

## Лекція 01

[1.1]

Сучасна мікропроцесорна техніка

Лекція 1

Судаков О.О, Радченко С.П.

«Сучасна мікропроцесорна техніка»

[1.2]

Актуальність

- Сьогодні мікропроцесори використовуються дуже широко
  - Побутова техніка
  - Транспорт
  - Промисловість
  - Наука
  - Зв'язок
  - Медична техніка
  - ...
- Необхідно розуміти основи, термінологію і вміти використовувати мікропроцесори
- Фахівці з мікропроцесорів - потрібні

[1.3]

Задачі курсу

- **Базові знання про мікропроцесори**
  - Термінологія
  - Класифікація
- **Основи роботи мікропроцесорів**
  - Принципи побудови
  - Характеристики
  - Технології
- **Використання мікропроцесорної техніки**
  - Програмування мікропроцесорів
  - Практичне застосування мікропроцесорів
  - Оцінка характеристик мікропроцесорних систем

[1.4]

Структура курсу

- **Модуль 1 Основи роботи мікропроцесорів.**
  - ЗАГАЛЬНІ ВІДОМОСТІ ПРО МІКРОПРОЦЕСОРИ
  - МІКРОКОНТРОЛЕРИ
  - ЦИФРОВІ МІКРОПРОЦЕСОРИ ЗАГАЛЬНОГО ПРИЗНАЧЕННЯ
  - Контрольна**
- **Модуль 2 Мікропроцесорні системи та технології.**
  - МІКРО ЕОМ
  - СПЕЦІАЛІЗОВАНІ ЦИФРОВІ ПРОЦЕСОРНІ СИСТЕМИ
  - СПЕЦІАЛІЗОВАНІ АНАЛОГОВІ ПРОЦЕСОРИ
  - ФІЗИЧНІ ОСНОВИ ТЕХНОЛОГІЇ МІКРОПРОЦЕСОРІВ
  - Контрольна**
- **Перевірка самостійної роботи**

- Демонстрація роботи пристроїв, виконаних студентами

- Залік

[1.5]

Самостійна робота

- Розробка пристрою на основі мікропроцесора

- Розробка електронної схеми
- Розробка програми
- Моделювання схеми і програми
- Відладка
- Тестування

- TI Launchpad ~30 грн <http://ti.com>

- Arduino 100-500 грн <http://dx.com>

- Програми

- AVRstudio – симулятор AVR
- VMLAB – симулятор AVR
- IAR workbench – симулятор MSP430
- Proteus – симулятор електронних схем

[1.6]

TI Launchpad <https://estore.ti.com/>

- Програматор, платформа для наладки і навчання MSP430

[1.7]

IAR workbench

<http://www.ti.com/lsds/ti/tools-software/devtools.page>

- Розробка для MSP430

[1.8]

Arduino <http://dx.com>

- Програматор і платформа для навчання і наладки AVR

[1.9]

VMLab – <http://www.amctools.com/>

- Симулятор AVR

[1.10]

AVR Studio <http://www.atmel.com/tools/ATMELSTUDIO.aspx>

- Розробка програм для AVR

[1.11]

Proteus

- Симуляція і наладка електронних схем

[1.12]

Документація на процесори

- User Guide – інструкція з використання

- <http://www.atmel.com/tools/ATMELSTUDIO.aspx?tab=documents>
- <http://www.ti.com/lsds/ti/microcontroller/home.page>

- Datasheet – технічні дані

- <http://www.ti.com/lsds/ti/microcontroller/home.page>
- <http://www.atmel.com/tools/ATMELSTUDIO.aspx?tab=documents>

[1.13]

Визначення мікропроцесора

- **Мікропроцесор** – процесор виконаний на інтегральних мікросхемах
- **Процесор** - щось для керування і обробки (пристрій, програма)
  - Центральний процесор комп'ютера
  - Текстовий процесор (програма)
  - Сигнальний процесор
- **Вузьке значення** – програмно-керований пристрій управління та обробки інформації
- **Походження терміну**
  - Процес – послідовність подій у часі
  - Процедура – поступ, просування
- **Не є процесорами** – пристрої де немає поступовості
  - логічні схеми, елементи пам'яті, радіоприймачі

[1.14]

Приклад процесорних систем

- Музична скринька
- Електронний дзвінок
- Персональний комп'ютер
- Векторно-конвейерний суперкомп'ютер
- Автосигналізація

[1.15]

Переваги мікропроцесорів

- Універсальність
- Гнучкість
- Малі розміри
- Вартість
- Мале енергоспоживання
- Простота розробки пристроїв
- Надійність
- Можна використовувати для нетрадиційних потреб

[1.16]

Класифікація мікропроцесорів

**За призначенням**

- Цифрові – для обробки даних у цифровій формі
- Аналогові – для обробки даних у аналоговій формі
- Універсальні (загального призначення)
- Спеціалізовані (ASIC)
- Мікроконтролери (MCU) - цифро-аналогові мікропроцесорні системи для керування пристроями
- Математичні
- Сигнальні (DSP, ASP)

- Для вбудовуваних систем (EMBEDDED)
- Для серверів
- Графічні
- Акселератори
- Комутаційні

[1.17]

Класифікація мікропроцесорів

**За місцем в системі**

- Центральний
- Периферійний
- Співпроцесор
- Контролер (пам'яті, переривань, мережевого адаптера...)

[1.18]

Класифікація цифрових мікропроцесорів

- За архітектурою
  - Паралельна
  - Векторна
  - Конвеєрна
  - Скалярна
  - Суперскалярна
  - Гарвардська
  - Фон-Нейманівська
  - З довгим словом
- За розрядністю
  - 4,8,16,32,64,128
- За системою команд
  - CISC, RISC

[1.19]

Класифікація за фізичними особливостями

- Тактова частота
- Технологія
- Енергоспоживання
- Фізичний принцип

[1.20]

Історична інформація

- З давнього світу до нашого часу
  - Водяні, механічні інші годинники з виконавчими механізмами
  - Автоматизовані астрономічні прилади

[1.21]

Історична інформація

- З 17 сторіччя
  - Механічні обчислювальні системи (Лейбніц, Паскаль, Бебідж)

- Точні механічні годинники

[1.22]

Історичні відомості

- З 19 сторіччя
  - Аналогові електричні релейні та електронні обчислювальні машини
  - Розвиток засобів комунікацій

[1.23]

Історична інформація

- З 1940-х років
  - Релейні і лампові цифрові обчислювальні системи
  - Феритова пам'ять
  - Поява транзисторів

[1.24]

Історична інформація

- З 1950-1960-х р
  - ЕОМ на транзисторах
    - ССРСР – МЭСМ, БЭСМ
    - США – корпорація DEC, CDC, перші комп'ютери IBM
    - Перші багатопроцесорні комп'ютери з віртуальною пам'яттю
  - Поява мікросхем

[1.25]

Історичні відомості

- З 1970-х років
  - Перші мікропроцесори Intel, TI
  - Перші великі інтегральні мікросхеми
  - Векторно-конвейерні системи
  - Масивно-паралельні і багато-комп'ютерні системи
  - Промислові суперкомп'ютери Cray
  - Електронні наручні годинники, мікрокалькулятори

[1.26]

Історичні відомості

- З 1980-х років
  - Персональні комп'ютери
  - Інтернет
  - Кластерні системи
  - Трансп'ютери
  - Вбудовувані системи керування

[1.27]

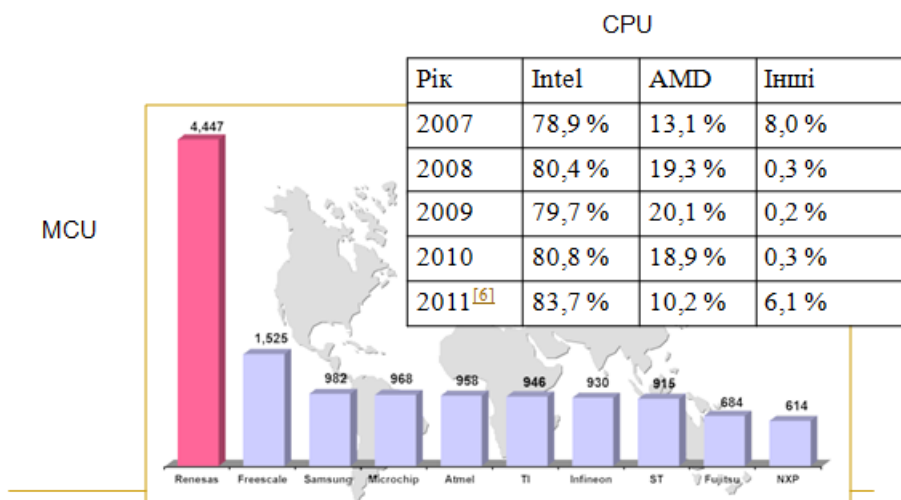
## Історичні відомості

- З 1990-х
  - Цифрові гаджети
  - Тонкі клієнти
  - Віртуалізація
  - Багатоядерність
  - ...

[1.28]

## Виробництво мікропроцесорів

### Виробництво мікропроцесорів



[1.29]

## Перспективи розвитку мікропроцесорів

- Нові ідеї
  - Оптичні
  - Квантові
  - Нейро
- Нові архітектури
  - Паралелізм
- Нові технології
  - Зменшення масштабу
  - Підвищення тактової частоти
  - Нові матеріали

## Лекція 02

[2.1]

### Цифрове представлення інформації

Лекція 2

Судаков О.О, Радченко С.П.

«Сучасна мікропроцесорна техніка»

[2.2]

Інформація

- **Інформація** – будь-які дані про зовнішній світ
  - Є інформація – щось відомо; Невідомо - немає інформації
  - Прийшла інформація – щось взнали; нічого не взнали – не прийшла інформація
  
- **Інформаційні системи** – працюють з інформацію (будь-якими даними)
  - Обробка інформації
  - Збереження інформації
  - Перетворення інформації
  - ..
  
- **Аналогове представлення** – інформація представляється за допомогою деякої фізичної величини, яка приймає неперервні значення
  
- **Цифрове представлення** – інформація представляється за допомогою деякої фізичної величини, яка змінюється дискретно (не неперервного)
  
- **Сигнал** – фізичні процеси (або деякі фізичні величини цих процесів), які переносять інформацію

[2.3]

Аналогове представлення

інформації

- Параметри зовнішнього світу взаємно-однозначно відображаються на значення деяких фізичних величин сигналу
  - Напруга на кінцях провідника відповідає якимось даним (електричний сигнал)
  - Тиск повітря відповідає якимось даним (звуковий сигнал)
  
- **Передача інформації** – переміщення інформації у просторі
  - Нова інформація – непередбачувана зміна сигналу
  - Передбачувана зміна не несе нової інформації
  
- **Модуляція** - зміна якогось параметра сигналу відповідно до зміни інформації
  
- Будь-яка інформаційна система на тому чи іншому етапі використовує аналогове представлення інформації
  - Фізичні величини на практиці здебільшого змінюються неперервним чином

[2.4]

## Дискретне представлення інформації

- Дані представляються у вигляді зліченного набору значень фізичної величини
  - Червоний, жовтий, зелений
  - Висока-низька напруга
- Фізичні величини змінюються неперервно – дискретною може лише інтерпретація діапазонів зміни фізичної величини і часових інтервалів
- **Дискретизація** - представлення неперервного у часі сигналу за допомогою дискретного набору відліків
- **Квантування** – представлення неперервного діапазону значень сигналу у вигляді дискретного набору рівнів
- **Маніпуляція** – модуляція дискретними даними (SK – shift key)

## [2.5]

### Параметри сигналів

- Електрична напруга, струм
  - Високий-низький рівень
- Частота
- Фаза
  - PSK
- Тривалість (ширина) імпульсів
  - PWM
  - Дельта
  - Дельта-сигма
- Амплітуда коливань
  - QAM

## [2.6]

### Датчики

- **Датчик** (джерело даних, сенсор) від англ. data – **дані**
  - Не від слова давати!
- Пристрій, який перетворює інформацію про зовнішній світ у сигнал
  - Датчик Холла – перетворювач магнітне поле-електрична напруга
  - Перетворювач температура-частота
  - Мікрофон
  - Фотодіод
  - Ємнісний датчик
- Основні характеристики
  - Максимальне та мінімальне значення вхідних і вихідних параметрів
  - Динамічний діапазон
  - крутизна перетворення

## [2.7]

### Проблема аналогового представлення

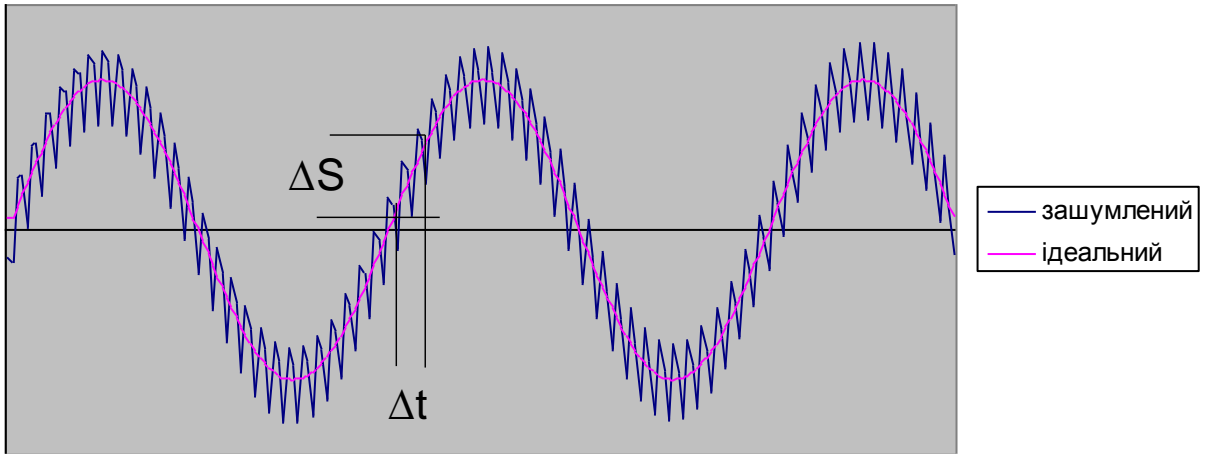
- Неперервний сигнал може нести нескінченну кількість інформації
- Взаємодія сигналу із зовнішнім середовищем призводить до неконтрольованих змін – шумів



- Гарантувано (з високою імовірністю) можна відновити лише попадання сигналу в інтервал значень  $\Delta S$  в інтервалі часу  $\Delta t$  – еквівалентно дискретному сигналу

$$N_q \approx \frac{S_{\max} - S_{\min}}{|\Delta S|} \approx \frac{\text{Signal}}{\text{Noise}} \quad \Delta t \approx \frac{|\Delta S|}{|dS/dt|} \approx \frac{\text{Noise}}{|dS/dt|}$$

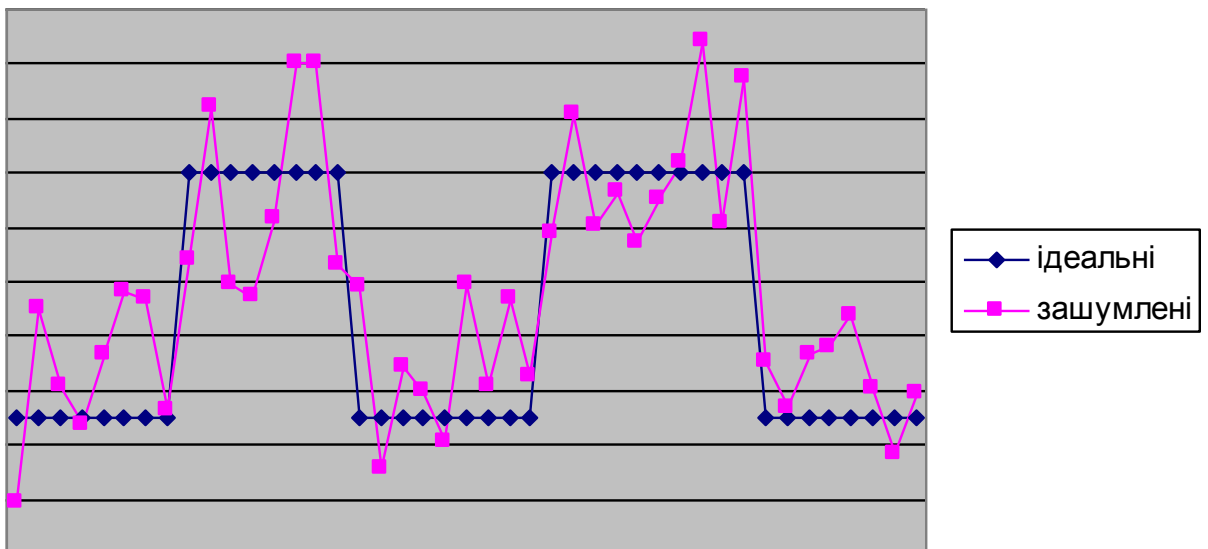
**Максимальний шум - Мінімальна кількість рівнів квантування 2**



[2.8]

Переваги цифрового представлення

- Повне відновлення сигналу (після передачі, запису, обробки)
- Простота і точність виконання перетворення, обробки
- Можна використовувати низькоякісні схеми, канали передачі
- Недолік – передається менше інформації, ніж по аналоговому каналу, необхідність перетворення аналог-цифра-аналог



[2.9]

Дискретне представлення

- Є максимальне і мінімальне значення сигналу
- Неперервний сигнал – нескінченний набір значень

- Дискретний сигнал - скінченний набір значень (рівнів квантування  $Nq$ ) у дискретні моменти часу  $i\Delta t$
- Для представлення дискретного сигналу достатньо цілих чисел від 0 до  $Nq-1$
- Для запису цілих чисел застосовують системи числення

[2.10]

Позиційні системи числення

- Цілі числа можна записати за допомогою різних систем числення

$$x = a_n a_{n-1} \dots a_1 a_0 = \sum_{i=0}^n a_i b^i$$

- $n$  – кількість розрядів,  $b > 0$  – основа системи числення  $0 \leq a_i < b$  – цифра  $i$ -го розряду ( $i$ -ї позиції)
- Розряд 0 – молодший (найменш значущий)
- Розряд  $n$  – старший (найбільш значущий)

$$n = \lfloor \log_b x \rfloor + 1$$

[2.11]

Переведення із однієї системи числення в іншу (1 підхід)

- Є декілька підходів

$$x = a_n a_{n-1} \dots a_1 a_0 = \sum_{i=0}^n a_i b^i$$

- $a_i$  – числа однієї системи числення в іншій
- $b$  – основа однієї системи числення в іншій

Зручно для переведення в десяткову

$$10100011_2 = 1 * 2^0 + 1 * 2^1 + 0 * 2^2 + 0 * 2^3 + 0 * 2^4 + 1 * 2^5 + 0 * 2^6 + 1 * 2^7 = 163_{10}$$

- Кількість розрядів

$$n_{10} = \lfloor \log_{10} 163 \rfloor + 1 = 2 + 1 = 3$$

$$n_2 = \lfloor \log_2 163 \rfloor + 1 = 7 + 1 = 8$$

[2.12]

Переведення із однієї системи числення в іншу (2 підхід)

$$x_0 = x \Rightarrow a_0 = x_0 \bmod b$$

$$x_1 = \text{int}(x_0 / b) \Rightarrow a_1 = x_1 \bmod b$$

$$x_2 = \text{int}(x_1 / b) \Rightarrow a_2 = x_2 \bmod b$$

...

