

## Maszyny liczące - rys historyczny

- *pierwszy kalendarz* - Stonehenge (obecnie Salisbury, Anglia) skonstruowany ok. 2800 r. pne.
- *abacus* - pierwsze liczydła (600-500 pne - Egipt lub Chiny)
- 1100 r. - pierwsza tabliczka mnożenia na piśmie
- 1612 - Szkot John Napier (1550-1617) odkrywa logarytmy i używa kropki dziesiętnej (wynalezionnej w Holandii)
- 1622 - William Oughtred (1574-1660) tworzy *suwak logarytmiczny*.
- 1623 - Wilhelm Schickard (1592-1635) skonstruował czterodziałaniowy kalkulator-zegar.
- 1642 - Blaise Pascal (1623-1662) tworzy "Pascalene" - 5-cio cyfrową *maszynę do dodawania*, uznaną za *pierwszą maszynę liczącą*.
- Gottfried Wilhelm von Leibniz (1646-1716) buduje czterodziałaniową maszynę liczącą

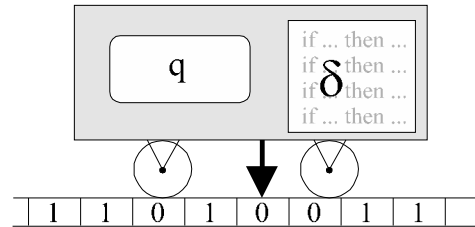
- 1903 - Nicola Tesla patentuje elektryczne bramki logiczne
- 1935-1938 - Konrad Zuse (1910- 1995) buduje **Z1** - *pierwszy komputer na przekaźnikach* (system dwójkowy).

## Komputer współczesny

- 1937 - **Alan Turing (1912-1954)** rozwija teorię maszyny uniwersalnej (wykonującej algorytmy)
- 1941 - Zuse tworzy **Z3** z wykorzystaniem arytmetyki zmiennoprzecinkowej
- 1943 - *Colossus* - komputer deszyfrujący
- 1944 - Howard Aiken (1900-1973) i inżynierowie z IBM budują Harvard Mark
- 1945 - John von Neumann publikuje ideę “maszyny z Princeton”
- 1943-1946 *ENIAC* - pierwszy komputer na lampach (Uniwersytet Pensylwania)
- 1948 - *EDSAC* komputer oparty na idei von Neumanna (Cambridge)
- 1954 - *IBM 704* - pierwszy komputer z systemem operacyjnym
- 1963 - *DEC PDP-5* - pierwszy minikomputer
- 1964 - komputery trzeciej generacji na obwodach scalonych

- 1971 - *Intel 4004* - pierwszy mikroprocesor
- 1974 - *procesor Intel 8080*
- 1981 - początek ery komputerów osobistych - pierwszy IBM PC

## Algorytm i maszyna Turinga



Opis formalny -  $\{Q, \Sigma, \delta, q_0, F\}$ , gdzie:

- $Q$  - zbiór *stanów maszyny*,
- $\Sigma$  - *alfabet* (zbiór symboli) taśmy,
- $\delta$  - *funkcja przejścia*:

$$\delta : Q \times \Sigma \longrightarrow Q \times \Sigma \times \{R, L, N\}$$

$R, L, N$  odpowiadają kierunkowi przemieszczenia się czytelnika na taśmie.

- $q_0$  - *początkowy stan*,
- $F$  - *zbiór końcowych stanów*.

