

# *Podstawy informatyki (1)*

**wykład : 15 godz.**

**ćwiczenia : 15 godz.**

**Prowadzący: dr inż. Jacek Piątkowski**

## Literatura

- ◆ J.G. Brookshear, *Informatyka w ogólnym zarysie*, WNT 2003,
- ◆ A.V. Aho, J.D. Ullman, *Wykłady z informatyki z przykładami w języku C*, Helion 2003,
- ◆ S.B. Lippman, J. Lajoie, *Podstawy języka C++*, WNT 2001,
- ◆ C.L. Tongo, B. P. Leung, *Podstawy języka C++ , ćwiczenia i rozwiązania*, WNT 2001,
- ◆ B. Eckel, *Thinking in C++. Tom 1 edycja polska*, Helion 2002,
- ◆ D. Harel, *Rzecz o istocie informatyki*, WNT 1992,
- ◆ J. Walter, D. Kalev, M. J. Tobler, P. Sainth, A. Kossoroukov, S. Roberts , *C++ w zadaniach*, Robomatic 1999

# Pojęcia podstawowe

## ❖ Algorytm

zbiór uporządkowanych, jednoznacznych i wykonywalnych kroków, które określają skończoną czynność.

przykłady :

- instrukcja obsługi pralki
- nutowy zapis utworu muzycznego
- sposób obliczania iloczynu liczb wielocyfrowych

☞ jeśli nie istnieje algorytm, za pomocą którego można rozwiązać określone zadanie, to jego rozwiązanie leży **poza zasięgiem komputerów**.

☞ poszukiwanie algorytmów to jedno głównych zadań informatyki.

# Pojęcia podstawowe

## ❖ **Program**

zrozumiała dla komputera reprezentacja algorytmu będąca zbiorem jednoznacznych instrukcji.

## ❖ **Języki programowania**

metody zapisu algorytmów w postaci zarówno wygodnej dla człowieka, jak i łatwo przekształcalnej do postaci rozumianej przez komputer.

## ❖ **Abstrakcja**

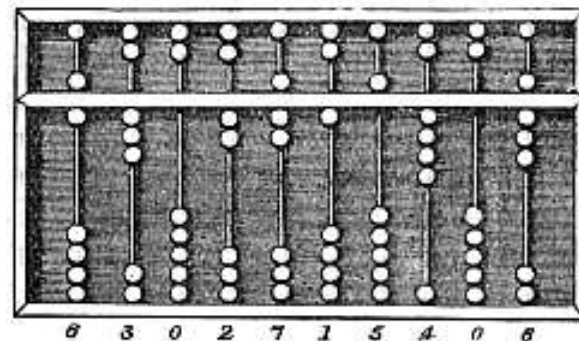
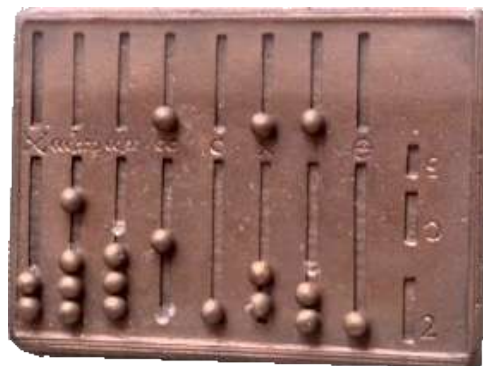
technika upraszczania pozwalająca na analizę złożonych zagadnień na różnych poziomach szczegółowości, pozwala skupić się na współpracy elementów tego samego poziomu i sposobie ich łączenia w elementy poziomu wyższego.

# Historia maszyn liczących

## ❖ **ABAKUS** 2600 lat p.n.e.

- deska z wyżłobieniami symbolizującymi kolejne potęgi dziesiątki
- obliczenia wykonywano przez przekładanie (w określony sposób) kamyków w wyżłobieniach.
- za wykonanie algorytmu odpowiedzialny człowiek

- ❖ .
- ❖ .
- ❖ .



☞ wynalazek używany przez starożytnych Greków i Rzymian w postaci różnych liczydeł stosowany praktycznie i w XX w.