- Зручно для переведення з десяткової

$$163_{10} \rightarrow b = 16$$

$$x_0 = 163 \Rightarrow a_0 = 163 \% 16 = 3$$

$$x_1 = \text{int}(163 / 16) = 10 \Rightarrow a_1 = 10 \% 16 = A$$

$$163_{10} = A3_{16}$$

- Кількість розрядів

$$n_{16} = \lfloor \log_{16} 163 \rfloor + 1 = 1 + 1 = 2$$

[2.13]

Кількість інформації

- Чим вища імовірність повідомлення – тим менше в ньому інформації (Шеннон 1948 р.)
- Імовірності двох незалежних повідомлень перемножуються, а кількість інформації повинна складатись

$$M = \langle \log_b p \rangle = \int \frac{dp(x)}{dx} \log_b p(x) dx = \sum_i p_i \log_b p_i$$

- Імовірність визначається кількістю можливих комбінацій

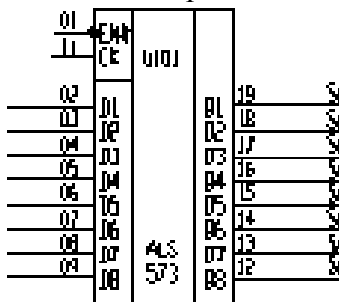
Коли всі рівні квантування рівномірні

$$M = - \sum_{i=0}^{N_q-1} \frac{1}{N_q} \log_b \frac{1}{N_q} = \log_b N_q = n_b$$

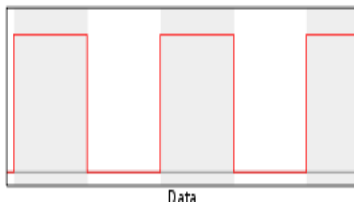
[2.14]

Використання двійкової системи числення

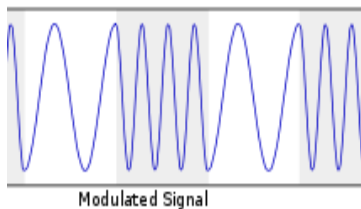
- Основа двійкової системи  $b=2$  відповідає найбільшій завадостійкості – мінімальна кількість рівнів квантування



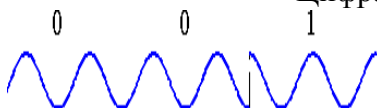
- Кожен розряд – окремий провідник
  - Цифра(біт) – є сигнал, або немає (BC)
  - Цифра(біт) – окрема частота (FSK)
  - Цифра(біт) – протилежні значення фази 0 або 180 PSK

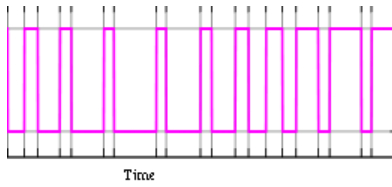


- Кожен розряд – окрема частота (FDM)
  - Цифра є сигнал, або немає
  - Цифра протилежні значення фази 0 або 180 (FSK)



- Кожен розряд – окремий момент часу (TDM)
  - Цифра (біт) є сигнал, або немає (TSK)
  - Цифра – протилежні значення фази 0 або 180 (PSK)
  - Цифра – окрема частота (FSK))
  - Цифра – тривалість імпульсу (PWM)



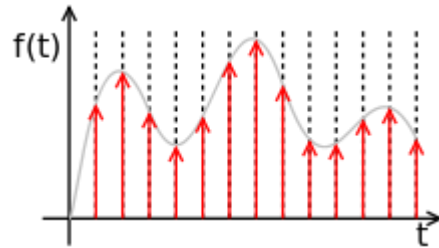


[2.15]

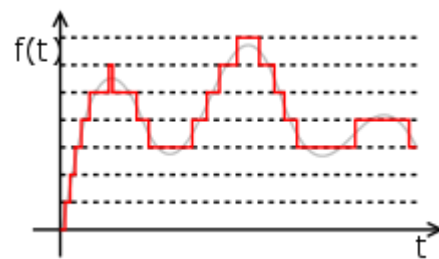
Аналого-цифрове і цифро-аналогове перетворення

■ АЦП (ADC)

□ Дискретизація

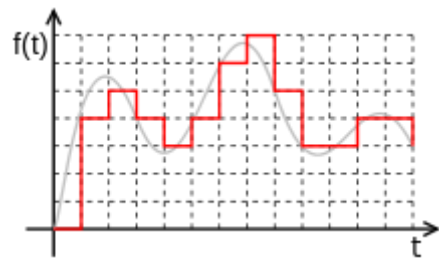
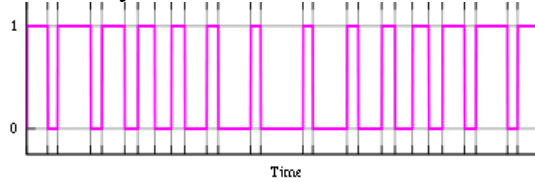


□ Квантування

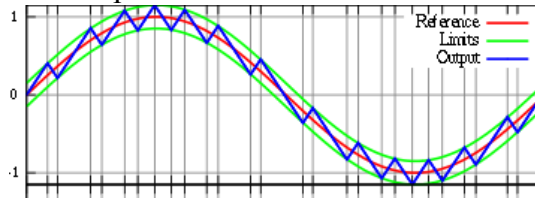


ЦАП (DAC)

□ Модуляція

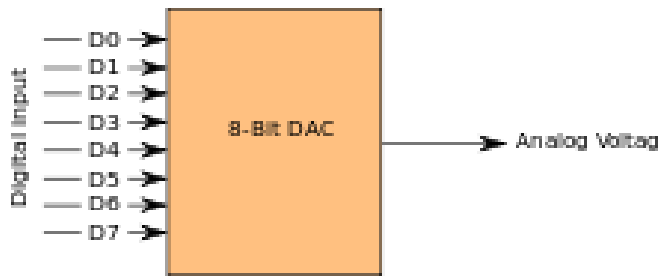


□ Інтерполяція



[2.16]

Цифро-аналогове перетворення



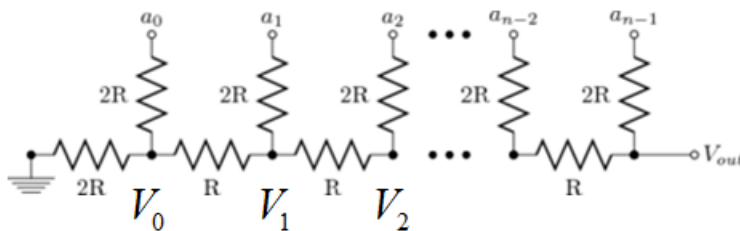
- R-2R матриця (сходинковий)
  - Миттєве перетворення
  - Середня якість
  - Дешева схема
  - Велика кількість резисторів
  
- Широтно-імпульсна модуляція
  - Висока якість
  - Низька частота перетворення
  
- Сигма-дельта (передискретизація)
  - Висока якість
  - Низька частота
  
- Інтерполяція
  - Інтегратор
  - Фільтр нижніх частот

[2.17]

Резистивні матриці

- По теоремі Тевенена
  - Значення 0 – резистори паралельно – опір R
  - Значення 1 – резистори паралельно опір джерела R

$$V_{out} = \frac{V_{ref} * a}{2^N}$$



[2.18]

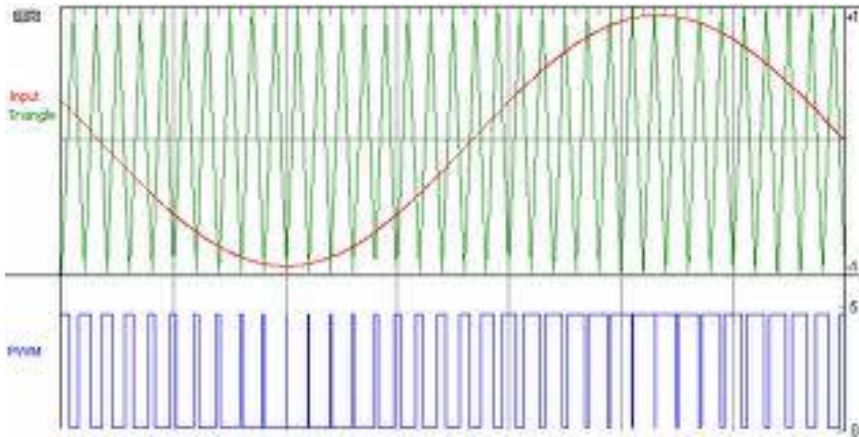
Широтно-імпульсна модуляція

- Змінюється ширина імпульса (скважність)

$$D = t/T$$

- На виході інтегратора встановиться значення напруги  $U_{ref} * D$
- Частота імпульсів значно вища, ніж частота сигналу

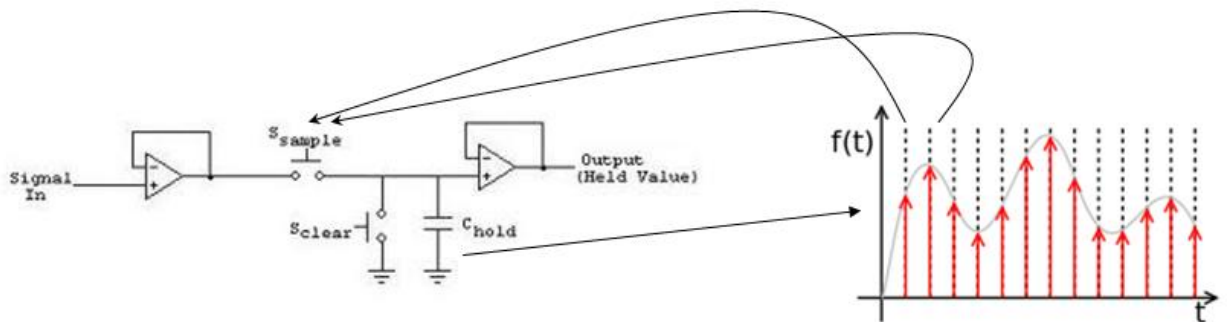
D: 0%



[2.19]

АЦП - Дискретизація (Sample and Hold)

- Період дискретизації  $\Delta t = t_s + t_q$
- Час накопичення (дискретизації)  $t_s = 3-10 \cdot R_{\text{вих}} \cdot C_{\text{hold}}$
- Час перетворення (квантування)  $t_q$

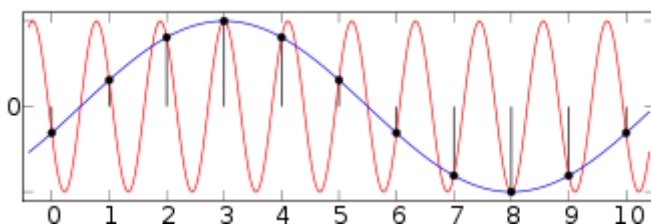


[2.20]

Вибір часу дискретизації

- Теорема Котельникова (Шеннона)

$$\Delta t \leq \frac{1}{2f_{\text{max}}}$$



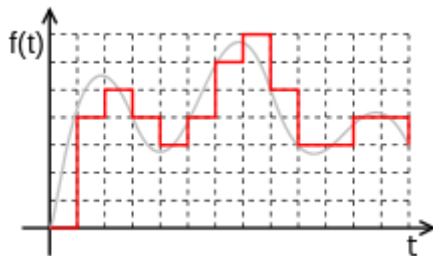
- Aliasing – однаковий результат для різних сигналів коли великий період дискретизації
- Для запобігання
  - великий  $C_{hold}$  малий  $\Delta t$
  - Передискретизація

$$\Delta t \leq \frac{1}{(5-10)f_{max}}$$

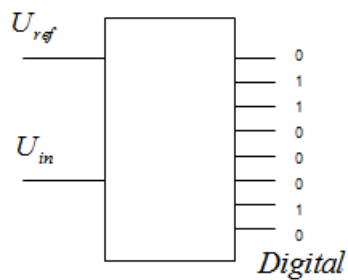
[2.21]

Квантування (перетворення)

- Необхідно задати референтну напругу  $U_{ref}$
- Неперервні (миттєві)
  - Малі часи накопичення  $\rightarrow 0$
  - Малі часи квантування  $\rightarrow 0$
  - Мала кількість рівнів (до 8 біт)
  - Дорога схема
  - Швидкі процеси – відео, осцилографи..  $N_s$



- SAR – successive approximation
  - Необхідність Sample and Hold
  - Тривалі часи перетворення ms-mks
  - Середня кількість рівнів (8-12 біт)
  - Мікроконтролери



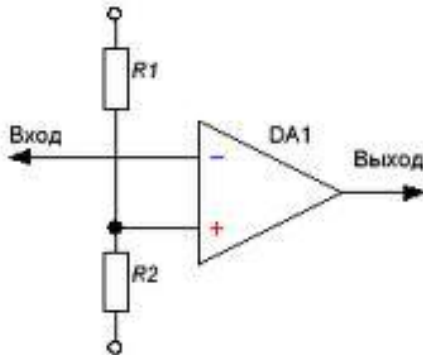
- Дельта-сигма
  - Перетворення напруга-частота
  - Тривалі часи перетворення ms
  - Велика кількість рівнів (8-12 біт)
  - Високоякісна обробка звуку,



[2.22]

## Компаратор

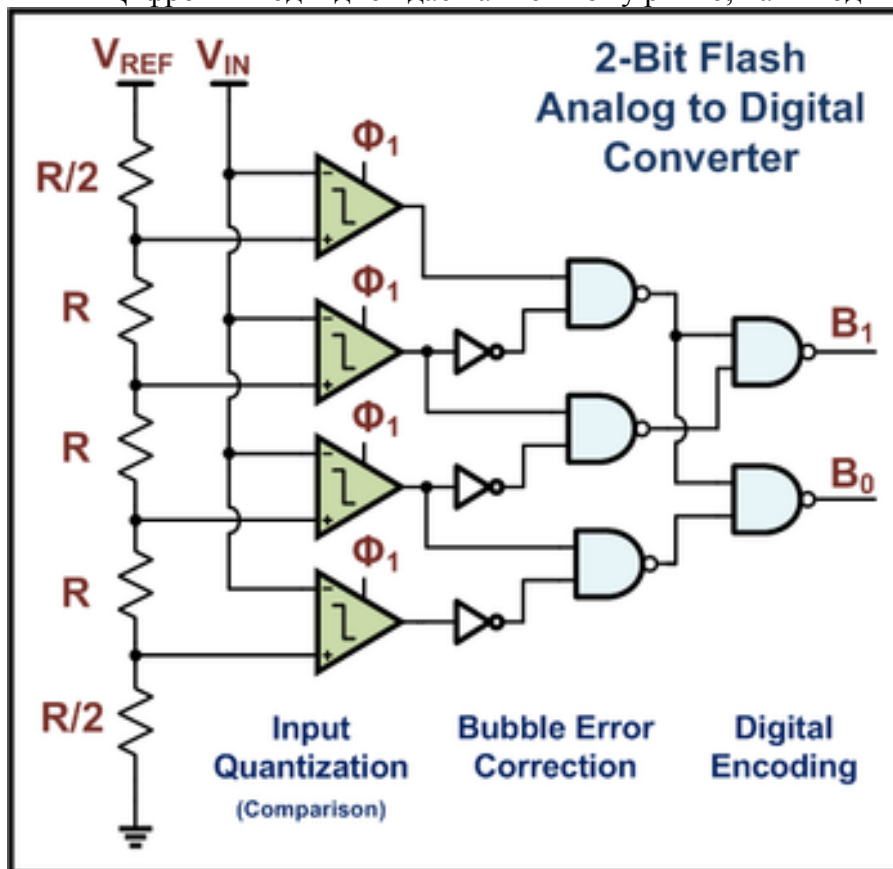
- Однобітний АЦП
- Напряга на вході порівнюється з референтною
  - $U_{вх} < U_{реф}$  – на виході 1
  - $U_{вх} > U_{реф}$  – на виході 0



[2.23]

## Миттєві перетворювачі

- Матриця резисторів ділить референтну напругу на рівні квантування
- Компаратори порівнюють вхідний сигнал із всіма рівнями
- Цифровий код відповідає найменшому рівню, на виході якого компаратор видає 1



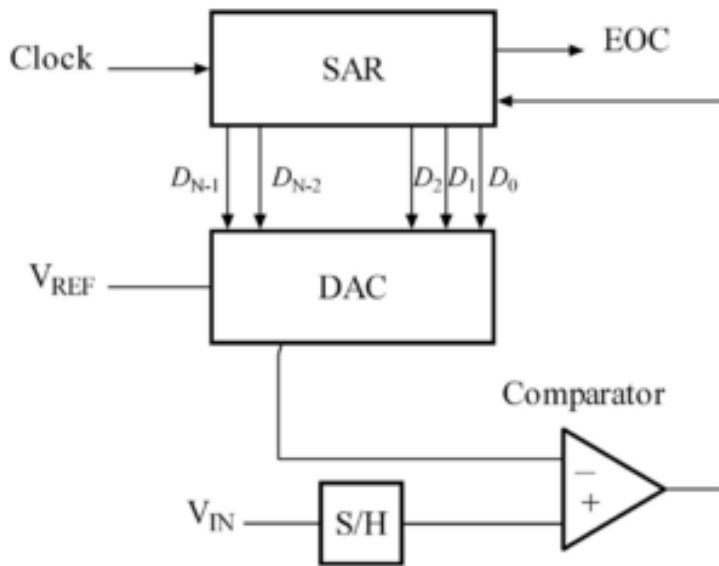
[2.24]

## SAR – successive approximation register

- ЦАП видає сигнал від 0 до  $U_{реф}$  на основі значення в регістрі SAR
- Сигнал на виході ЦАП порівнюється з вхідним компаратором
  - $U_{вх} > U_{сар}$  1



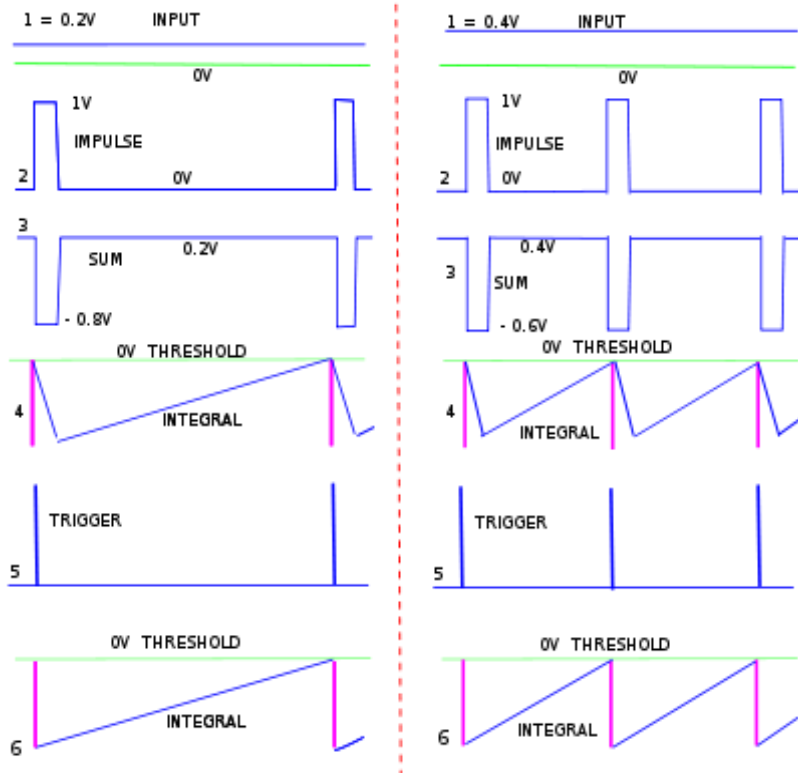
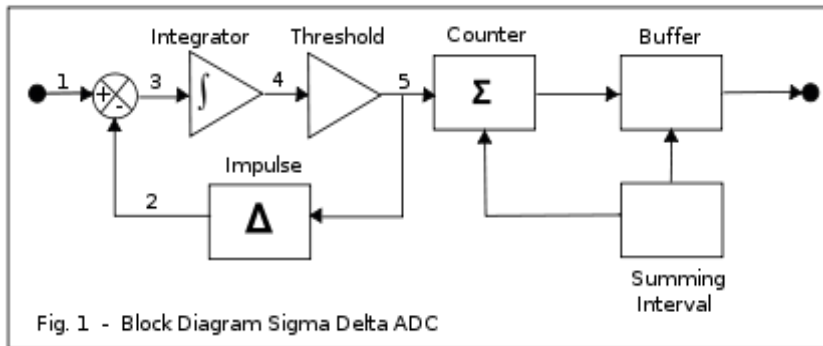
- $U_{вх} < U_{сар} = 0$
- Цей біт записується у відповідну позицію в регістр SAR
- Для  $n$  розрядів треба  $n$  порівнянь і  $n$  тактів генератора



[2.25]

Дельта-сигма

- Вхідний сигнал складається з коротким від'ємним імпульсом  $\Delta$  амплітуди  $U_{реф}$
- На виході інтегратора збільшується напруга поки не спрацює тригер
- Коли спрацьовує тригер генерується новий імпульс  $\Delta$
- Чим вища напруга на вході, тим швидше наростає напруга на виході інтегратора і частіше слідує імпульси
- Лічильник  $\Sigma$  рахує імпульси за певний проміжок часу – ця кількість – код на виході
- За час  $\Delta t$  треба зробити 1 підрахунок
- Частота імпульсів повинна бути дуже великою



[2.26]

Представлення цілих чисел в процесорних системах

- Додатні та від'ємні числа
  - Додатні – старший біт -0
  - Від'ємні старший біт – 1
- Короткі short 16 біт
- Довгі long 32 біт
- long long 64 bit
- Порядок байтів
  - LSB first перший найменш значущий байт
  - MSB first перший найбільш значущий байт

[2.27]

Числа з плаваючою точкою

- Експоненціальне представлення
  - $A \cdot e^B$
  - $Abs(A) < 10$
- Float



## Лекція 03

[3.1]

Основи цифрової схемотехніки

Лекція 3

Судаков О.О, Радченко С.П.