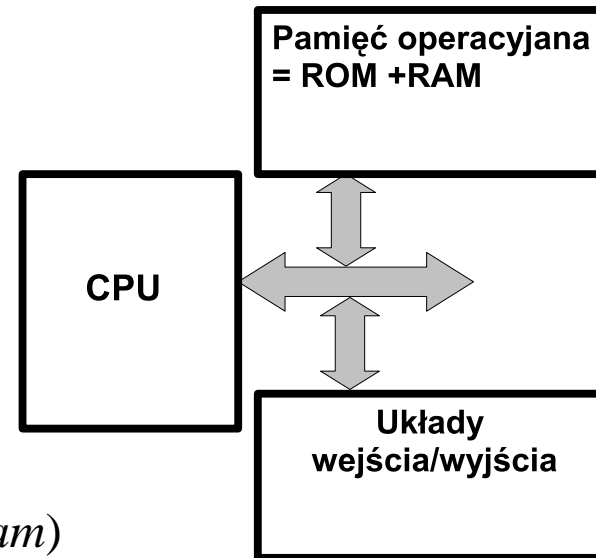
## Komputer von Neumanna - 1945

Elementy składowe komputera von Neumanna:

- procesor z ALU
- pamięć komputera (zawierająca *dane i program*)
- urządzenia wejścia/wyjścia

Cechy komputera von Neumanna:

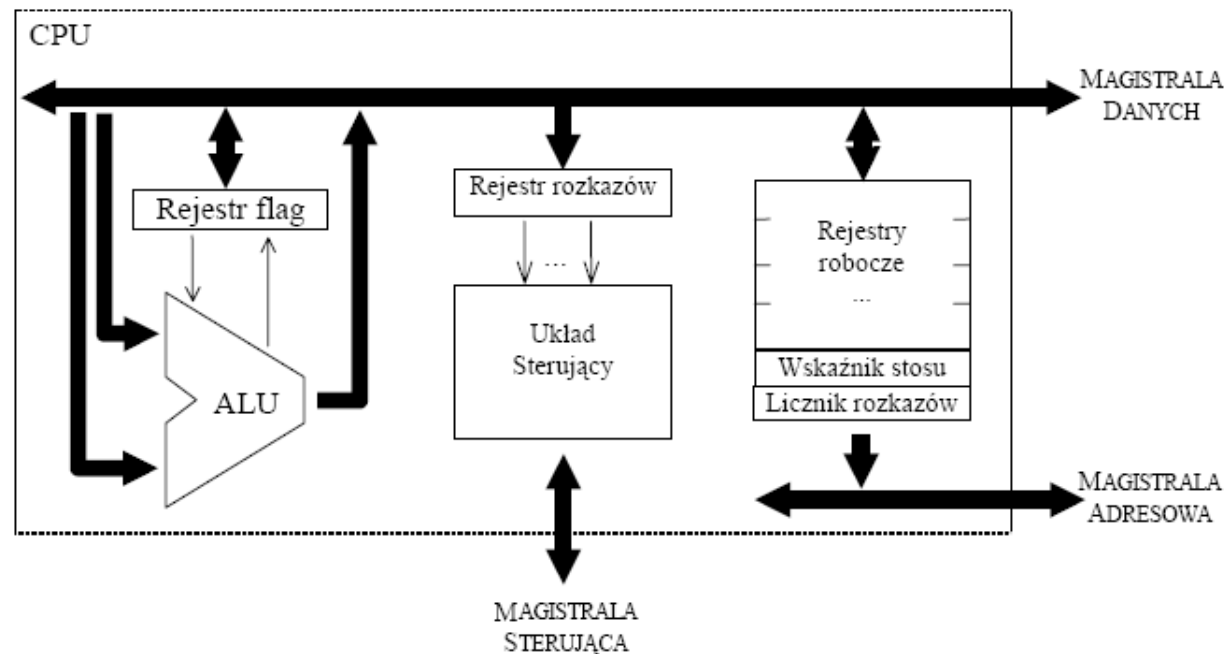
- skończona lista rozkazów,
- możliwość wprowadzenia programu i jego przechowywanie w pamięci (tak jak dane),
- sekwencyjne odczytywanie instrukcji z pamięci i ich wykonywanie.