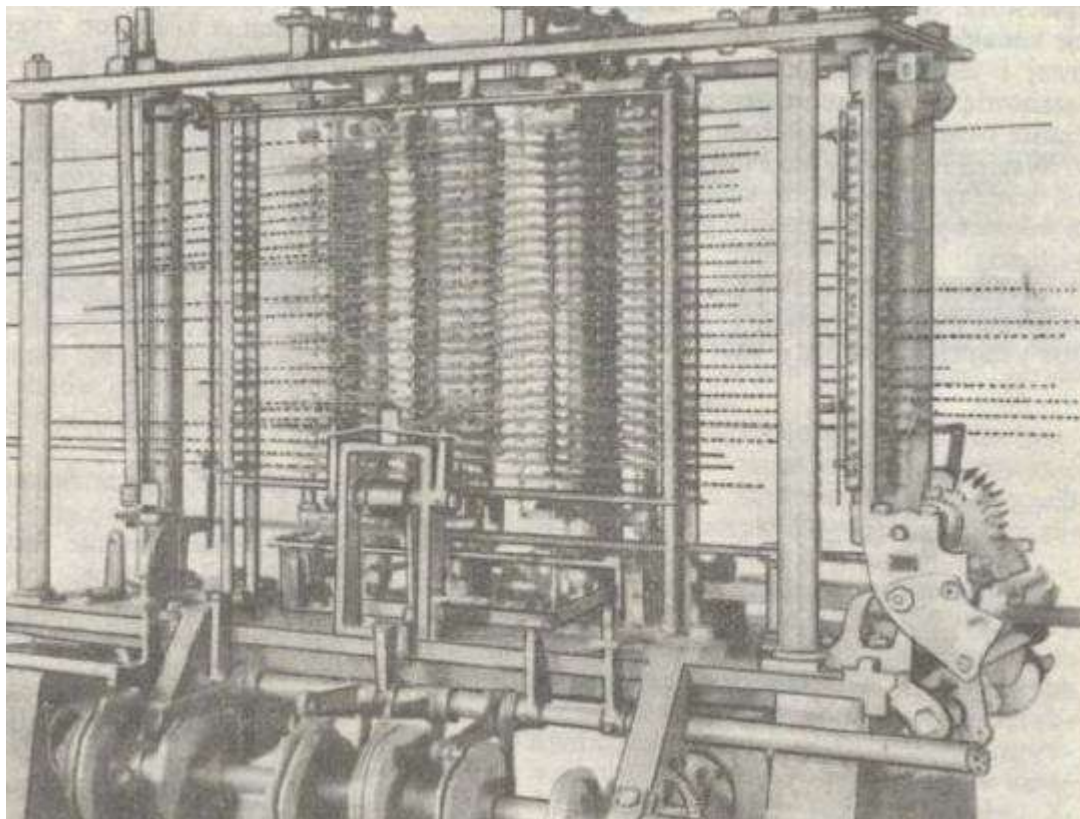
## Historia maszyn liczących

- ❖ 1623 Wilhelm Schickard – Niemcy,  
prof. matematyki i astronomii z Uniwersytetu w Tybindze przedstawia projekt zegara, który może wykonać 4 operacje arytmetyczne i wyciągnąć pierwiastek kwadratowy.
  - ❖ 1638 Blaise Pascal – Francja,  
rozpoczyna konstruowanie maszyny do realizacji algorytmu dodawania; wykonał 50 egzemplarzy, ale dopiero po 20 latach maszyna zadziałała poprawnie.
  - ❖ 1673 Wilhelm Leibniz – Niemcy,  
twierdzi, że opracował kalkulator wykonujący mnożenie i dzielenie.
- ☞ w maszynach tych dane reprezentowane były za pomocą pozycji kół zębatych.

# Historia maszyn liczących

- ❖ 1801 Joseph Jacquard – Francja, konstruuje maszynę tkacką, w której poszczególne kroki wykonywane w trakcie tkania definiowano za pomocą wzoru złożonego z otworów na karcie papierowej.
  - ❖ 1833 Charles Babbage, prof. astronomii w Cambridge, opracowuje maszynę, w której algorytm można było definiować za pomocą kart perforowanych.  
Asystentka Babbage'a - Augusta Ada Byron uznawana dziś bywa za pierwszą programistkę.
  - ❖ 1890 Herman Hollerith konstruuje maszyny rachunkowe do przeprowadzenia spisu powszechnego ludności w Stanach Zjednoczonych, w których nośnikami informacji są karty perforowane; to usprawnienie doprowadza do powstania IBM.
- ☞ możliwości współczesnej technologii nie nadążały za odkryciami teoretycznymi raczkującej informatyki aż do pojawienia się układów elektronicznych.