«Сучасна мікропроцесорна техніка»

[2.2]

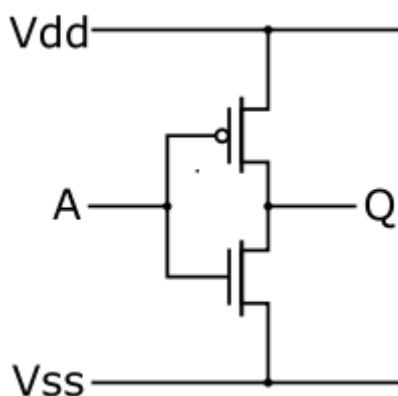
Цифрові інформаційні пристрої

- Обробка інформації у цифровому вигляді
  - Ключі, вимикачі, комутатори
  - Логічні елементи
  - Генератори
  - Тригери
  - Лічильники
  - Перетворювачі частоти
  - Регістри
  - Елементи пам'яті
  - Шифратори-дешифратори
  - Формувачі
  - Математичні операції
  
- Найчастіше використовують двійкове представлення
  - Високий рівень напруги
  - Низький рівень напруги

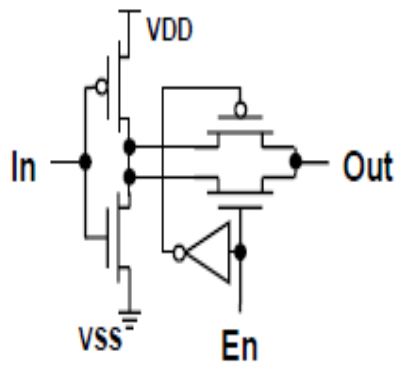
[3.3]

Електронні КМОП (CMOS) ключі (switch)

- Напруга на виході залежить від напруги на вході
- 2 стани
  - Високий рівень
  - Низький рівень



- 3 стани
  - Високий
  - Низький
  - Заборонений (високий опір, плаваючий, Z)

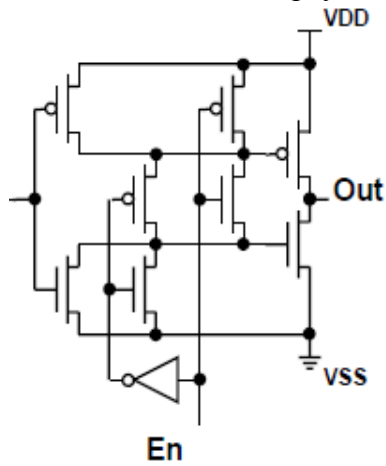


- VDD – вивід стока (drain)
- VSS - вивід витoku (source)
- En - дозвіл

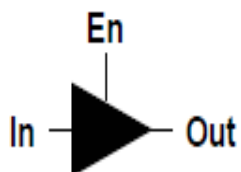
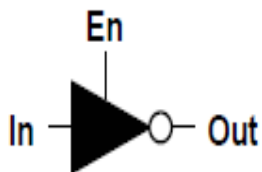
[3.4]

КМОП Буфер (порт, gate)

- Послідовно з'єднані ключі для забезпечення високого входного опору, низького вихідного, високої навантажувальної спроможності
  - Інвертуючі
  - Неінвертуючі

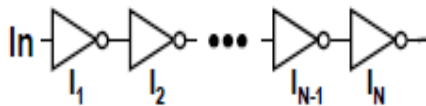


- Підтримка третього стану



- Комутатори
  - Вимикачі

- Перемикачі

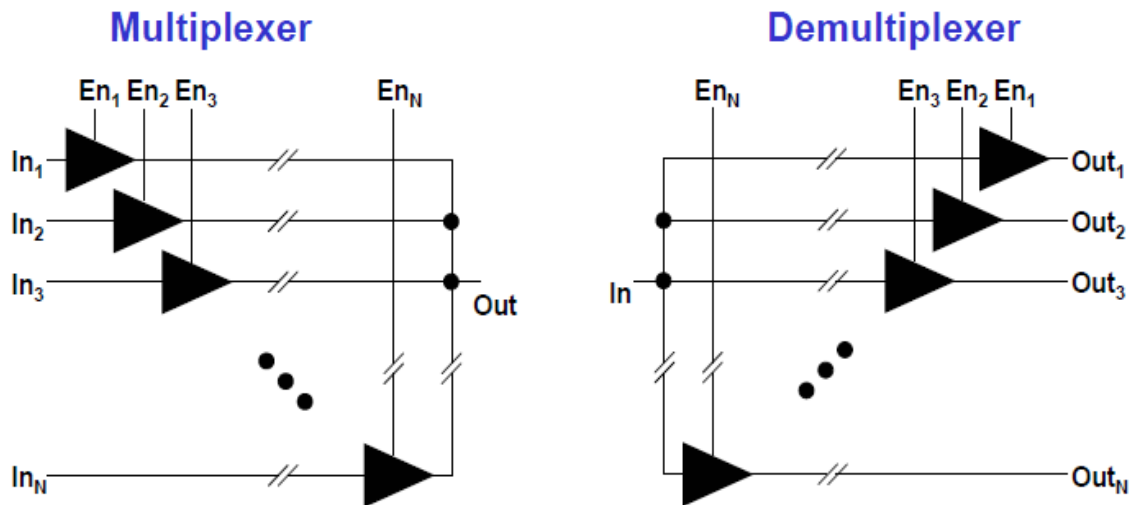


- Виводи мікросхем підключені через порт до зовнішнього світу

[3.5]

Мультимплектори, демультимплектори

- Комутатор з багатьох входів на 1 вихід
- Комутатор з одного входу на багато виходів
- Реалізуються через буфери

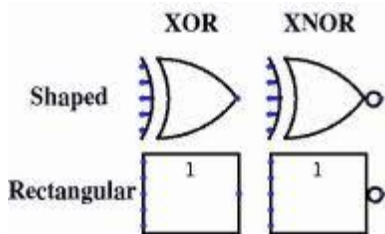


[3.6]

Логічні елементи

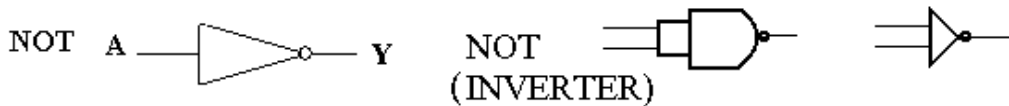
- Виконання логічних операцій
  - НЕ (NOT, INV)
  - АБО (OR, +)
  - І (AND, \*)
  - І-НЕ (NAND)
  - АБО-НЕ (NOR)
  - Виключне або (XOR)
  - Виключне або-не (XNOR)
- Рівні сигналу
  - логічна 1 як правило високий рівень
  - логічний 0 як правило низький рівень
- Основа блок-схем процесорів і мікроконтролерів – треба знати
- Основа роботи багатьох пристроїв

		Binary logic circuit	Used in this document	IEC approved
AND		AND		
NAND		NAND		
OR		OR		
NOR		NOR		
NOT		NOT (INVERTER)		



[3.7]

Логічні операції (NOT)



■ Інверсія біта

NOT 1 = 0

NOT 0 = 1

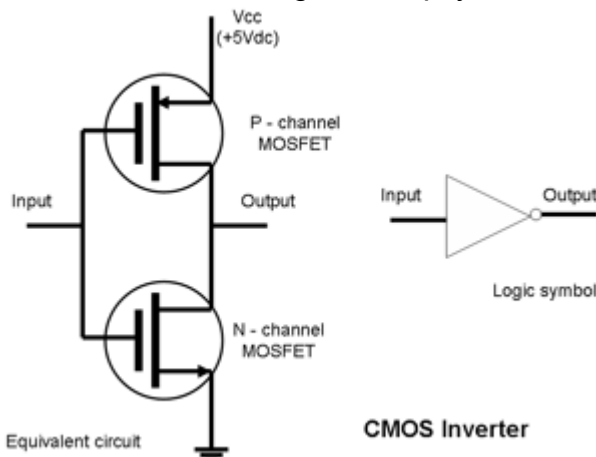
$\bar{1} = 0$

$\bar{0} = 1$

■ Реалізується

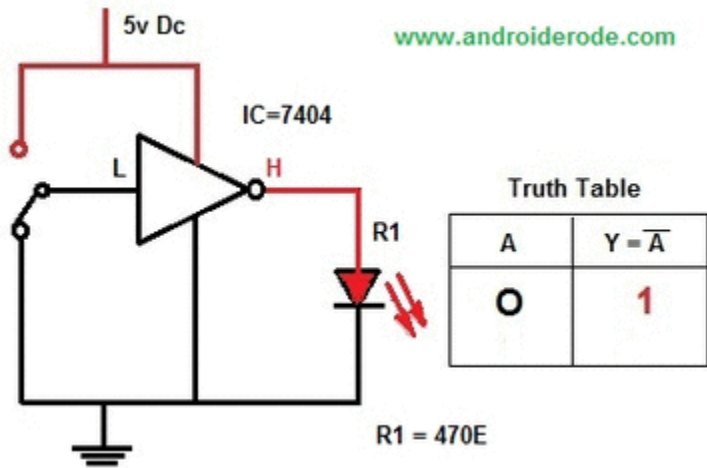
КМОП ключ

КМОП gate інвертуючий



■ Використання

- Зміна рівня на обернений



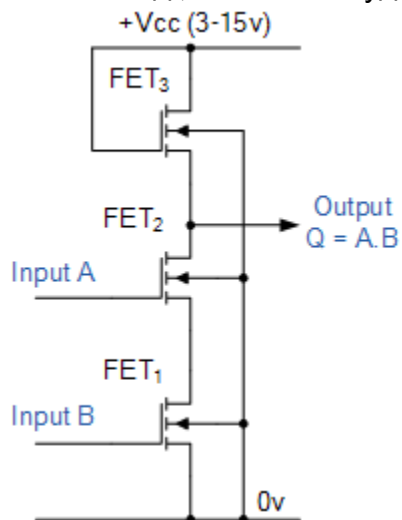
[3.8]

Логічні операції NAND



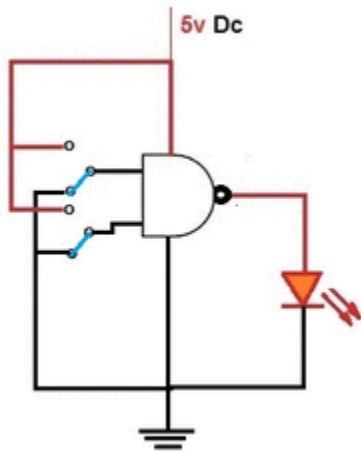
- Логічна операція І-НЕ

- 0 на виході, тільки коли всі входи 1
- 1 виході, коли 0 на будь-якому вході



- Базовий елемент для конструкції інших



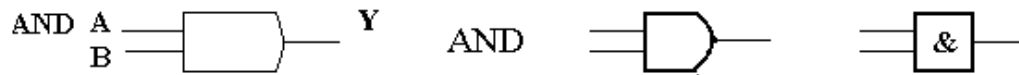


Truth Table

A	B	$Y = \overline{A \cdot B}$
0	0	1

[3.9]

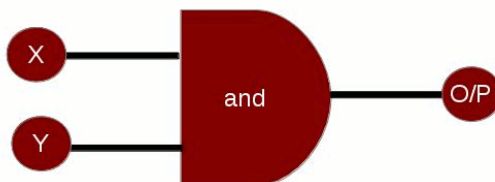
Логічні операції (AND)



- Логічне I
  - 1 на виході, коли на всіх входах 1
  - 0 на виході, коли хоча б 1 вхід 0
- Реалізація
  - NOT NAND
- Використання
  - Керування вимикачами для ввімкнення пристроїв
  - Перемноження сигналів
  - Множення і ділення частоти
  - Видає мінімум всіх вхідних сигналів

and Gate Animation

X	Y	O/P
0	0	0
0	1	0
1	0	0
1	1	1



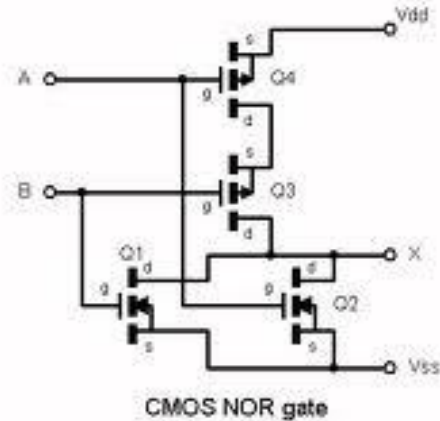
[3.10]

Логічні елементи (NOR)



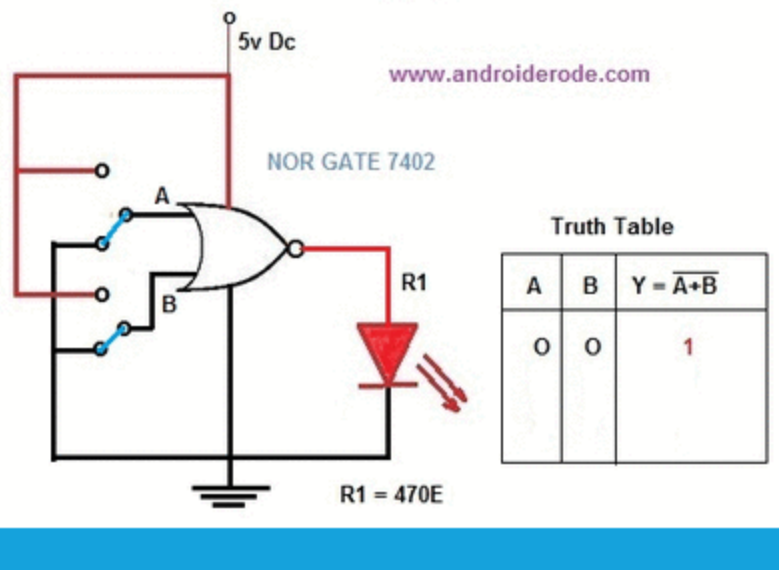
- АБО-НЕ

- На виході 1 лише, коли на всіх входах 0
- Коли хоч на одному вході 1 на виході 0



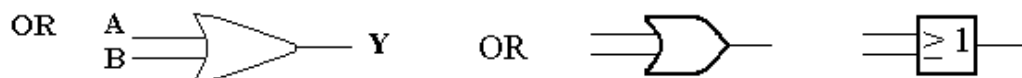
■ Застосування

- Базовий елемент для створення інших



[3.11]

Логічні операції (OR)



■ Логічне додавання

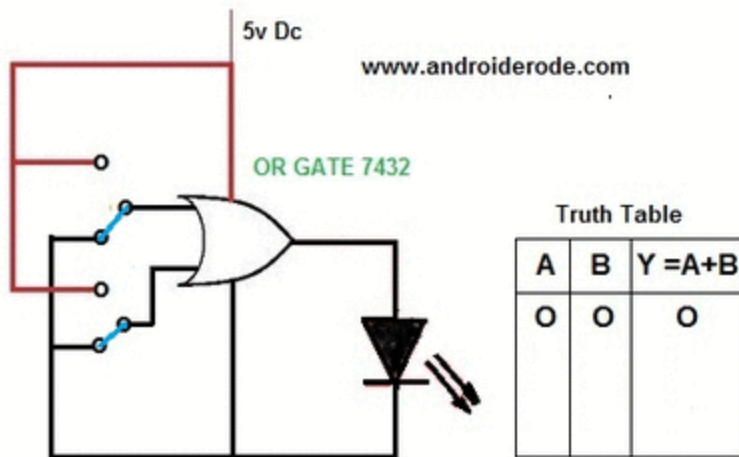
- 0 на виході, тільки коли на всіх входах 0
- 1 на виході – коли хочаб на одному вході 1

■ Реалізація

- NOR NOT

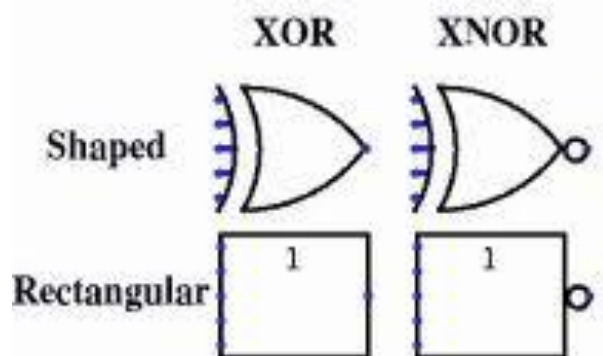
■ Застосування

- Керування вимикачами
- Визначення максимуму
- Додавання сигналів



[3.12]

Логічні операції XOR XNOR



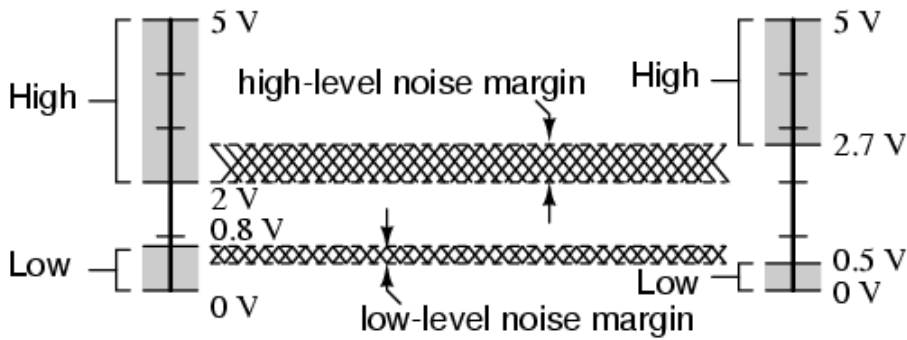
- Виключне АБО
  - 1, коли непарна кількість одиниць на входах
  - 0 коли парна кількість одиниць на входах
- Реалізація
  - $A \text{ XOR } B = (A \text{ AND NOT } B) \text{ OR } (B \text{ AND NOT } A)$
- Використання
  - Додавання чисел у двійковій системі
  - Кодування
  - Обмін між регістрами

[3.13]

Високий і низький рівень – різні значення в різних технологіях

- TTL (транзисторно-транзисторна логіка)
  - Напруга живлення 5В
  - Вихід
    - високий рівень 2.7-5 В
    - Низький рівень 0-0.5 В
  - Вхід
    - високий рівень 2-5 В
    - Низький рівень 0-0.8 В
  - Великі струми споживання – в мікропроцесорах не використовується

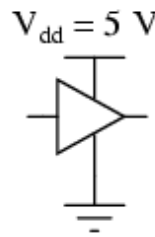
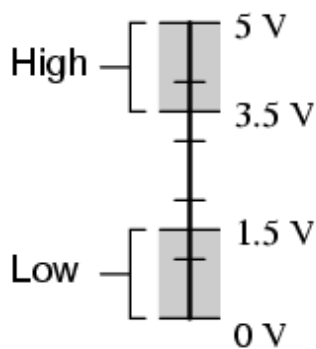
Acceptable TTL gate input signal levels



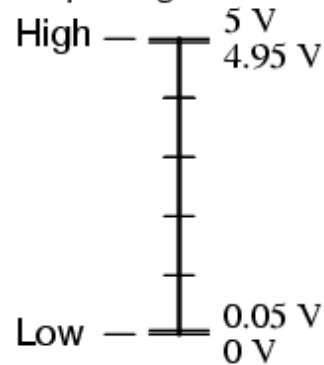
- CMOS (КМОП комплементарні МОП)
  - КМОП 5 В
    - Високий 3.5-5 В
    - Низький 0.05 – 1.5 В
  - КМОП 1.8-3.3 В
    - Високий >1.1-2.5 В
    - Низький <0.35-0.5 В

Acceptable TTL gate output signal levels

Acceptable CMOS gate input signal levels



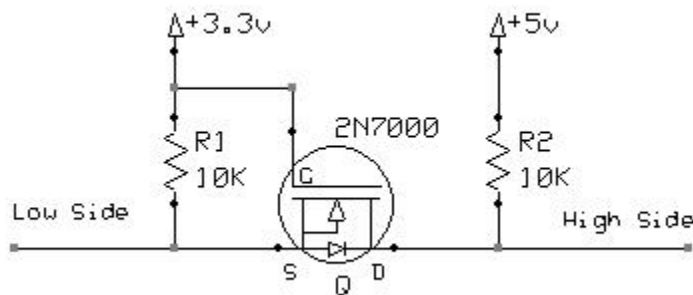
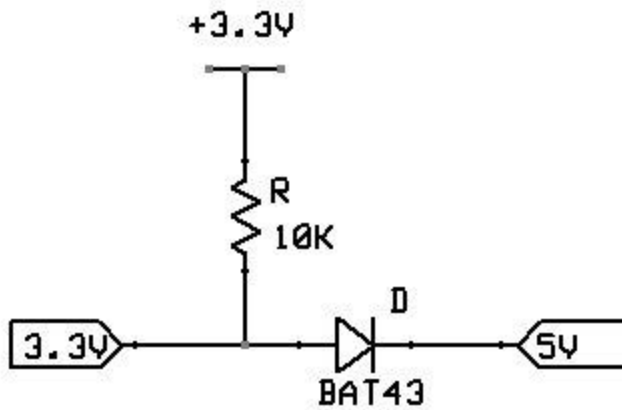
Acceptable CMOS gate output signal levels