## Architektura harwardzka

- *Pamięć danych programu* jest oddzielona od *pamięci rozkazów* (inaczej niż w architekturze von Neumanna).
- Prostsza, w stosunku do architektury von Neumanna, budowa ma większą szybkość działania - wykorzystuje się w procesorach sygnałowych oraz przy dostępie procesora do pamięci cache.
- *Architektura harwardzka* jest obecnie powszechnie stosowana w *mikrokomputerach jednoukładowych* (program w pamięci ROM (ang. Read Only Memory), dane w RAM (Random Access Memory)).

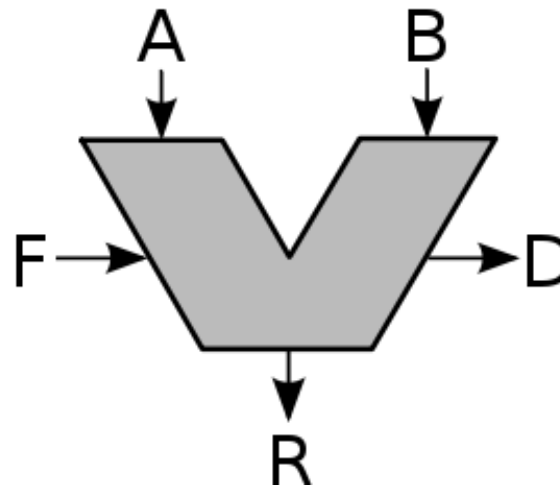
## Architektura procesora vs organizacja



Rysunek 1: Architektura procesora - funkcjonalna



## ALU i układ sterujący



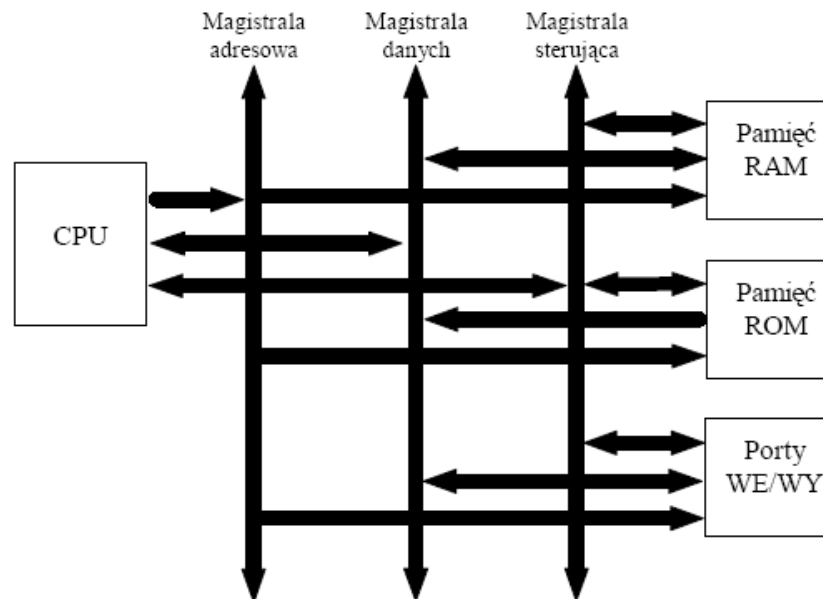
Rysunek 2: A i B - dane; R - wyjście; F - wybór operacji; D - status wyjścia

- *Jednostka Arytmetyczno-Logiczna* (ang. Arithmetic Logic Unit, ALU) układ kombinacyjny, wykonujący na danych w rejestrach operacje arytmetyczne (np. suma, różnica) oraz logiczne (np. OR, AND),
- *Układ sterujący* - dekoduje rozkazy i steruje jego wykonaniem.

