet
- User Guide



[6.16]

Керування підключеннями

- Є спеціальні регістри
  - Часто з відображенням на пам'ять

- Запис бітів у комірку пам'яті змінює стан під'єднання чи від'єднання виводу
- Адреси комірок є в документації
- Ці адреси мають визначені імена
- Приклад
  - Регістр **P1DIR** – ніжка на ввід чи на вивід
    - **P1DIR.0=0** (біт0=0) “ніжка” 1 (**P1.0**) для вводу даних
  - Регістр **P1SEL** - до якого пристрою підключати «ніжку»
    - **P1SEL.0=1** (біт0=1) “ніжка” 1 (**P1.0**) підключена до входу таймера

[6.17]

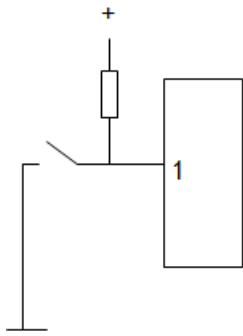
#### Програмування пристроїв

- Кожен пристрій має регістри відображені на пам'ять
  - Запис бітів у комірку пам'яті змінює режим роботи пристрою
  - Читання бітів – одержання результату
- Приклад АЦП
  - REF2\_5V =1** – опорна напруга АЦП 2.5 В
  - SAMPCON =1** запуск АЦП
  - ADC10MEM** – містить результат аналого-цифрового перетворення

[6.18]

#### Опитування пристроїв

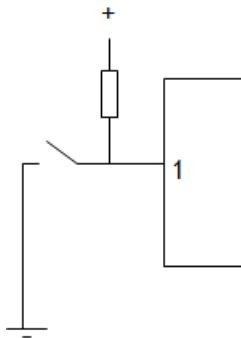
- Періодично перевіряється стан пристрою
  - Ніжки мікроконтролера
  - Регістра
- Приклад
  - Очікування натиснення на кнопку
  - Кнопка приєднана до ніжки 1
  - `while (P1IN & 0x01);`



[6.19]

#### Переривання

- Реєстрація функції –обробника переривання
- Дозвіл переривання
- Очікування переривання
- Приклад – кнопка
  - Реєструємо функцію blink для обробки переривання `attachInterrupt(0, blink..);`
  - Дозволяємо переривання `P1IE=1`
  - Зупиняємо роботу `HLT`



[6.20]

Масковані, немасковані, пріоритетні

- Коли під час виконання обробника переривання приходить інше переривання?
  - Масковані – забороняються одні переривання під час виконання інших
  - Немасковані – не можуть бути заборонені іншими перериваннями (забороняються окремо)
  - Більш високопріоритетні маскують більш низькопріоритетні
- Вихід з ладу генератора – вимагає негайної реакції – немасковане переривання
- Таймер – пріоритетніший ніж кнопка

[6.21]

Тактові генератори (system clock)

- Генерування тактових коливань
- Декілька генераторів для процесора і периферії
  - Високочастотні – для високої продуктивності
  - Низькочастотні – для енергозбереження
  - Високостабільні (кварцеві)
- Керування частотою в широких межах
  - Для високочастотних
- Підключення генераторів до різних пристроїв
- Часто необхідно налаштовувати генератори перед роботою

[6.22]

Контроль живлення

- Що робити при ввімкненні мікроконтролера?
  - Запустити генератор
  - Викликати переривання по перезавантаженню
- Що робити при стрибках живлення
  - Brown Out Reset (BOW) – перезавантажити процесор
- Як економити електроенергію
  - Вимикати непотрібні пристрої
  - Зменшувати частоту тактового генератора під час “сну”

[6.23]

GPIO (General Purpose Input Output)

- Цифрові порти вводу-виводу
  - Запис 0 або 1 на “ніжки” мікроконтролера
  - Зчитування поточного стану “ніжок” мікроконтролера

- Виклик переривання при зміні стану “ніжок” мікроконтролера
- Ввімкнення резисторів “підтягування”
  - Вивід через резистор під’єднується до + або –
- Використовується
  - “Спілкування” із зовнішніми пристроями
  - ШИМ і сигма-дельта ЦАП

[6.24]

#### Таймери

- Таймер – лічильник тактових імпульсів
  - Скільки імпульсів за певний час
  - Скільки імпульсів між певними подіями
  - Запис значення лічильника в регістр при настанні певної події
  - Зміна значення на “ніжці” мікросхеми при досягненні таймером певного значення
- Процесор не може точно відміряти часові інтервали
  - Різні команди мають різний час виконання
  - Наближено можна здійснювати затримку циклом
  - Таймер – незалежний від процесора пристрій контролю часу

[6.25]

#### Режими таймерів

- Режим захоплення (capture)
  - При настанні певної події значення лічильника записується в спеціальний регістр
  - Викликається переривання
  - Для точного слідування за зовнішніми подіями
- Режим порівняння (compare)
  - При досягненні таймером значення в спеціальному регістрі змінюється напруга на виводі таймера
  - Викликається переривання
  - Для точного керування зовнішніми процесами

[6.26]

#### Переривання таймерів

- По переповненню
  - Для врахування ходу часу процесором
- По захопленню
  - Для збереження процесором моменту часу настання події
- По порівнянню
  - Для повідомлення процесору, що запланована дія виконана
- Асинхронний процесор може керувати синхронними подіями

[6.27]

#### Використання таймерів

- Вимірювання тривалості і частоти сигналів
  - Скільки імпульсів тактового генератора проходить за період сигналу
- Генерування імпульсів заданої тривалості і частоти
  - Зміна стану “ніжки” при нарахування певної кількості імпульсів тактового генератора
- Керування
  - Кроковими двигунами
  - ЦАП (ШИМ і сигма-дельта)
  - Схема співпадінь
- Вимірювання

□ Ємності