[3.14]

Перетворювачі і формувачі рівнів

- Пристрої 5 В можуть не працювати на 3.3
- Пристрої 3.3 можуть згоріти на 5 В
- Дуже не бажано напряму подавати сигнал з мікросхем однієї технології на іншу!
- Треба використовувати перетворення рівнів!
  - TTL-> CMOS
  - 3.3->5 В
  - 1.2->15 В



[3.15]

Елементи із стійкими станами

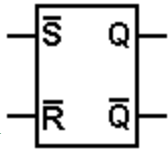
- Логічні елементи
  - Стани не стійкі
  - Не можна зафіксувати стан
- Елементи пам'яті
  - Стійкі стани
  - Вихід з якогось стану можливий шляхом переходу в інший стан
  - Перехід "миттєвий"
- Елементи із стійкими станами
  - Тригери
  - Лічильники
  - Регістри
  - Елементи пам'яті

[3.16]

Тригери (flip-flop, latch)

- Пристрій з двома стабільними станами
  - RS (reset-set, latch)
  - D (delay)
  - T (toggle)
  - JK
- Перекидання між станами миттєве через зворотній зв'язок

[3.17]



RS та JK

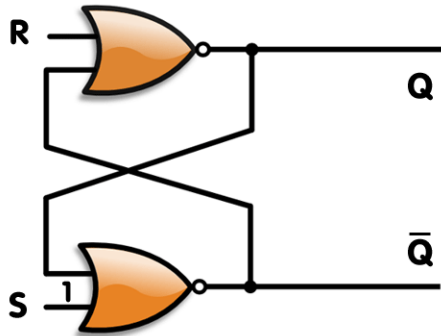
- **RS**-Два входи два інверсні виходи
- Встановлення (set=1)
  - S=1 -> Q=1
  - R=0
- Скидання (reset=1)
  - R=1 -> Q=0
  - S=0
- Збереження (утримання, hold)
  - S=0, R=0, Q=const
- Заборонена комбінація
  - S=1 R=1->Q-невизначене

**JK** – майже те ж саме, що і RS

Відмінність

J=1 K=1 -> перекидання (flip-flop)

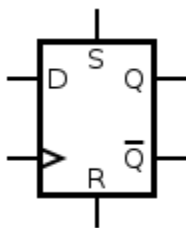
Q=NOT Q



[3.18]

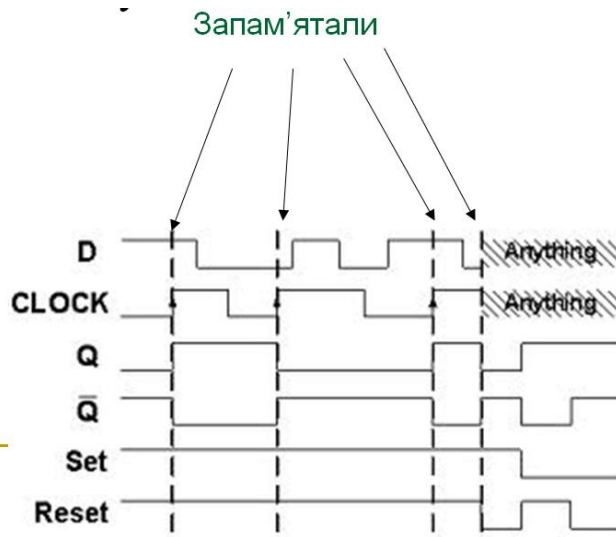
ТЕЖ САМЕ!!!

[3.19]



D тригер (delay, data) flip-flop

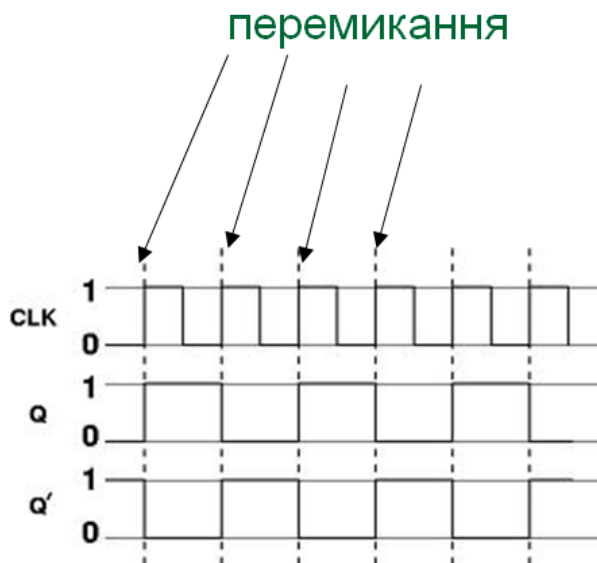
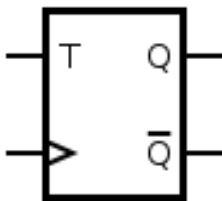
- Передає значення на D вході на Q вихід
- Лише в момент зміни стану входу ">", "Clock"
  - За фронтом
  - За спадом
- Основа
  - Регістрів
  - Комірок пам'яті



[3.20]

### T триггер

- Тригер перемикання (toggle)
- Перемикає свій стан на протилежний
  - По фронту імпульса на вході ">"
  - Або по спаду імпульса на вході ">"
- Вхід T
  - T=1 перемикання дозволене
  - E=0 перемикання недозволене
- Основа
  - Лічильників
  - Подільників частоти



[3.21]

### Тригери з заборорою (gated latch)

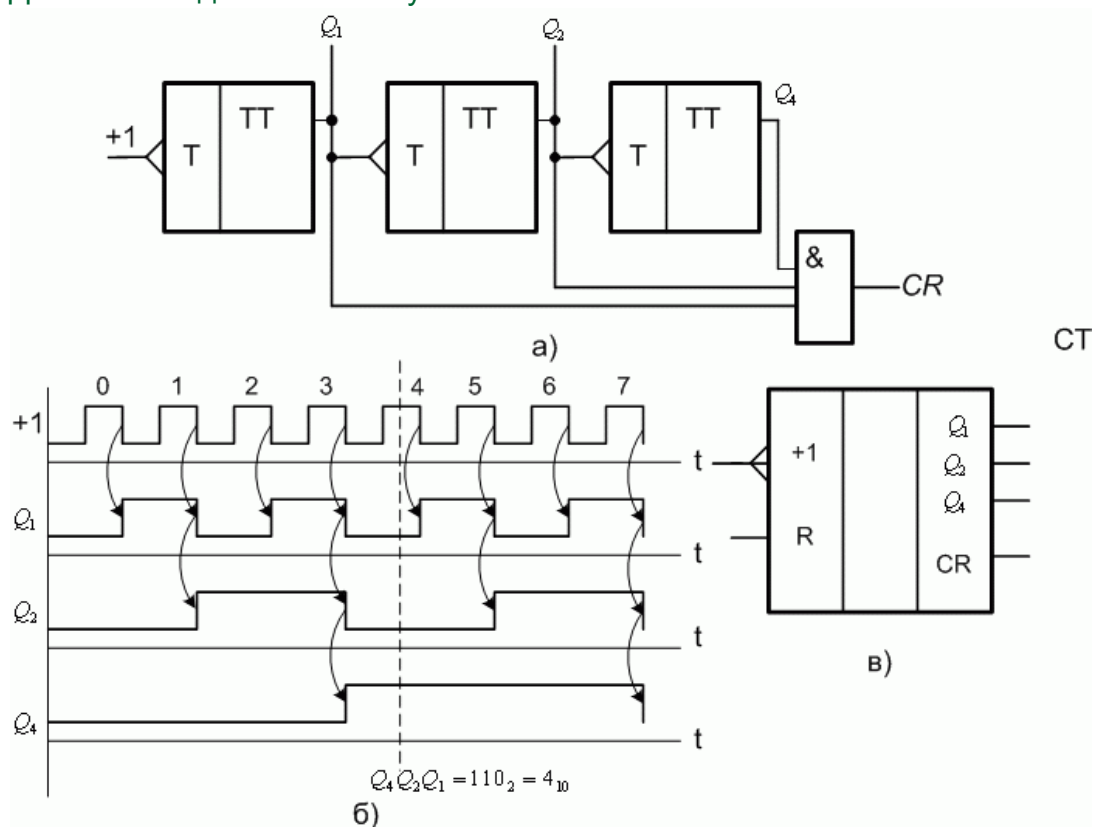
- Synchronous RS, D, JK, T
- Ще один вхід E
  - E=1 пристрій використовується
  - E=0 пристрій зберігає останній стан

[3.22]

### Лічильники, подільники частоти (counter, divider)

- Лічильники- пристрої, на виході яких кількість імпульсів
- Подільник частоти
  - На виході частота в n разів менша за вхідну
- Реалізація – послідовність T- тригерів
- Типи
  - Синхронні
  - Асинхронні

### Двійковий код кількості імпульсів

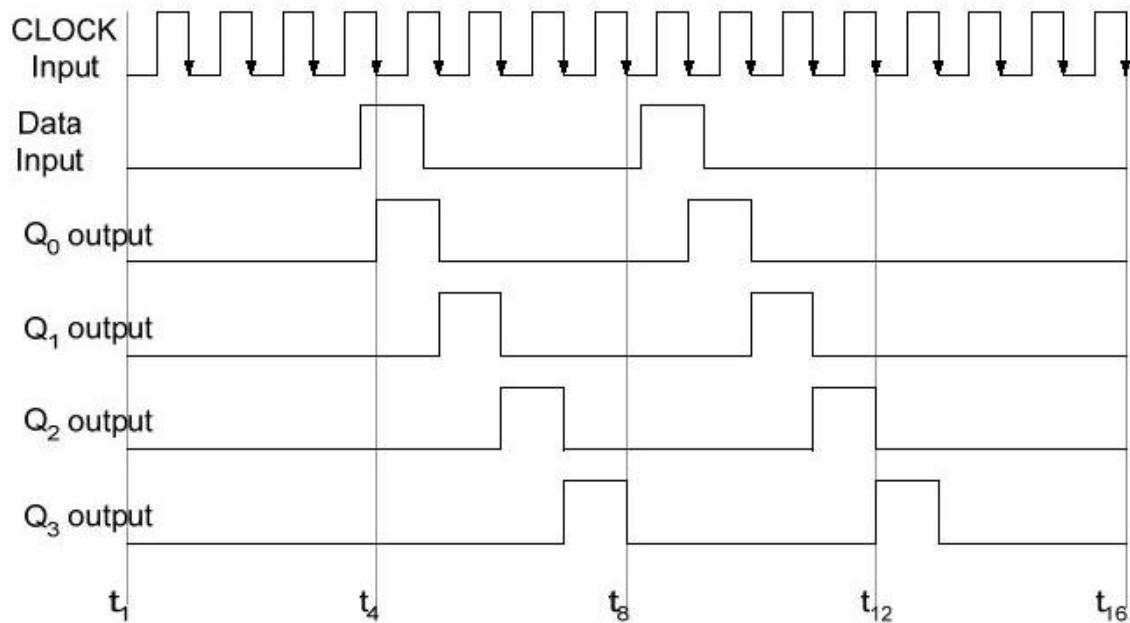
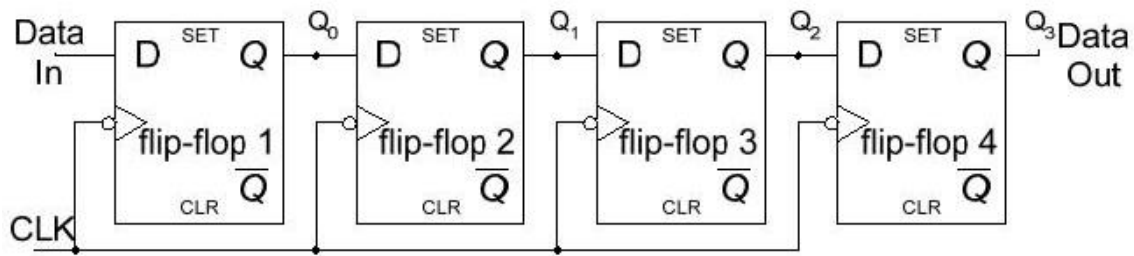


[3.23]

### Регістри

- Запам'ятовуючий пристрій
  - Паралельні
  - Зсуві
- Множення чисел

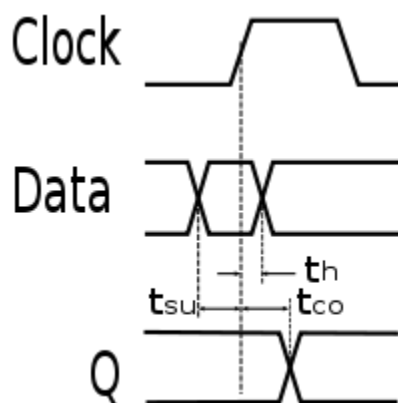




[3.24]

Часові параметри пристроїв із стійкими станами

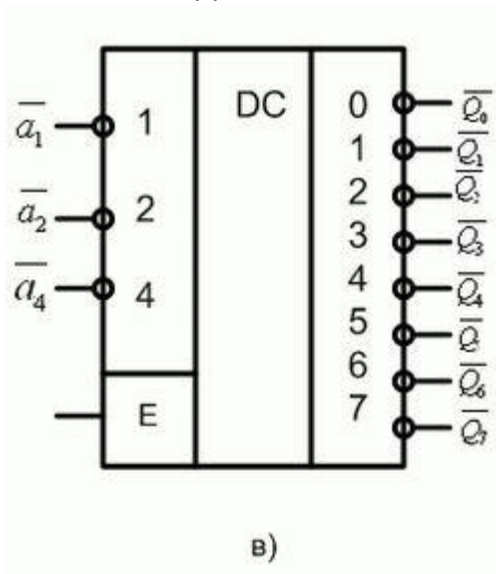
- Перемикання тригерів вимагає часу
- Це визначає швидкість роботи процесорних пристроїв
- Параметри
  - Час встановлення ( $t_{su}$ ) – скільки треба тримати вхідний сигнал, до тактового імпульсу, щоб тригер спрацював
  - Час утримання ( $t_{hi}$ ) – скільки треба тримати вхідний сигнал, до тактового імпульсу, щоб тригер спрацював
  - Час затримки ( $t_{co}$ ) час між встановленням входу і зміною виходу
  - Час наростання фронту



[3.25]

### Шифратори-дешифратори

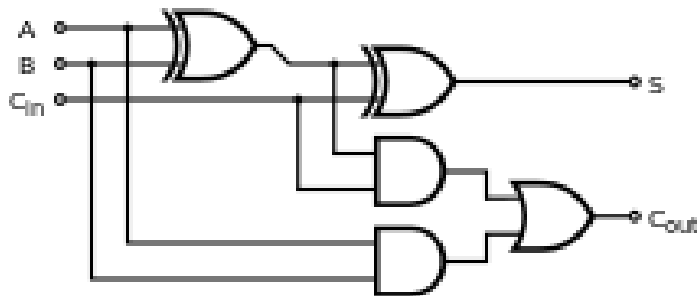
- Пристрої, які одному числу ставлять у відповідність інше за певним законом
  - Двійкове-десятькове
  - Двійкове-семисегментний індикатор



[3.26]

### Арифметичні операції

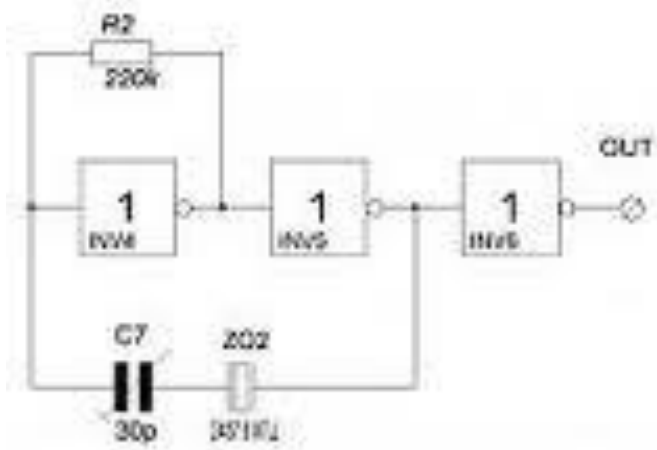
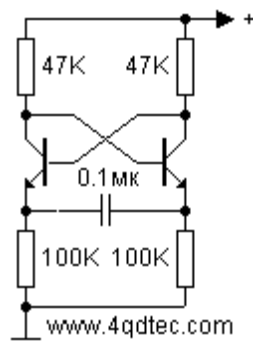
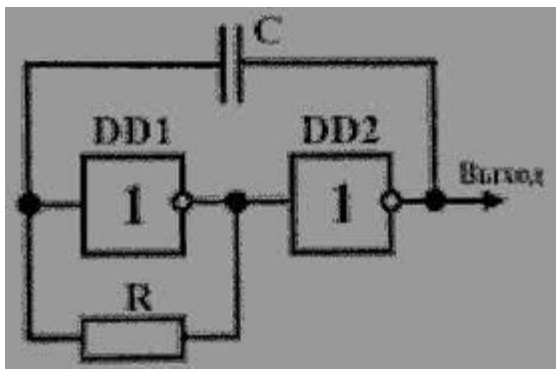
- Додавання
  - Суматор (adder)
  - Входи A B Cin(carry)
  - Виходи S Cout(carry)
- Множення
  - Логічний зсув
  - Бінарне множення



[3.27]

### Генератори коливань

- Системний годинник важлива частина процесорної системи
  - Стабільна частота
  - Можливість перестроювання частоти



## Лекція 04

[4.1]

Складові компоненти мікропроцесорів [2.2]

Цифрові інформаційні пристрої

Лекція 4

Судаков О.О, Радченко С.П.