## Rejestry procesorze

- *Akumulator A, ACC* - rejestr bezpośrednio współpracujący z ALU
- *Wskaźnik stosu SP* - wskazuje koniec stosu (wyróżnionego obszaru pamięci)
- *Licznik rozkazów PC* - adres komórki pamięci programu z następnym rozkazem do wykonania
- *Rejestr rozkazów* - zawiera kod rozkazu wykonywanego rozkazu
- *Rejestr flag* - zawiera flagi (znaczniki bitowe) ustawiane w zależności od wyniku wykonanej operacji (np. nadmiar, zero, bit parzystości)
- *rejestry ogólnego przeznaczenia* - robocze

## Magistrale systemu $\mu$ -procesorowego



- *magistrala adresowa* - przesyła adres (wybiera komórkę pamięci lub urządzenie we/wy),
- *magistrala danych* - przesyła dane między  $\mu$ P a pamięcią lub urządzenie we/wy),
- *magistrala systemowa* - zawiera sygnały sterujące.

## **Wielkość rejestru i słowo maszynowe**

*Słowo maszynowe* jest ilość informacji, przetwarzanej w jednym rozkazie, tj.

- odpowiada wielkości rejestra (wyrażonej w bitach),
- zazwyczaj jest wielokrotnością bajta,
- odpowiada szerokości magistrali danych.

## Pamięć jako tablica rejestrów

Organizacja pamięci:

- wielkość komórki pamięci (np. 8-bitowa, 1 bajt)
- wielkość szyny adresowej (np. 16-bitowa - zakres  $0 \div 2^{16} - 1$ )

*Rodzaje pamięci i technologie:*

- RAM - *Random Access Memory*
- ROM - *Read Only Memory*
- PROM - *Programmable ROM*
- EPROM - *Erasable ROM*
- EEPROM - *Electrically Erasable PROM* (pamięć Flash)

## Dekodowanie adresów

- *pełne* - w adresowaniu biorą udział wszystkie bity magistrali adresowej, wówczas każda komórka ma jednoznacznie określony adres,
- *niepełne* - w przeciwnym przypadku.

## Cykl maszynowy

Format rozkazu:

kod rozkazu	argumenty rozkazu
-------------	-------------------

Cykl rozkazu:

- *pobranie kodu rozkazu* - pobranie do rejestru rozkazu kodu rozkazu. Kody rozkazów przechowywane są w pamięci tak jak dane (architektura von Neumanna)
- *zdekodowanie rozkazu* - interpretacja wczytanego kodu rozkazu (zazwyczaj bajtu) jako polecenia z listy rozkazów procesora
- *wykonanie rozkazu* - wczytanie kolejnych argumentów rozkazu, w zależności od konkretnego rozkazu wykonanie ciągu operacji przez układ sterujący. Zapisanie wyniku w pamięci zewnętrznej lub rejestrze procesora

## Cechy architektury CISC

CISC (ang. *Complex Instruction Set Computers*) – nazwa architektury mikroprocesorów o następujących cechach:

- Występowanie złożonych, specjalistycznych rozkazów (instrukcji) - wymagają od kilku do kilkunastu cykli maszynowych (zmienna liczba cykli),
- Szeroka gama trybów adresowania (skomplikowana konstrukcja dekodерów adresu),
- Stosunkowo długa listy rozkazów procesora.

Wady architektury CISC:

- zbyt długa lista rozkazów - część z nich jest rzadko używana,
- zbyt dużo czasu traci się na operacje przepisania z pamięci do rejestrów i odwrotnie,
- ogólnie mała efektywność w obliczeniach numerycznych.



## Cechy architektury RISC

RISC (ang. *Reduced Instruction Set Computers*) - nazwa architektury mikroprocesorów o następujących cechach:

- Zredukowana liczba rozkazów do niezbędnego minimum
- Redukcja trybów adresowania, dzięki czemu kody rozkazów są prostsze, bardziej zunifikowane, (upraszcza dekodery rozkazów).
- Ograniczenie komunikacji pomiędzy pamięcią, a procesorem.
- *Przetwarzanie potokowe* (ang. *pipelining*) - równoległe wykonywanie rozkazów.

Obecnie popularne procesory z punktu widzenia programisty są widziane jako CISC, ale ich rdzeń jest RISC-owy. Rozkazy CISC są rozbijane na mikrorozkazy (ang. *microops*), które są następnie wykonywane przez RISC-owy blok wykonawczy.

## Języki programowania $\mu P$

- Języki wysokiego rzędu (np. VB, C, Java)
- Asembler

## **Mikrokontrolery**

*Mikrokontroler* - komputer zrealizowany w postaci pojedynczego układu scalonego, zawierającego jednostkę centralną (CPU), pamięć RAM, na ogół, pamięć programu oraz rozbudowane układy wejścia-wyjścia.

Określenie mikrokontroler pochodzi od głównego obszaru zastosowań, jakim jest sterowanie urządzeniami elektronicznymi.

## Budowa mikrokontrolerów

Typowy mikrokontroler zawiera:

- Jednostkę obliczeniową (ALU) - przeważnie 8-bitową,
- Pamięć danych (RAM),
- Pamięć programu,
- Uniwersalne porty wejścia - część tych portów może pełnić alternatywne funkcje, wybierane programowo,
- Kontrolery transmisji szeregowej lub równoległej (UART, SPI, I2C, USB, CAN, itp.),
- Przetworniki analogowo-cyfrowe lub cyfrowo-analogowe,
- timery,
- Układ kontroli poprawnej pracy (watchdog)
- wewnętrzne czujniki wielkości nieelektrycznych (np. temperatury)

## **Pamięć programu mikrokontrolerów**

Ze względu na rodzaj użytej pamięci programu, możemy wyróżnić mikrokontrolery:

- z pamięcią stałą (ROM) programowaną fabrycznie (jednokrotnie programowane)
- wyposażone w pamięć EPROM, EEPROM lub Flash (wymagające użycia zewnętrznych programatorów)
- z urządzeniu docelowym (In-System Programmable - ISP), przez interfejs szeregowy, bez własnej pamięci programu.

## Taktowanie mikrokontrolerów

Zegar systemowy mikrokontrolera może być taktowany:

- *zewnętrznym sygnałem taktującym* (rozwiązanie często stosowane w dużych układach wymagających synchronicznej współpracy wielu jednostek),
- *własnym generatorem*, wymagającym podłączenia zewnętrznych elementów ustalających częstotliwość taktowania (najczęściej jest to rezonator kwarcowy i dwa kondensatory),
- *wewnętrznym układem taktującym*, nie wymagającym podłączenia dodatkowych elementów

Zegary współczesnych mikrokontrolerów osiągają częstotliwości do kilkuset MHz, jednak w większości zastosowań taktowanie może być znacznie wolniejsze.

## Sposoby programowania mikrokontrolerów

*Mikrokontrolery* można programować na trzy sposoby:

1. *High voltage Programming* czyli sposób programowania wprowadzony ponad 15lat temu do programowania pamięci EPROM za pomocą sygnałów 12V.
2. *Emulacja pamięci ROM*
3. *ISP (In-System Programmable)* które nie wymaga wyjmowania układu z systemu w którym pracuje.
4. wykorzystuje *Bootloader*, czyli "kawałek" kodu który programuje nasz mikrokontroler np. przez port szeregowy, Bluetooth czy USB. Nie wymaga posiadania programatora poza pierwszym programowaniem (jakoś trzeba wgrać Bootloader).

## Przegląd obecnych mikrokontrolerów

Do najbardziej popularnych mikrokontrolerów należą:

1. Niekwestionowany standard dla rynku masowego narzuciła firma *Intel*, która wprowadziła na rynek mikrokontroler *8051*,
2. Bardzo popularne są również mikrokontrolery *AVR* firmy *Atmel* - w oparciu o nie będą prowadzona zajęcia laboratoryjne,
3. *PIC* firmy *Microchip Technology*,
4. inne.



## **Zadania na ćwiczenia**

1. Odczytaj z portu wartość słowa 8-bitowego. Warunkowo przekształć bajt danych w sposób wskazany przez prowadzącego. Wynik wyświetl na drugi port.