«Сучасна мікропроцесорна техніка»

[4.2]

Комбінаційні та послідовносні схеми

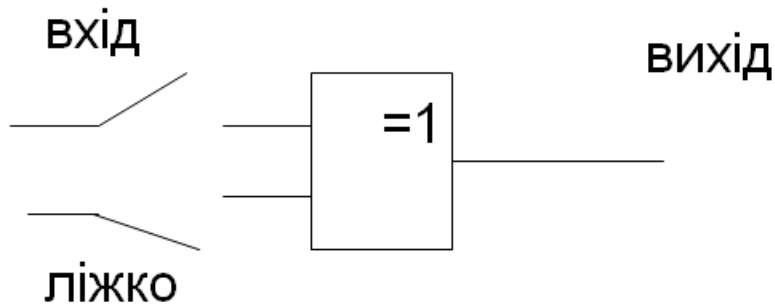
- Комбінаційна –результат на виході залежить від поточного значення параметрів на входах
  - комутатори
  - логічні схеми
  - Генератори
  - Шифратори -дешифратори
- Послідовносна – результат на виході залежить від входів, часу, стану
  - тригери
  - лічильники
  - регістри
  - Пам'ять
  - процесори

[3.3]

Комбінаційна схема (вимикач світла)

- В кімнаті 2 вимикачі
  - 1-й біля входу
  - 2-й біля ліжка
- Зробити схему яка вмикає і вимикає світло в кімнаті
  - Вмикаєш при вході
  - Вимикаєш з ліжка
  - Вмикаєш з ліжка
  - Вимикаєш при вході

Вхід	Ліжка	Вихід
0	0	0
1	0	1
0	1	1
1	1	0

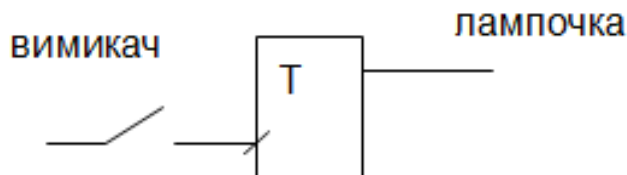


Вихід = (вхід AND NOT ліжко) OR  
 (NOT вхід AND ліжко) = вхід XOR ліжко

[4.4]

Послідовнісна схема

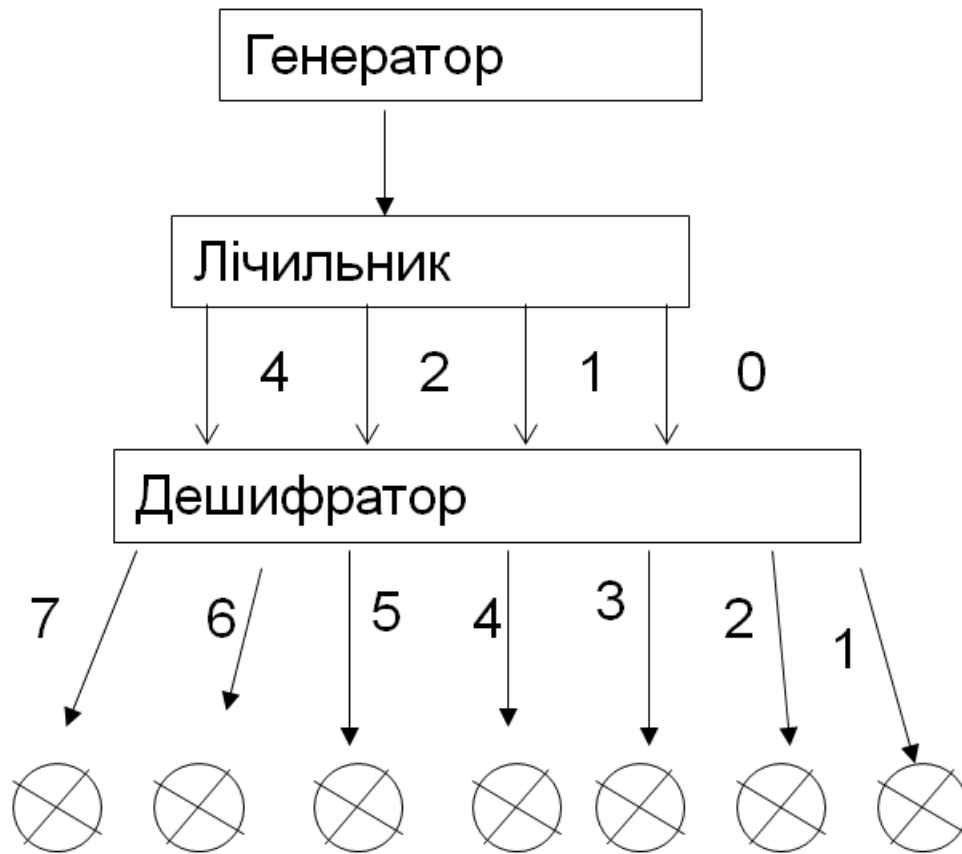
- Один вимикач і одна лампочка
  - Вимикання вимакача нічого не міняє
  - Вмикання вимакача вимикає лампочку, коли вона горить і вмикає – коли не горить



[4.5]

Біжучі вогні

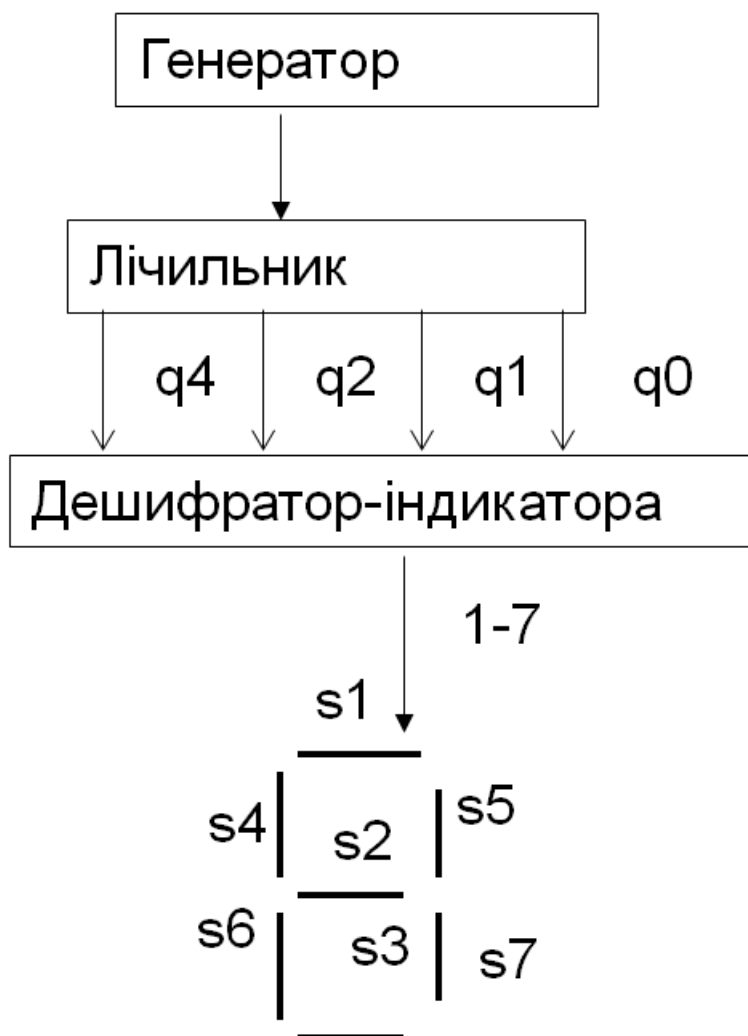
- Система з послідовним виконанням
- Примітивний процесор
  - Лічильник рахує імпульси генератора
  - Видає на виході двійковий код
  - Дешифратор запалює лампу з відповідним номером
- Компоненти
  - Генератор – системний годинник (SC)
  - Лічильник-програмний лічильник (PC)
  - Дешифратор – пам'ять команд
  - Лампочки- виконувальний пристрій
- Вихід лічильника
  - Шина адрес
  - Адреса команди



[4.6]

Дешифратор

- Дешифратор –комбінаційна схема
  - Кожному вхідному коду співставляється вихідний код
  - Аналог програми і пам'яті
- Кожній адресі можна співставити
  - Пристрій
  - Команду
  - Дані
- Постійна пам'ять



0:s1 & s4 & s6 & s3 & s7 & s5

S1= (NOT q0 & NOT q1 & NOT q2 & NOT q4) or ..

S2=

[4.7]

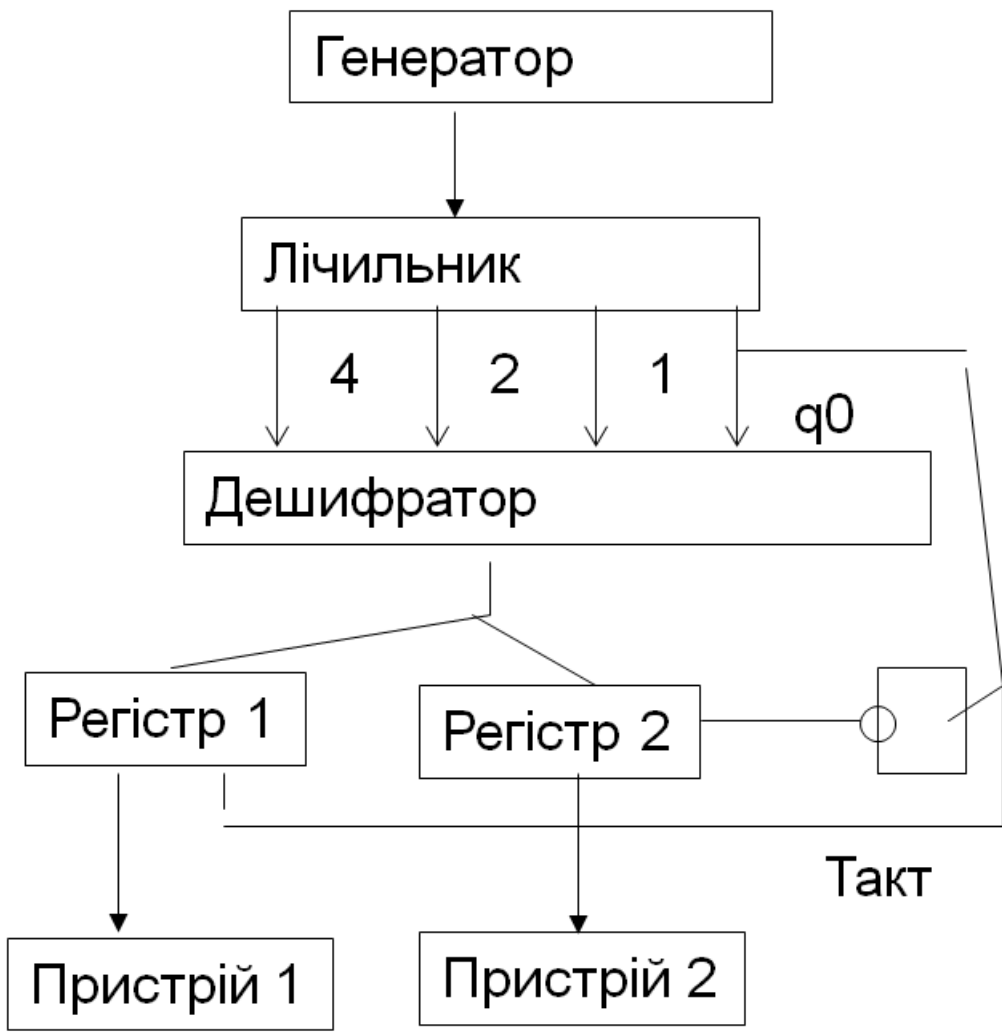
Керування декількома пристроями

■ **Задача**

- Пристрій 1 – кожна парна команда
- Пристрій 2 – кожна непарна команда

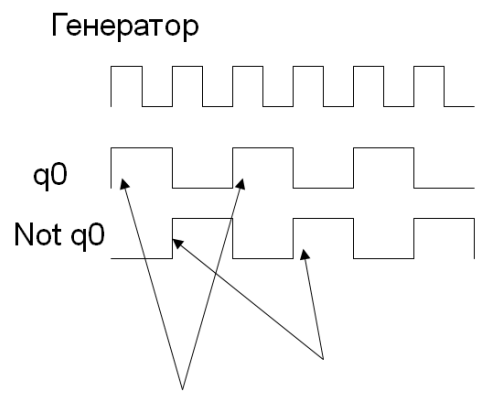
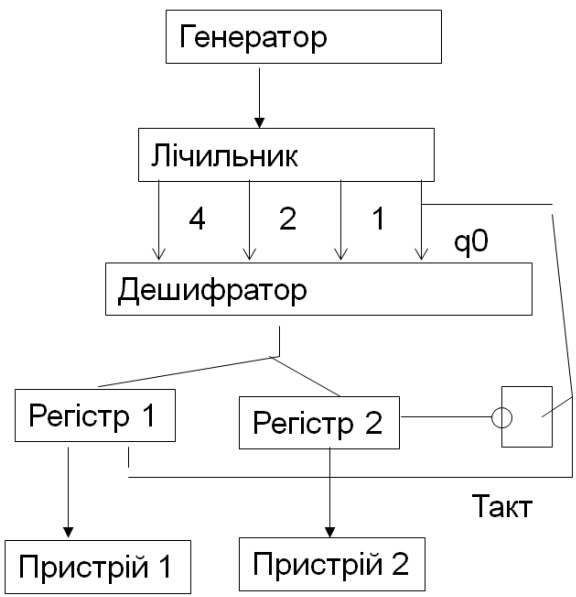
■ **Реалізація**

- Дешифратор видає на вихід команди
- Регістр 1 запам'ятовує непарні команди
- Регістр 2 запам'ятовує парні команди



[4.8]

Робота схеми



Запис в регістр 1      Запис в регістр 2

Операція з пристроєм виконується за 2 такта – вдвічі повільніше



[4.9]

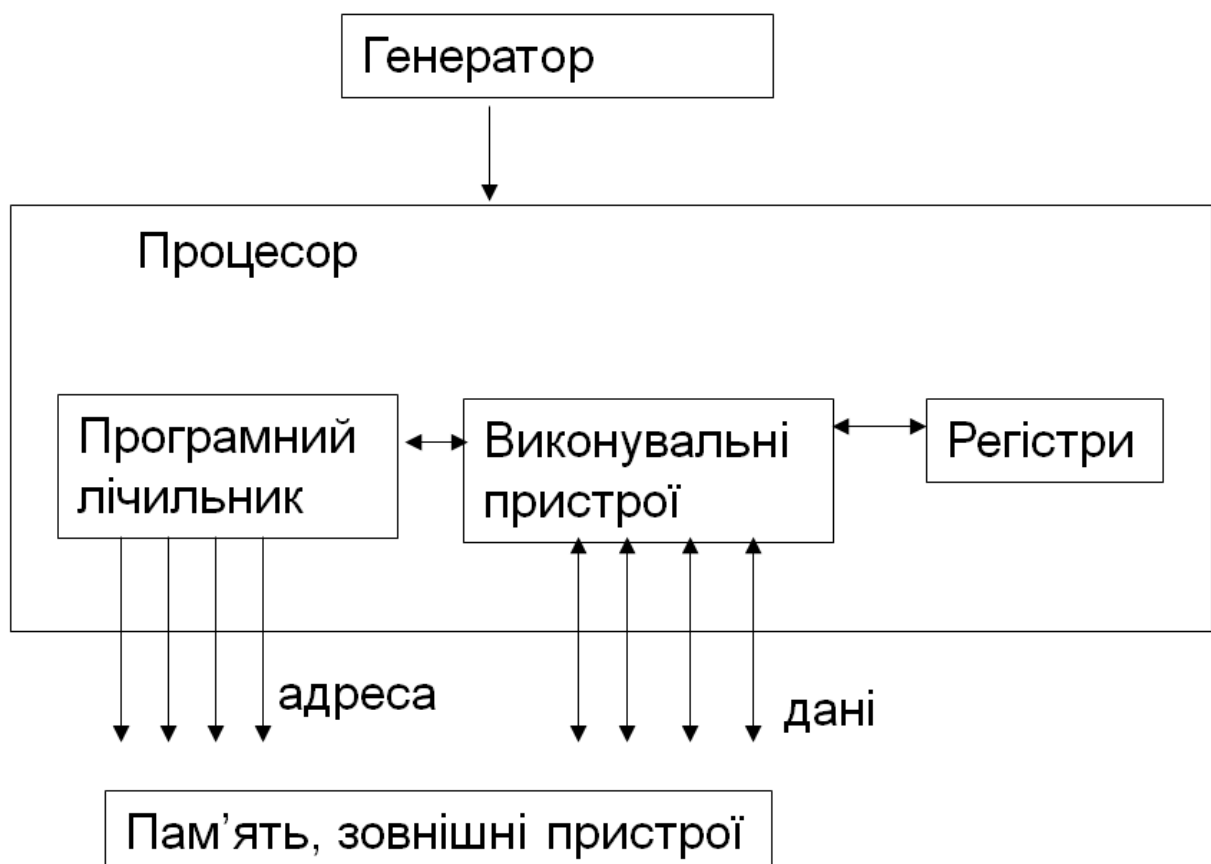
#### Застосування реєстрів

- Передача даних різним пристроям по одним проводам
  - Такт1 – видача даних і запис в реєстр пристрою 1
  - Такт2 – видача даних і запис в реєстр пристрою 2
- Виконання математичних операцій з декількома порціями даних
  - Такт1 – видача даних1 і запис в реєстр1
  - Такт2 - видача даних2 і запис в реєстр 2
  - Такт3 дані1+дані2 і запис в реєстр 3
  - Виконання команди – 3 такти
- Оперативна пам'ять
  - Такт 1 - По проводам адреси – дозвіл запису-читання в певний реєстр
  - Такт 1 По проводам керування – керування записом-читанням
  - Такт 1 По проводам даних – передача, або прийом даних
  - Такт 2 По проводам тактування – запис або читання
- Збільшення кількості функцій за рахунок зменшення швидкості

[4.10]

#### Схема процесора

- Процесор
  - Програмний лічильник
  - Виконувальні пристрої (АЛП)
  - Регістри
- Як працює
  - На шину адреси видається адреса команди
  - Пам'ять видає команду чи дані за цією адресою
  - Виконувальний пристрій виконує роботу
  - Результати записуються в реєстри або в пам'ять



[4.11]

Арифметико-логічний пристрій (АЛУ, ALU, пристрій керування, функціональний пристрій)

- Керування роботою і виконання функцій
  - Дешифратор команд
  - Буферні регістри (для тимчасового зберігання команд і даних)
  - Арифметичні пристрої (додавання, віднімання, множення)
- Розрядність АЛУ
  - Максимальна кількість бітів які можуть бути оброблені за один такт
  - 2,4,8,16,32,64
- Система команд
  - Які команди може виконати процесор
  - ADD, MUL, AND, XOR, NOR...

[4.12]

Регістри процесора

- **Програмний лічильник** (регістр адреси, вказівник інструкції, Instruction Pointer, IP, Program Counter, PC)
  - Адреса наступної комірки пам'яті
- **Регістри загального призначення** (GPR, РЗП)
  - A,B,C,D - аргументи і результати виконання команд
- **Акумулятори (A)**
  - Регістри, які по замовчанню використовуються для операцій
  - Команда **A=A ADD B** виглядає простіше **ADD B**
- **Регістр стану** (Status Register, SR, регістр прапорців, ознак)
  - Чи було перенесення, чи був результат операції 0, ...
- **Індексні регістри**
  - Спрощення адресації пам'яті
- **Регістри стеку** (магазинної пам'яті, BP,SP)
  - Спрощення адресації пам'яті

[4.13]

Система команд процесора

- Система команд (набір інструкцій)
  - Які дії може виконувати процесор
    - Арифметичні операції
    - Логічні операції
    - Запис-зчитування даних з пам'яті чи інших пристроїв
    - Виклик підпрограм ...
  - Набір команд жорстко "зашитий" в дешифраторі команд процесора
  - Кожна команда має машинний код (унікальну бітову комбінацію)
  - Команди зберігається в зовнішній пам'яті і завантажуються в процесор
  - Деякі команди мають аргументи, тоді треба завантажити команду і аргументи
  - Мова асемблера – кожна команда процесора – одна команда мови
- Різні команди виконуються впродовж різного часу
  - Командний цикл** – послідовність дій виконання команди
  - Мікрокоманда** – етап виконання команди
  - Регістр-регістр – швидкі команди (мало обміну із зовнішніми пристроями)
  - Регістр-пам'ять/ пристрій – повільніші команди (більше обміну із зовнішніми пристроями)
  - Команди без аргументів – швидші
  - Чим більше аргументів (1,2,3...) тим повільніші

[4.14]

#### Шина (BUS)

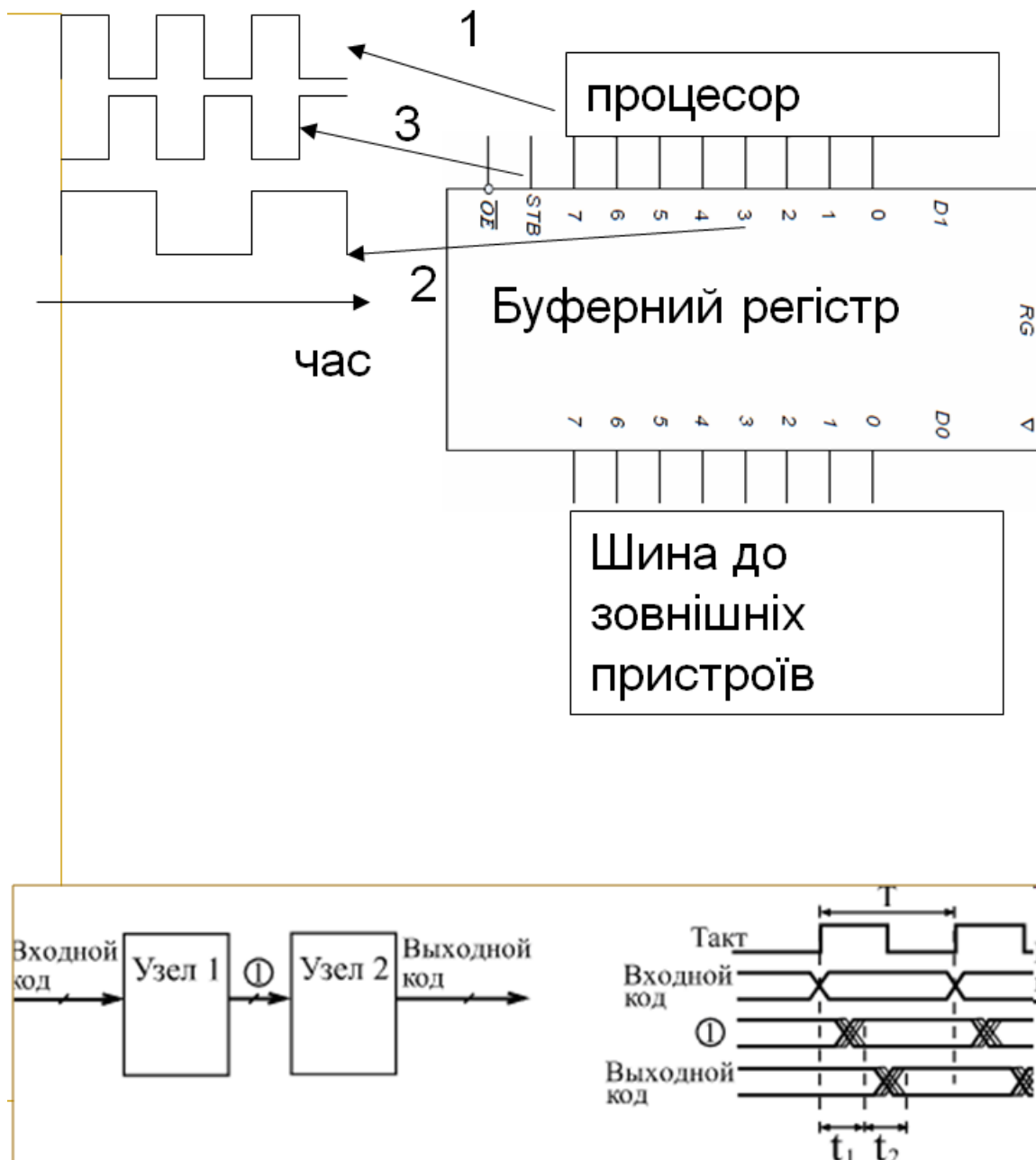
- **Шина** - набір проводів спільного призначення
  - Шина адреси – передача адреси команди чи даних в пам'яті
  - Шина даних – передача самої команди чи даних
  - Шина керування – керування пристроями
  - Внутрішні шини процесора
- До шин приєднано багато пристроїв
  - Внутрішні пристрої процесора
  - Пам'ять
  - Зовнішні пристрої
- Розрядність шини – кількість проводів (бітів)
- Контролер шини – пристрій який керує обміном по шині
  - Буферний регістр – регістр запису-зчитування даних на шині
  - Шинний формувач – порт з високою навантажувальністю здатністю
- Шини мають свої протоколи передачі інформації
  - Протокол – набір правил, якими керують пристрої для правильного обміну (1 передає – інші приймають тощо)

[4.15]

#### Контролер шини, буферні регістри, формувачі

- Багато пристроїв підключені до шини
  - Процесор не витримає такий струм
  - Можуть бути затримки через різні швидкості
- Буферний регістр
  - Висока навантажувальна спроможність
- Запис-зчитування даних
  - Тактовий імпульс 1 – виставлення даних процесором на шину
  - Дані доступні впродовж такту 2
  - Тактовий імпульс 3 (інверсний через півперіода) – запис в регістр і на шину
  - Дані передаються з затримкою на 1 такт
  - Відключення - переведення в стан високого вихідного опору (hold, утримання)
- Є двонаправлені, програмовані

## Тактові імпульси



[4.16]

### Розрядність

- Шина адреси – найбільша розрядність
  - 8,16,32,64,128
- Шина даних
  - 8,16,32,64
- Шина керування
  - Часто суміщена з даними чи адресою
- Розрядність процесора
  - Розрядність шини даних 8,16,32
  - Розмір машинного слова – порція даних з якою ефективно може працювати процесор 8,16,32,64
  - Максимальна адресована область – розрядність лічильника команд  $2^8, 2^{16}, 2^{32}, 2^{64}$
  - Максимальний об'єм пам'яті  $2^8, 2^{16}, 2^{32}, 2^{64}, 2^{128}$
  - Байт – одиниця передачі, адресації і збереження даних

[4.17]

Приклади команд процесора

- Без аргументів
  - HLT –зупинити роботу
  - MOV A,B пересилка з регістра B в A (A=B)
  - ADD B додати до акумулятора вміст регістра B
- З одним аргументом
  - MVI A,<B> пересилка значення <B> в регістр A (A=<B>)
- З одним аргументом великої довжини
  - LDA AB завантажити акумулятор значенням в області пам'яті з адресою AB
  - CALL AB викликати підпрограму за адресою області пам'яті AB
- Кожна вимагає певну кількість машинних тактів і циклів

[4.18]

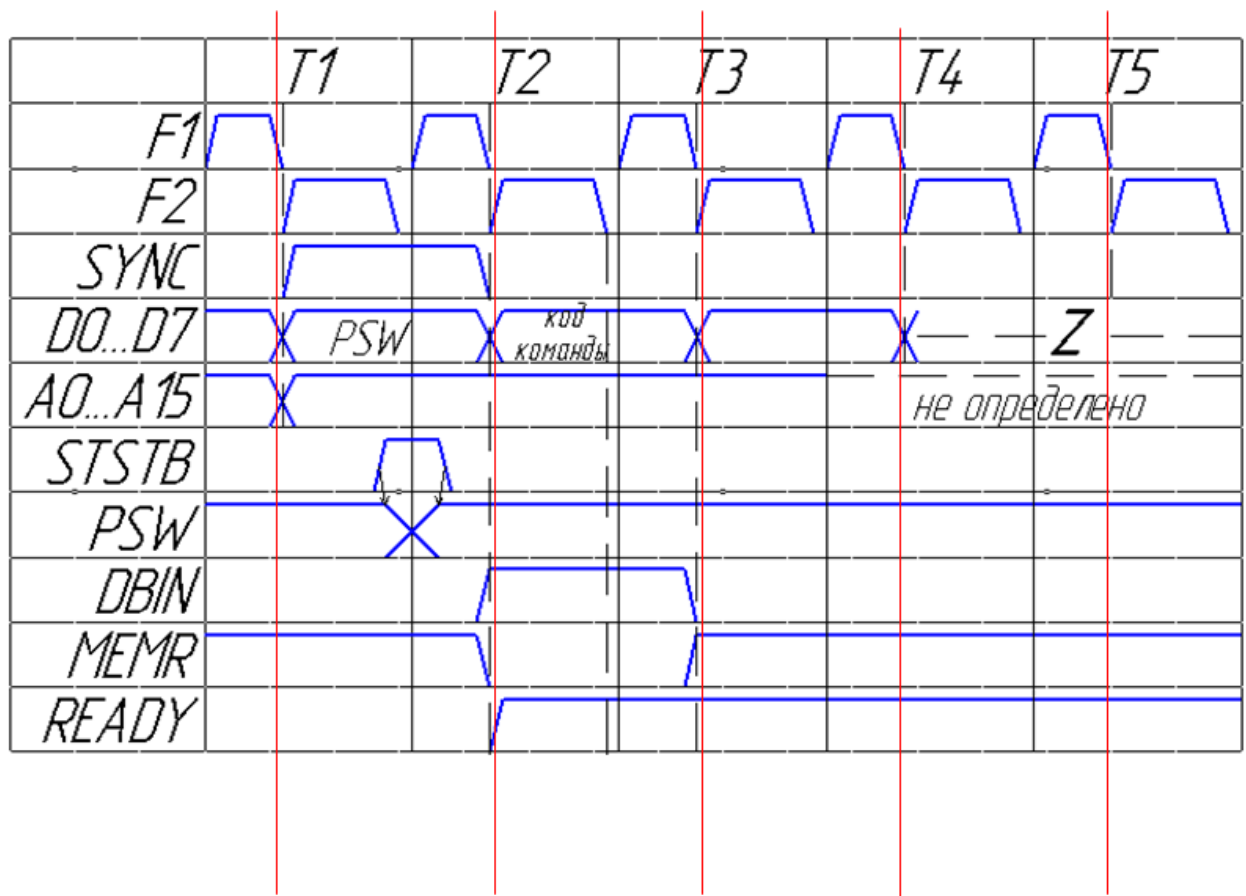
Такт, командний цикл, машинний цикл, мікрокоманда

- Час виконання команди – командний цикл
- Кожна пересилка даних по шині – машинний цикл
  - Командний цикл 1-7..10 машинних циклів
  - Пересилка по шині даних 1-2 байти
- Машинний цикл
  - Декілька мікрокоманд
- Мікрокоманда – дії за один такт
  - Такт 1. Видача на шину даних байта стану, на шину адреси – адреси команди, на шину керування сигналу синхронізації
  - Такт 2. Очікування готовності зовнішнього пристрою чи пам'яті
  - Такт 3 Отримання команди
  - .....
  - Такт N Виконання

[4.19]

Приклад машинного цикла

- Такт1 –видача адреси команди і слова стану
  - Інформація про те, що процесор “хоче” робити
- Такт 2 – перевірка готовності зовнішній пристроїв
  - Перевіряє поки не буде сигналу READY
- Такт 3 – читає команду по шині адреси
- Такт 4 – виконання команди
- Може бути багато циклів і тактів в залежності від команди
  - 1-7 циклів 3-25 тактів



В сучасних процесорах за такт може виконуватись декілька операцій

[4.20]

Особливості роботи з пам'яттю

- Кожен елемент даних (байт, слово) має адресу в пам'яті
  - При виставленні адреси на шину адреси на шині даних контролер пам'яті видає дані з цієї комірки
- Пряма адресація
  - Адрес пам'яті є параметром команди
  - Повільна передача по шині
- Непряма адресація
  - Адреса пам'яті знаходиться в регістрі
  - Швидка передача по шині
- Індексна адресація
  - Адреса пам'яті знаходиться в регістрі
  - В іншому регістрі знаходиться зміщення
  - Ефективно оптимізується
- Стек
  - В регістрі знаходиться адреса
  - При записі до адреси дані пересилаються і адреса зменшується на 1
  - При зчитуванні дані пересилаються і адреса збільшується на 1
  - Ефективна для викликів підпрограм

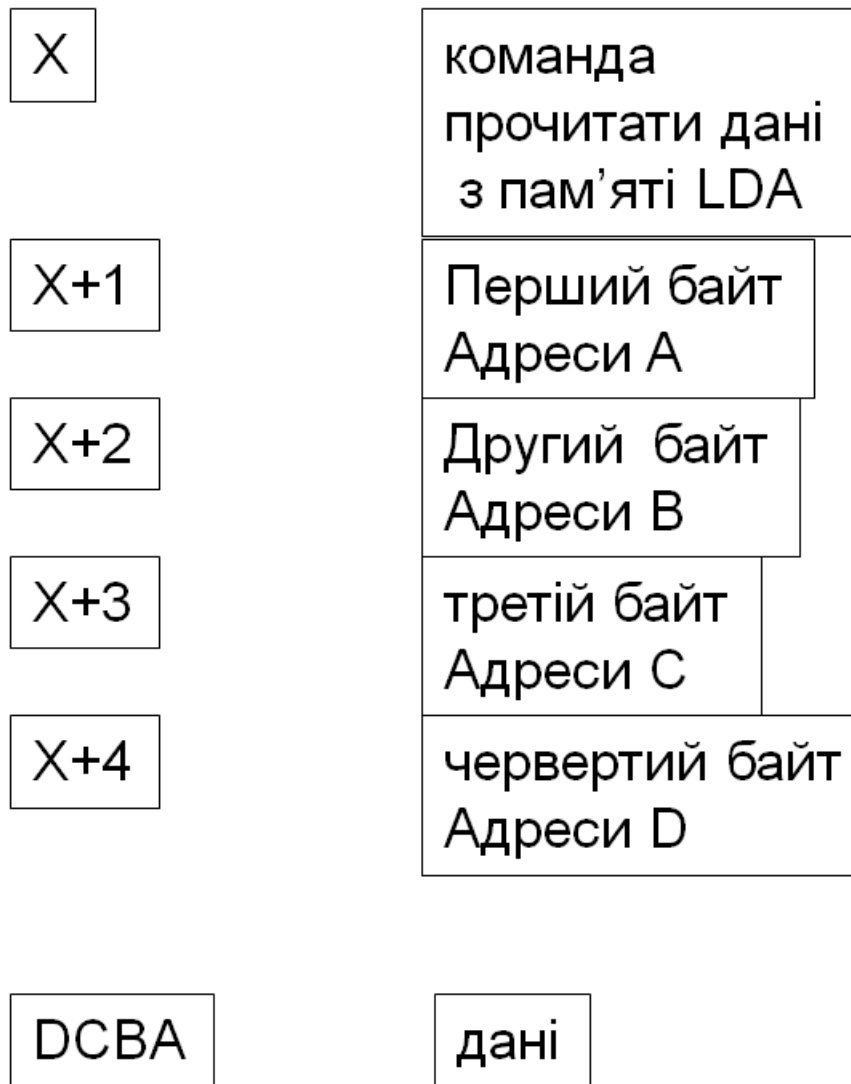
[4.21]

Пряма (безпосередня) адресація

- Адреса вказується як параметр команди
- LDA DCBA – завантажити в акумулятор дані, які зберігаються в пам'яті за адресою DCBA
- Застосовується рідко

- 5 циклів читання команди
- 1 цикл читання даних
- Дуже повільно (~20 тактів)
- Абсолютні адреси не дають можливість переміщувати дані в пам'яті

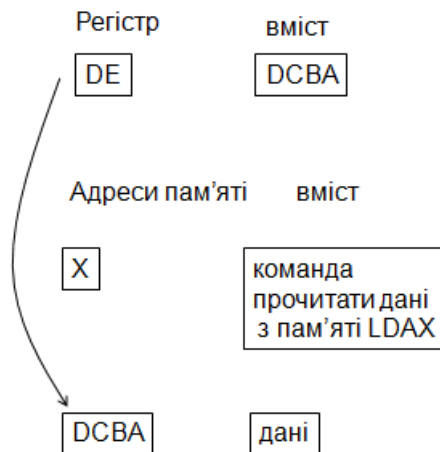
## Адреси пам'яті      ВМІСТ



[4.22]

Непряма (регістрова) адресація

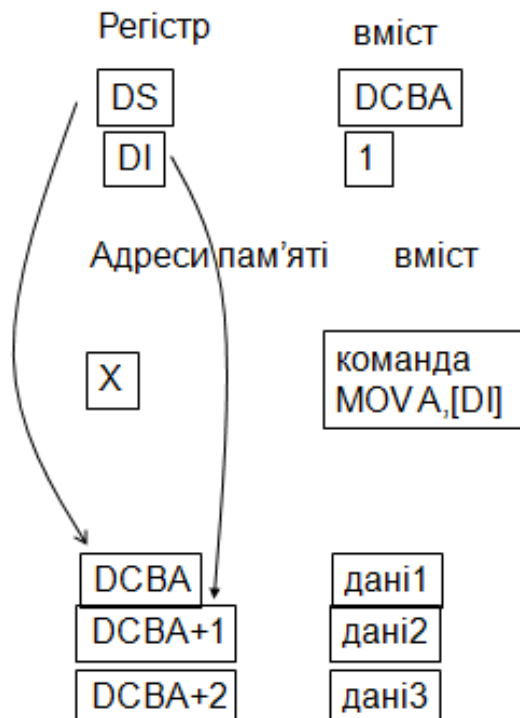
- Адреса зберігається в регістрі
- LDAX завантажити в акумулятор дані, які зберігаються за адресою пам'яті, яка записана в регістрі DE
- Застосовується часто
  - 1 цикл читання команди
  - 1 цикл читання даних
  - Швидко (~5 тактів)
  - Легко переміщувати дані в пам'яті



[4.23]

Індексна адресація – спеціальна форма непрямой

- Адреса зберігається в одному регістрі, зміщення – в іншому
- MOV A,[DI] завантажити дані, які містяться в пам'яті за адресою, яка рівна сумі значень в регістрі DS та DI
- Широко застосовується
  - 1 цикл читання команди
  - 1 цикл читання даних
  - Швидко (~5 тактів)
  - Ефективно працювати з масивами (кожна наступна команда 3 такти)



[4.24]

Пряма відносна адресація – спеціальна форма прямої

- В параметрі команди вказується зміщення відносно адреси команди
- Часто застосовується
  - 2 цикли читання команди
  - 1 цикл читання даних



- Можн апереміщувати програму в пам'яті

Адреси пам'яті      вміст

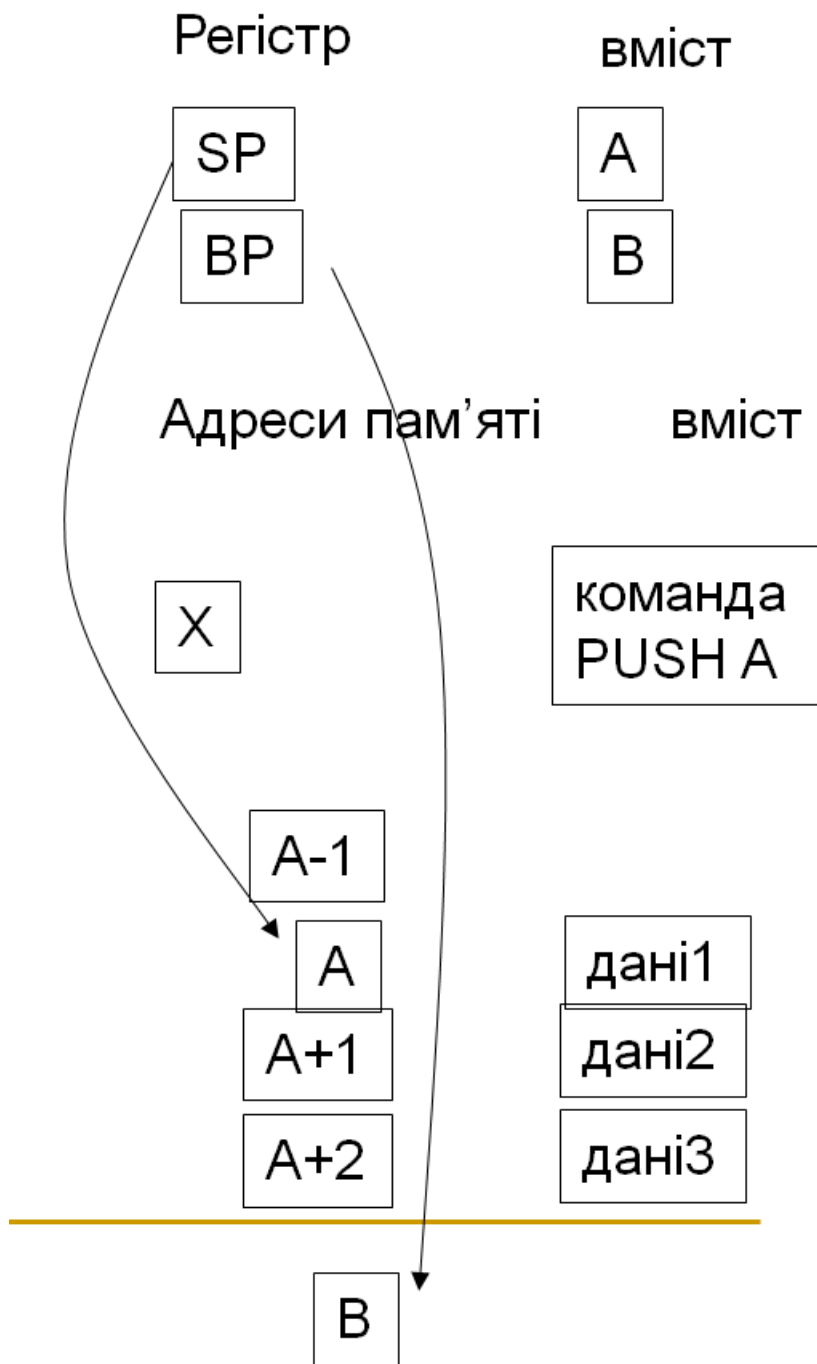
X	команда прочитати дані з пам'яті LDA
X+1	Зміщення A

X+A      дані

[4.25]

Стек - спеціальна форма непрямої адресації

- Останнім зайшов-першим вийшов
- В регістрі SP (stack pointer) зберігається адреса пам'яті де знаходяться останні записані дані
- Запис даних PUSH A
  - Значення SP зменшується на 1
  - Дані з A записуються за новим значенням SP (у вершину стека)
- Читання даних POP A
  - Дані за адресою SP зчитуються в регістр A
  - Значення SP збільшується на 1
- Регістр BP-основа стека ( $[SP] \geq [BP]$ )

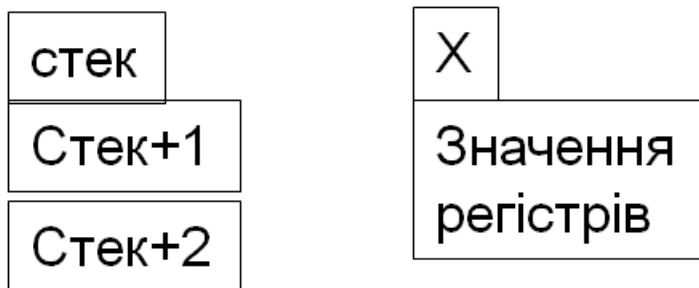
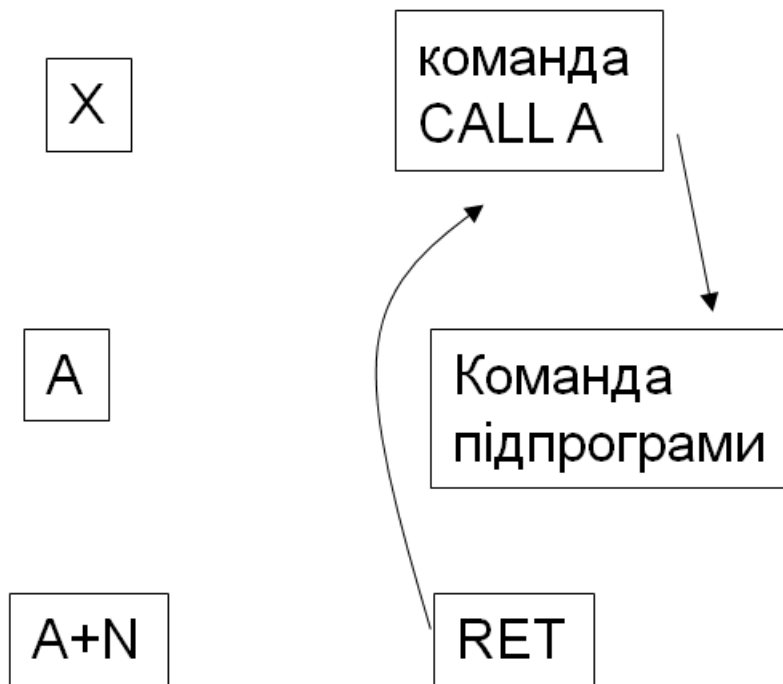


[4.26]

#### Виклик підпрограм

- Підпрограма – частина програми, яка може викликатись з різних частин програми
- Команда виконання підпрограми CALL ADDR
  - Запис в стек поточного значення командного лічильника
  - Запис в стек регістра стану і інших регістрів
  - Перехід за адресою ADDR
- Команда повернення з підпрограми RET
  - Відновлення із стеку регістра стану і інших регістрів
  - Відновлення командного лічильника

## Адреси пам'яті      вміст



[4.27]

### Області пам'яті

- Часто область пам'яті ділять на функціональні частини
- Сегмент коду (Code segment, CS)
  - Адреси пам'яті в яких зберігаються команди процесора
- Сегмент даних (Data Segment DS)
  - Адреси пам'яті в яких зберігаються дані
- Стек (Stack Segment, SS)
  - Адреси пам'яті в яких знаходиться стек
- Куча
  - Адреси пам'яті, які використовуються для тимчасового виділення пам'яті

[4.28]

### Архітектури ЕОМ – концепція побудови

- фон-Нейманівська архітектура
  - Процесор відділений від пам'яті
  - Система команд не міняється
  - Одна й та є сама пам'ять може використовуватись для команд, даних, стеку, кучі, тощо
  - Пам'ять адресується послідовно
  - Програма виконується послідовно
  - Надзвичайно поширена і універсальний підхід
- Гарвардська архітектура
  - Команди і дані зберігаються в різних областях пам'яті і передаються по різних каналах
  - Модифікована гарвардська архітектура – в середині і ззовні процесора різні шини адрес і даних
  - Переваги: пам'ять програм може бути більшою і дешевшою ніж даних, можлива паралельна передача команд і даних
  - Недоліки – не універсальна
- Гібридні архітектури
  - Різний кеш команд і даних, але спільна оперативна пам'ять

## Лекція 05

[5.1]

### Передача даних між компонентами мікропроцесорних систем

#### Лекція 5

Судаков О.О, Радченко С.П.

«Сучасна мікропроцесорна техніка»

[5.2]

#### Проблема взаємодії компонентів процесорних систем

- Велика кількість пристроїв взаємодіє між собою і з процесором
- Всі ці пристрої треба з'єднати між собою
- Застосовуються спільні шини (BUS, bidirectional universal switch )
- Застосувати окремі шини для кожного пристрою - не можливо
- Спільні шини – конфлікт при доступі
- Застосовуються різні шини і різні протоколи
  - Паралельні
  - Послідовні
  - Синхронні
  - Асинхронні

[5.3]

#### Спеціальні стани процесора

- Стан **читання-запису пам'яті (memory read/write)**
  - По шині адреси передаються адреси пам'яті
  - По шині даних передаються команди або дані
  - Спеціальні сигнали керування вказують на доступ до пам'яті
- Стан **читання-запису портів введення виведення (port in/out)**
  - По шині адреси передаються номер порта введення-виведення
  - По шині даних передаються команди або дані
  - Спеціальні сигнали керування вказують на доступ до зовнішніх пристроїв
- Стан **зупинки (stop)**
  - Процесор нічого не виконує
  - Спеціальні сигнали керування вказують на зупинку
- Стан **утримання (hold)**
  - Процесор нічого не передає по шині і переводить виводи шин стан високого опору
  - Спеціальні сигнали керування вказують на утримання
- Стан **переривання (interrupt)**
  - Процесор перериває виконання за запитом зовнішнього пристрою
  - Спеціальні сигнали керування вказують на переривання
- Стан **скидання (reset)**
  - Початковий стан
- Стан **ввімкнення**
  - Відразу після ввімкнення живлення

[5.4]

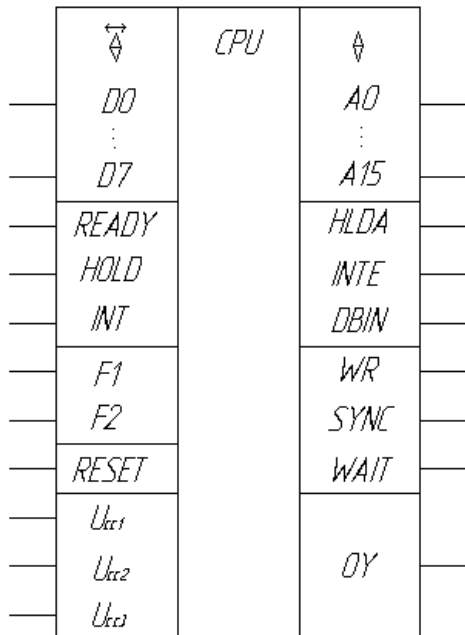
#### Стан ввімкнення (power on)

- Регістри процесора невизначені
- Процесор не можна використовувати поки він не буде переведений у стан ініціалізації
- При ввімкненні живлення деякі процесори вміють це обробляти
  - Переривання по ввімкненню живлення
- Деякі процесори вимагають ініціалізації
  - Штучно подати високий чи низький рівень на вивід reset
- Деякі процесори при збогах чи заниженні живлення виконують ініціалізацію

- Brown out reset

[5.5]

- Стан ініціалізації (скидання, reset)  
Виникає при подачі сигналу на вивід reset
- Всі регістри, крім регістрів лічильника команд (IP, Code Segment) не змінюються
- Регістри лічильника команд (IP, CS) встановлюються у певне значення (reset vector)
- Виконання починається з машинної інструкції за цією адресою

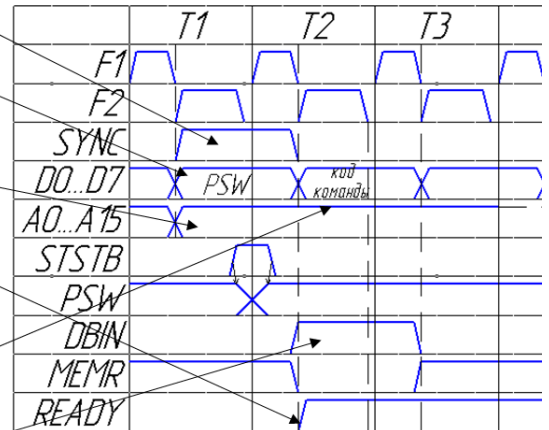


[5.6]

Цикл Читання (вибірка)-запис пам'яті (MEMORY)

- Такт 1
  - Проц. Сигнал синхронізації
  - Проц. Слово стану на шину даних
  - Проц. Спеціальний біт вказує читання/запис пам'яті
  - Проц. Адреса команди
- Такт 2 перевірка готовності
  - Проц. Перевірка сигналу ready
  - Проц. Перевірка сигналів переривання, захоплення
  - Очікування поки немає цих сигналів
- Такт 3
  - Пам'ять. Видає вміст пам'яті на шину даних
  - Пам'ять. Видає сигнал запису
- Аналогічні цикли можуть посторюватись

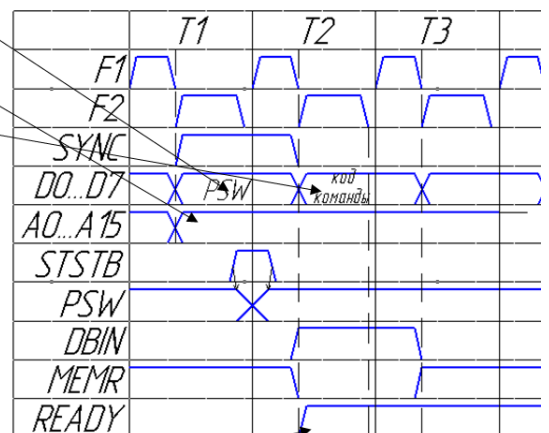
- Такт 1
  - Проц. Сигнал синхронізації
  - Проц. Слово стану на шину даних
  - Проц. Спеціальний біт вказує читання/запис пам'яті
  - Проц. Адреса команди
- Такт 2 перевірка готовності
  - Проц. Перевірка сигналу ready
  - Проц. Перевірка сигналів переривання, захоплення
  - Очікування поки немає цих сигналів
- Такт 3
  - Пам'ять. Видає вміст пам'яті на шину даних
  - Пам'ять. Видає сигнал запису
- Аналогічні цикли можуть посторюватись



[5.7]

Цикл читання порту вводу виводу (IO)

- Відрізняється від цикла читання пам'яті
  - бітами у слові стану
  - На шину адреси видається не адреса пам'яті а номер порта
  - Дані передаються не в пам'ять а ззовнішній пристрій
- Сама команда вводу- виводу одного байта триває декілька циклів
  - Цикл 1 - вибірка команди
  - Цикл 2 – вибірка номера порту
  - Цикл 3 – читання порту вводу виводу
- Пристрій повинен відслідковувати сигнали SYNC, шину даних, шину адреси, видавати сигнал готовності
- Відрізняється від цикла читання пам'яті
  - бітами у слові стану
  - На шину адреси видається не адреса пам'яті а номер порта
  - Дані передаються не в пам'ять а ззовнішній пристрій
- Сама команда вводу- виводу одного байта триває декілька циклів
  - Цикл 1 - вибірка команди
  - Цикл 2 – вибірка номера порту
  - Цикл 3 – читання порту вводу виводу
- Пристрій повинен відслідковувати сигнали SYNC, шину даних, шину адреси, видавати сигнал готовності



[5.8]

- Стан зупинки (stop, halt)
  - Процесор зупиняє роботу
  - При виконанні команди HLT

- Зупиняється в наступному циклі після видачі слова стану, де встановлений спеціальний біт зупинки
- Запустити процесор можна
  - Перериванням
  - Переведенням в стан захоплення
  - Переведенням в стан reset
- Використовується для налагодження програм, економії електроенергії

[5.9]

Стан захоплення (утримання,hold)

- Переведення всіх виводів шин у стан високого вихідного опору
  - Здійснюється в такті перевірки будь-якого циклу
  - Для переведення в hold зовнішній пристрій подає сигнал на вхід hold
  - Після переведення в стан hold процесор на виводі ahold встановлює сигнал підтвердження захоплення
  - Після цього зовнішні пристрої можуть передавати дані по шині
- Застосовується для обміну по шині без участі процесора
  - Прямий доступ до пам'яті (ПДП, Direct Memory Access, DMA)
- Зовнішній пристрій записує/читає дані в пам'ять без участі процесора
  - Швидко виконується ввід-вивід
  - Не споживається енергія процесором

[5.10]

Контролер ПДП

- Спрощує обмін ПДП
- Має декілька каналів
- Кожен канал – буферні регістри
  - Адреса
  - Дані
  - Керування
  - Стан
  - Лічильник
- При запиті на ПДП від пристрою
  - Запис в регістри номер каналу, адресу початку або кінця, кількість байт
  - Запит на ПДП
  - При підтвердженні захоплення шини пристрій передає дані контролеру
  - Контролер передає дані в пам'ять





#### [5.11]

##### Стан переривання (interrupt)

- Переривання – процесор перериває виконання програми і виконує запит від зовнішнього пристрою
  - ❑ Процесор повинен дозволити переривання команда IE
  - ❑ Для заборони переривання команда ID
  - ❑ В режимі ПДП переривання заборонені
  - ❑ З режиму STOP виводиться перериванням
  - ❑ Пристрій видає на вивід процесора INT сигнал
  - ❑ Процесор в кінці кожного циклу перевіряє запит на переривання
  - ❑ При наявності запиту на переривання процесор видає сигнал підтвердження на виводі INTA
  - ❑ Процесор чекає на команду на шині даних
  - ❑ При отриманні команди починається обробка переривання

#### [5.12]

- Після підтвердження переривання процесор зчитує на шині даних код команди, або число – індекс вектора переривань (номер переривання)
  - ❑ Вектор переривань – масив адрес за якими знаходяться підпрограми обробки переривань
  - ❑ Переривання 5 – елемент вектора 5
- Відбувається виклик підпрограми за адресою, яка відповідає номеру переривання у векторі (або виконання команди яку зчитали з шини)
- Після завершення обробки відбувається вихід із підпрограми
- Широко застосовується для обробки запитів від зовнішніх пристроїв
- Можна програмно змінювати код обробника, дозволяти і забороняти переривання

#### [5.13]

##### Контролер переривань

- Спрощує взаємодію із зовнішніми пристроями
- Приймає запити від пристроїв по виводам переривань (номер вивода номер переривання)
- Встановлює пріоритетність переривань
  - ❑ Коли обробляється переривання з меншим номером – переривання з більшим номером не обробляється
  - ❑ Таймер - найпріоритетніше

#### [5.14]

##### Спеціальні переривання

- Нештатні стани процесора можуть генерувати переривання
  - ❑ Переривання reset
- Немасковане переривання – переривання яке має найвищий пріоритет і забороняється, чи дозволяється окремо
  - ❑ Окремий вивід процесора (Non-Maskable Interrupt, NMI)
  - ❑ Генерується при фатальних помилках (пам'ять, тощо)
- Програмні переривання
  - ❑ Виклик вектора переривань

#### [5.15]

##### Шина ISA

- Industrial Standard Architecture
- Використовує порти вводу виводу, переривання, канали ПДП
- Пристрій має
  - ❑ Діапазон портів вводу-виводу IO RANGE
  - ❑ Номер переривання IRQ
  - ❑ Номер каналу DMA

- Діапазон адрес пам'яті DMA
- Лінії (62(ХТ)-98(АТ))
  - Адреса 20(ХТ)-24(АТ) ліній
  - Дані 8(ХТ)-16(АТ) ліній
  - Переривання до 11 ліній
  - Керування
- Швидкість 4-20 МГц
- Зараз використовується рідко в основному для серверів
- Проблема
  - Неможливо спільно використовувати переривання
  - Обмежена швидкість і кількість пристроїв
  - Проблеми з plug-and-play

[5.16]

#### Шина PCI (Peripheral Components Interconnect)

- Шина незалежна від типу процесора
- Швидкість 33, 66 та 133 МГц 32 та 64 біти
- Автоконфігурація
  - Пристрої визначаються автоматично
- Взаємодія з іншими пристроями та шинами за допомогою мостів
  - «Північний міст» – взаємодія з процесором, пам'яттю, швидкими пристроями
  - «Південний міст» – взаємодія з повільними пристроями вводу виводу
  - PCI-PCI міст
  - PCI-ISA міст
- Мости складні контролери, які забезпечують роботу шини
- Передача по шині без участі процесора
- Можлива паралельна передача даних і роботи процесора

[5.17]

#### Послідовні шини

- Паралельні шини швидкі, але є проблеми
  - Наводки від сусідніх ліній
  - Необхідність однаковості параметрів ліній
  - Необхідність гарантування синхронності передачі
  - Велика імовірність виходу з ладу
  - Дорого
  - Мала дальність передачі
- Послідовні шини
  - Прості
  - Можна масштабувати швидкість
  - Працюють на великій відстані і великій швидкості
  - Дешеві
- Дуже часто застосовуються
  - UART
  - SPI
  - I<sup>2</sup>C

[5.18]

#### UART (Universal Asynchronous Receiver Transceiver)

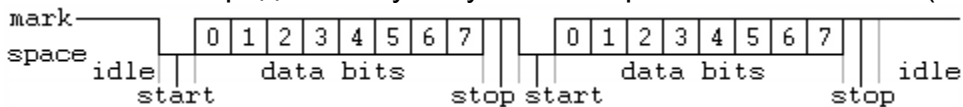
- Протокол передачі послідовною шиною
- Потрібно 2-3 проводи (одно двонаправлена передача)
  - Асинхронний – приймач і передавач не повинні мати спільного тактового генератора
  - Універсальний – підходить для будь-яких даних
  - Може приймати і передавати
- Дані представляються у вигляді потоку бітів (серіалізуються) і назад (десеріалізуються)
- Застосовується для керування пристроями
  - Консолі
  - Модеми
  - Програматори

- Недоліки
  - Повільний (стандарт до ~100 кБайт/с)
  - Великі накладні витрати (передається багато непотрібної інформації)

[5.19]

#### Передача-прийм

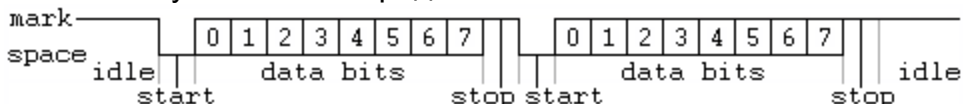
- Дані передаються по одному біту
- Приймач і передавач налаштовуються на однакову швидкість і однакові особливості протоколу передачі
- Біти кодуються рівнем 0- низький 1 –високий
- Передача кожного біту відбувається впродовж однакового фіксованого проміжку часу
  - Час передачі біта =  $1/(\text{швидкість передачі})$
- Дані передаються порціями по 1 байту
  - 1байт = 8-9 біт (налаштову)
- Перед порцією даних можуть бути стартові біти (завжди 0)
- Після передачі можуть бути біти парності і стопові біти (завжди 1)



[5.20]

#### UART – передавач

- В режимі коли нічого не передається передавач видає на лінію високий рівень (1)
- Передача байта починається з передачі стартового біту (0)
  - Передавач перемикається з високого на низький рівень
- Передавач чекає впродовж часу передачі біту
- Передавач перемикає на лінію у високий чи низький стан в залежності від значення бітів даних, які передаються і чекає впродовж часу передачі бітів
  - Так 8 або 9 разів
- Після передачі останнього біта з байта даний передавач передає стоповий біт
  - Перемикає лінію у високий стан і чекає час передачі біту
- Наступний байт передається аналогічно



[5.21]

#### UART-приймач

- Приймач чекає перепаду з високого на низький рівень
- Коли такий перепад трапляється приймач чекає половину часу передачі біту
  - Якщо все ще нуль (стартовий біт), то приймає далі
- Приймач чекає час передачі біту і зчитує поточне значення біту
  - Повторює так 8-9 разів і записує прийняті дані
- Приймач чекає час передачі біту і перевіряє наявність 1
- Якщо стартовий або стоповий біт неправильні – помилка
- Далі все повторюється



[5.22]

#### Швидкість передачі

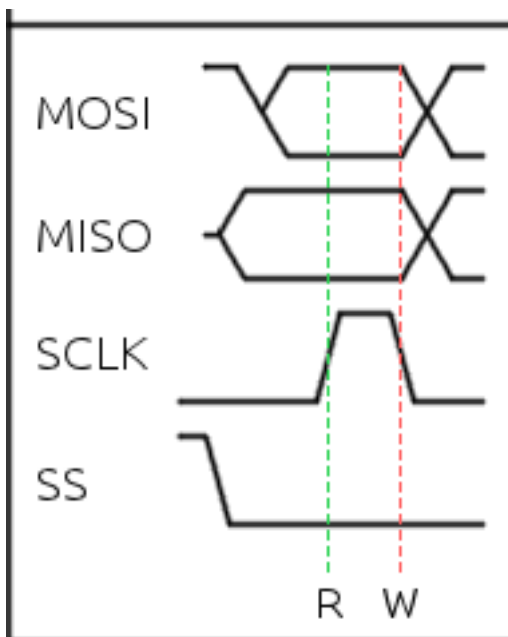
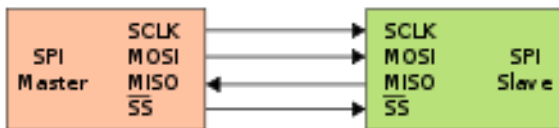
- Приймач і передавач повинні мати тактові генератори з **приблизно** однаковою частотою
  - Точність така, що за період передачі даних не повинна набігти похибка більша, ніж 0.5 часу передачі біта
  - 0.5біт/9.5 біт~5% (реально 1-3 %)

- ❑ Це обмежує максимальну швидкість, бо складніше забезпечити стабільність
- Швидкість передачі (бод, baud rate)
  - ❑ Кількість фізичних бітів за секунду
  - ❑ Швидкість передачі корисних даних менша
- Є стандартні швидкості
  - ❑ 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600
- Є стандарти на рівні напруг і протоколи
  - ❑ RS-232, RS-485, LIN, IrDa

[5.23]

Шина SPI (до 50 МГц)

- 1 Головний - багато ведених
  - ❑ 4 проводи
- Головний видає тактовий сигнал на всі ведені пристрої
- За допомогою виводу SS=0 головний вибирає ті пристрої з якими від взаємодіє
- Передача даних двонаправлена
  - ❑ Головний з виводу MOSI на вивід MISO веденого
  - ❑ Ведений з виводу MISO на вивід MOSI головного
- Виведення даних по спадтактового імпульсу
- Введення даних по фронту тактового імпульсу



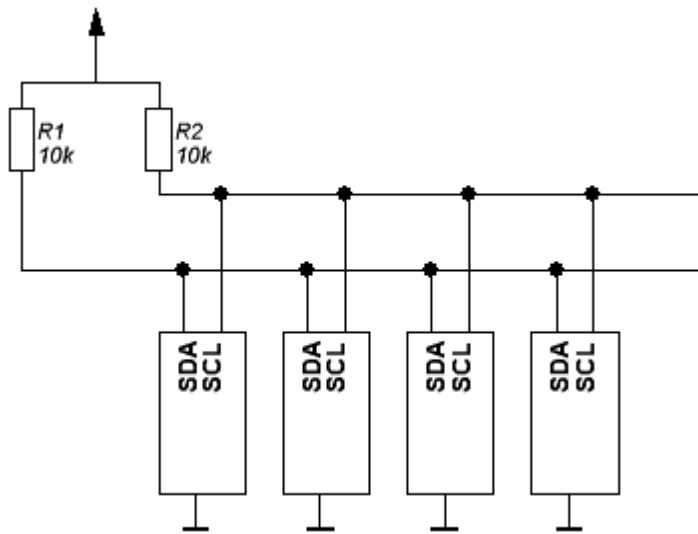
[5.24]

2

IC - Inter-Integrated Circuit (10 кбіт/с-3.4 Мбіт/с)

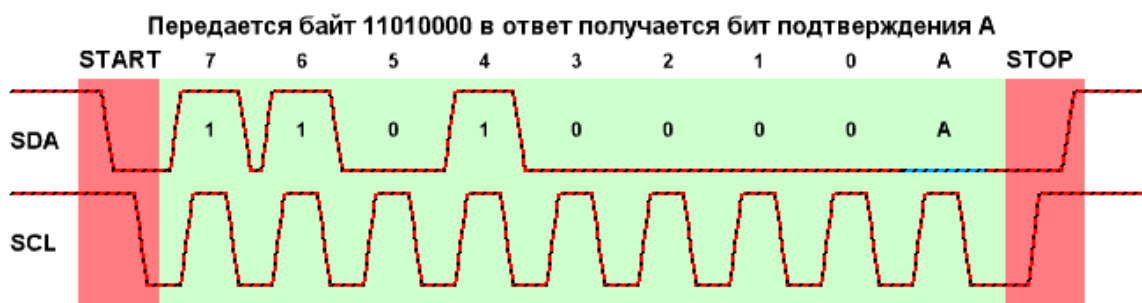
- Багато головних і багато ведених пристроїв
- 3 проводи
  - ❑ SDA – дані
  - ❑ SCL- такт
- Відкритий колектор, або стік
  - ❑ Низький рівень на шині виграє
- Тактовий сигнал видається лише головними, а читається всіма

- Дані видаються всіма, читаються всім
- Арбітраж – коли декілька одночасно
  - Поки всі дані однакові – нормально
  - Відключається той, хто бачить не свої дані



[5.25] 2

- І С – передача -прийом
- Запис на шину – при низькому такті
- Читання шину – при високому такті
- Старт – перехід старту і даних з високого в низьке значення
- Стоп – при високому такті відбувається запис на шину спочатку низького а потім високого значення
- Можна регулювати швидкість змінюючи тривалість нуля на тактовому проводі
- Потужний протокол, широко застосовується



[5.26]

#### Шина USB Universal Serial Bus

- USB 1.x,2.x,3.x
  - 4 проводу
  - Швидкість 1,5 Мбіт/с- 4,8 Гбіт/с).
  - Струм до 0.5 А
  - Дальність 3-5 м
- Гаряче підключення-відключення
  - Спочатку відключаються два проводу даних, потім два проводу живлення
- Ініціатором завжди є хост – комп'ютер тощо
- Дані передаються способом схожим на UART
  - Викорстовується декілька рівнів сигналу

## 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

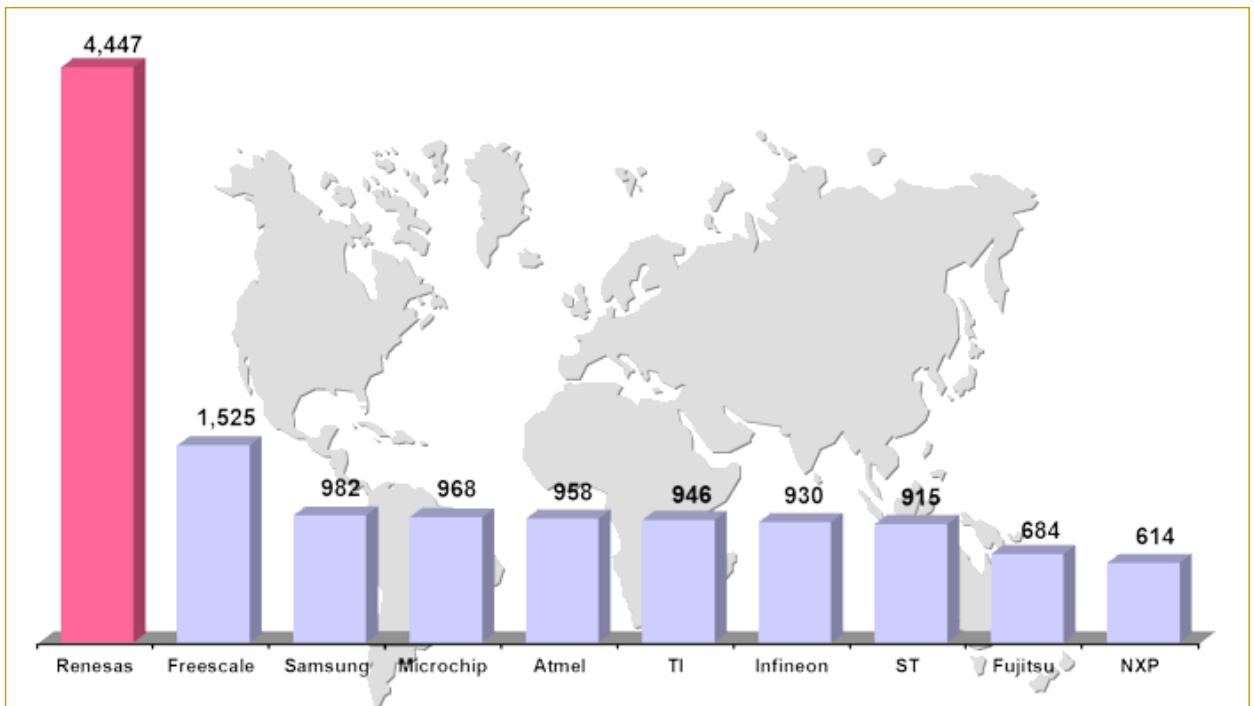
(Як правило стосується процесора)

- **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

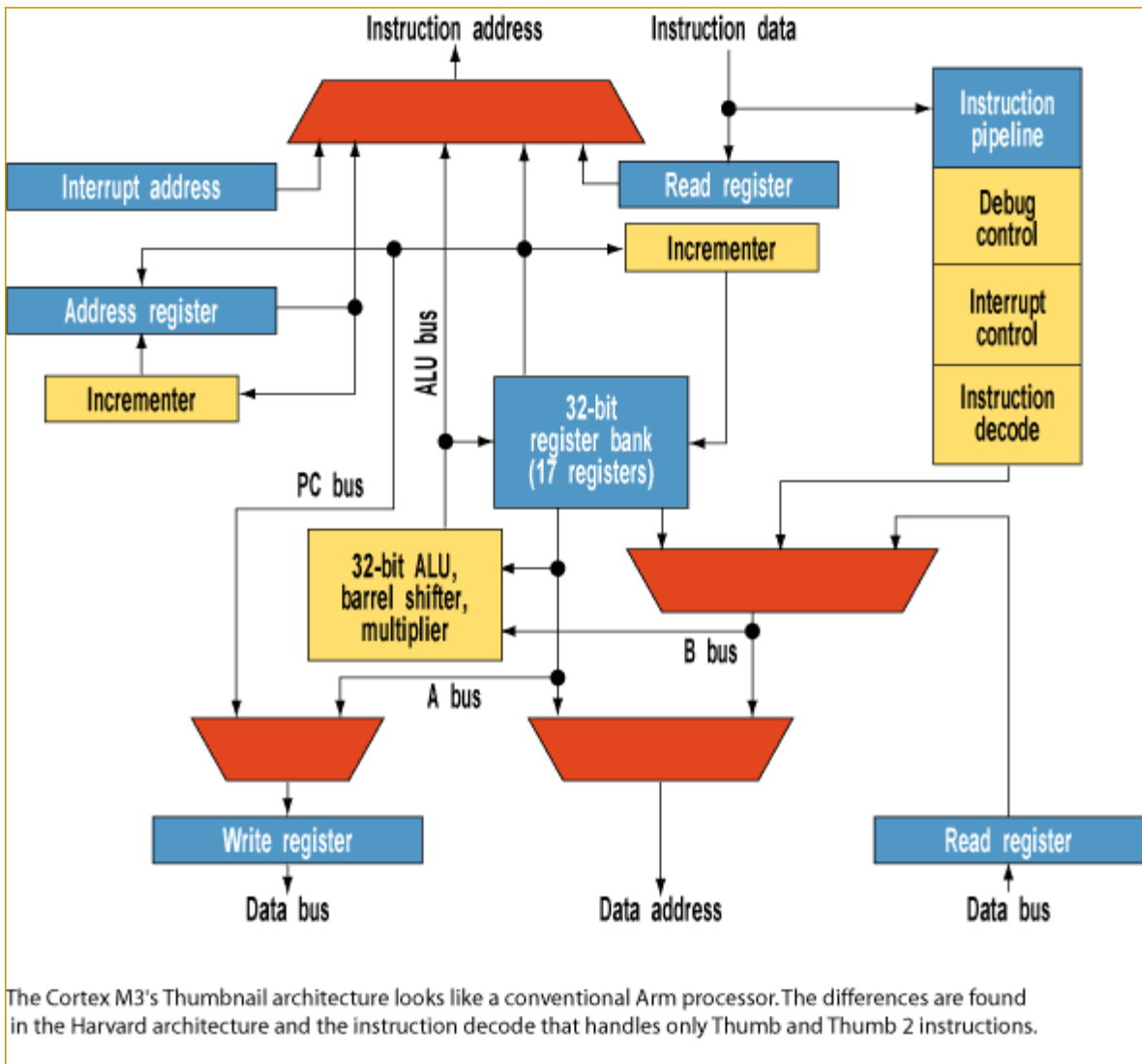


[6.7]

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

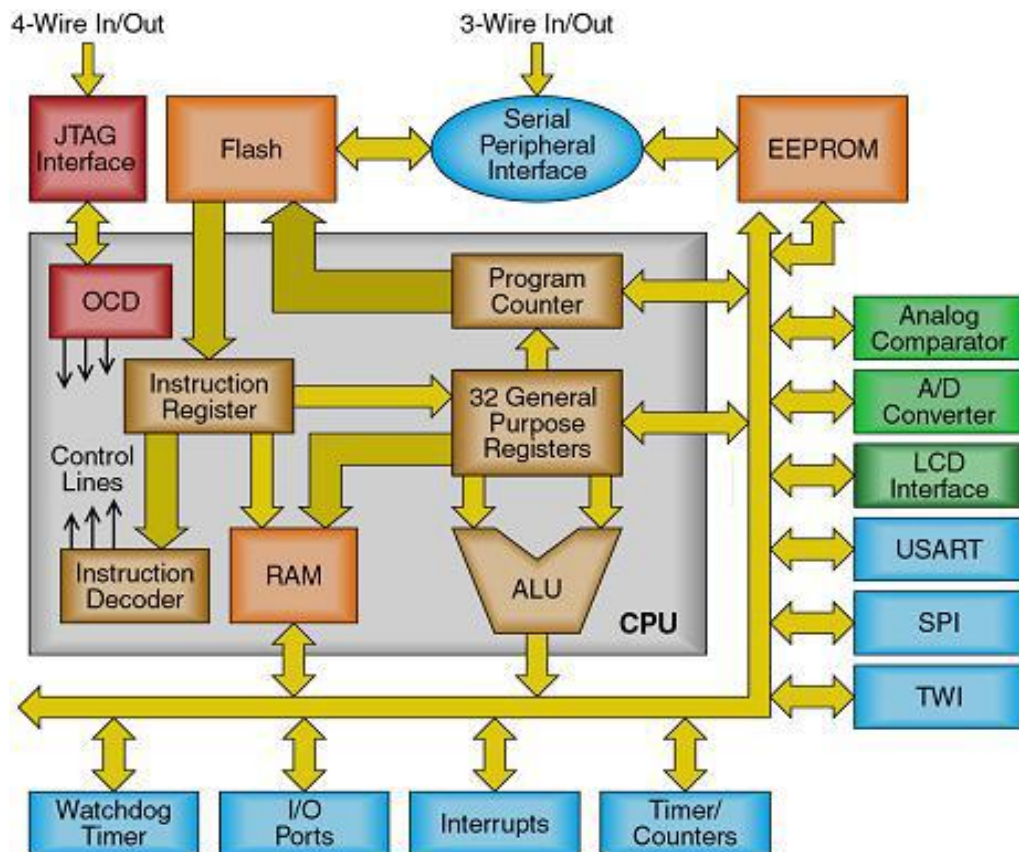




[6.8]

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

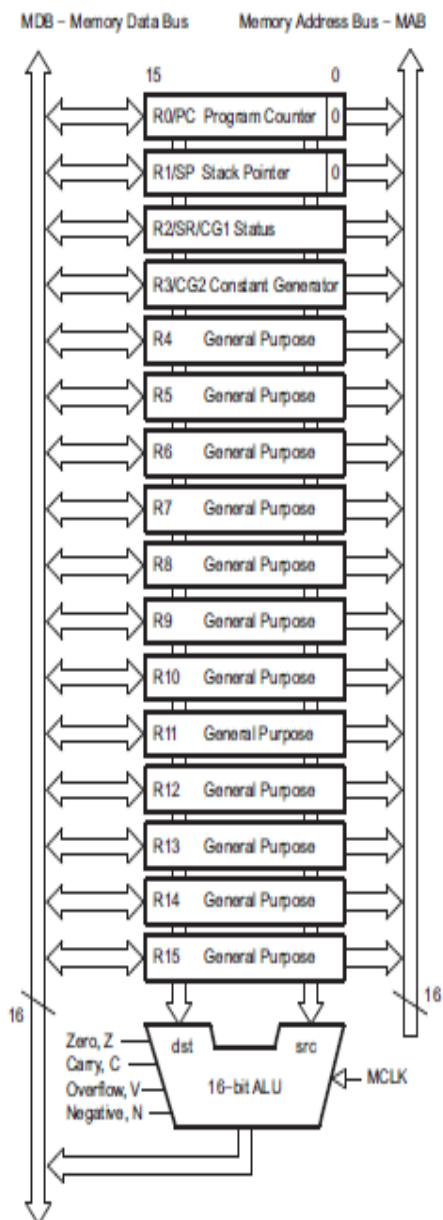
- Виробник ATMEL
  - ATmega
  - ATtiny
  - ..
- RISC процесор
  - 8 або 32 біти
- 16-32 регістра загального призначення
- Гарвардська архітектура



[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

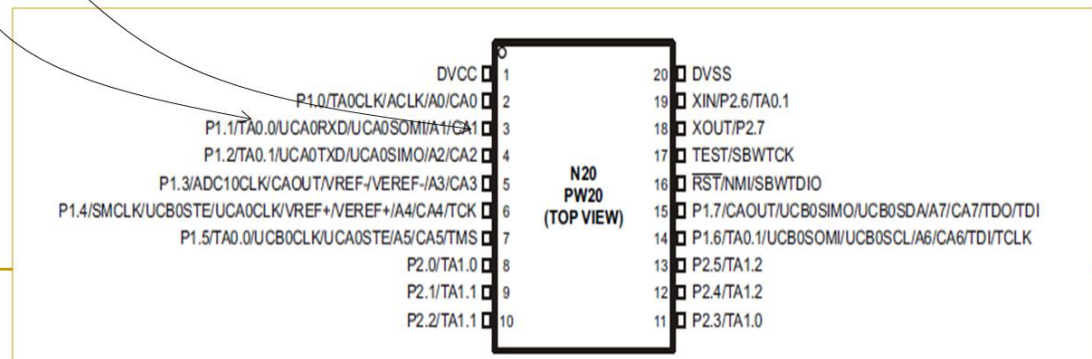
- **Кожен вивід мікросхеми може бути підключений до різних виводів різних пристроїв**

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

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

- Datasheet
- User guide

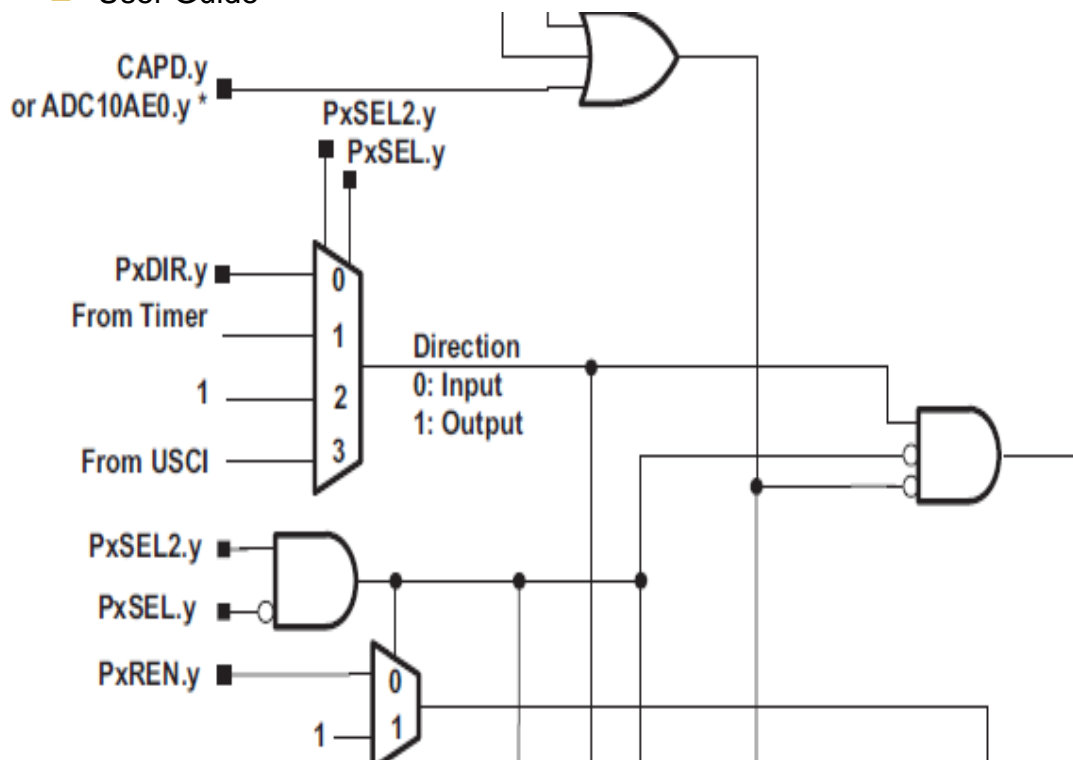
- Кожен вивід мікросхеми може бути підключений до різних виводів різних пристроїв
  - Таймер
  - Порт цифрового вводу-виводу
  - Компаратор
  - АЦП
- Повна інформація – в документації на мікроконтролер
  - Datasheet
  - User guide



[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]

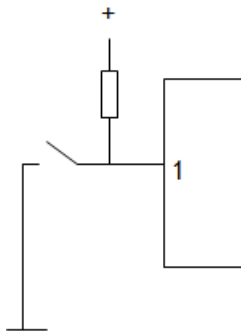
#### Програмування пристроїв

- 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

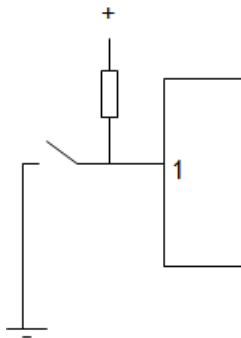
- Періодично перевіряється стан пристрою
  - Ніжки мікроконтролера
  - Регістра
- Приклад
  - Очікування натиснення на кнопку
  - Кнопка приєднана до ніжки 1
  - `while (P1IN & 0x01);`



[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