# Historia maszyn liczących



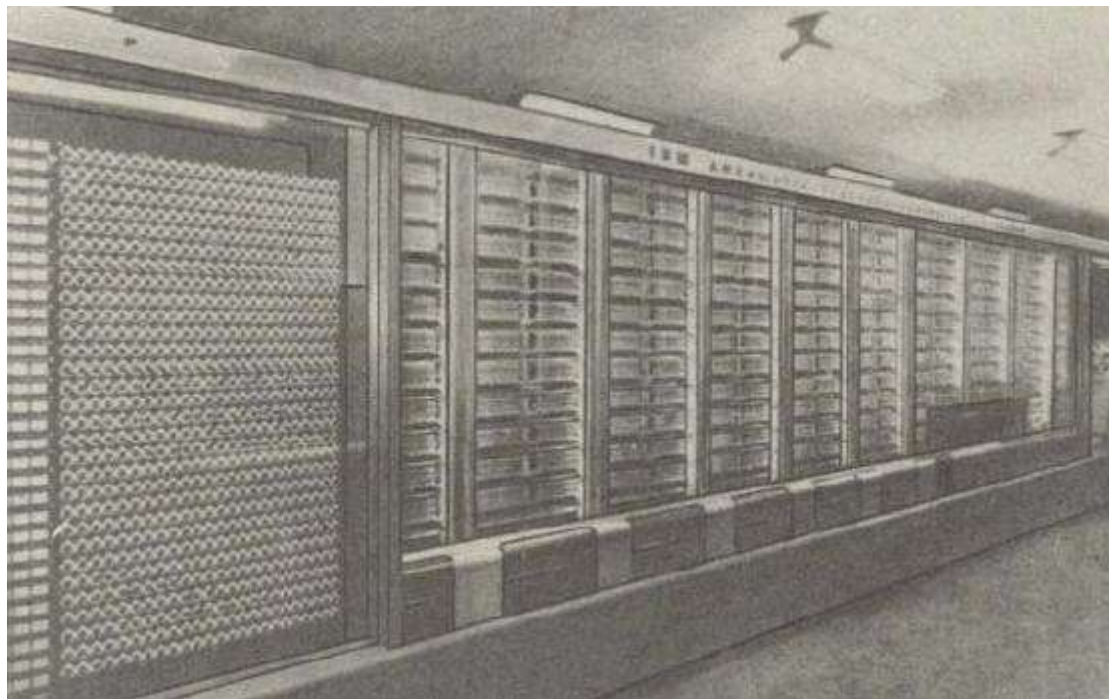
Maszyna analityczna Babbage`a



## Historia maszyn liczących

- ❖ 1940 Niemiec Konrad Zuse i Amerykanin George Stibitz niezależnie opracowują zasady obliczeń zmiennoprzecinkowych. W 1941 r. Zuse konstruuje prototyp maszyny Z3 do takich obliczeń.
- ❖ 1944 Howard Aiken z Harvard University wraz z grupą inżynierów z IBM konstruuje przekąźnikową maszynę MARK-I, wzorowaną na modelu maszyny Babbage'a, wykorzystującą taśmę papierową jako nośnik danych i programu.
- ❖ 1945 John von Neumann opisuje architekturę maszyny z oddzielnie zapisywanym programem. Idee von Neumanna stają się podstawą konstrukcji wszystkich maszyn, aż do czasów współczesnych.  
Rok później John von Neumann konstruuje EDVAC (Electronic Discrete Variable Computer).
- ☞ maszyny przekąźnikowe stawały się przestarzałe praktycznie natychmiast po ich skonstruowaniu.

# Historia maszyn liczących



## MARK-1

- dodanie lub odjęcie dwóch liczb 23 cyfrowych trwało 0,3 s,  
przemnożenie 6 s, a podzielenie 11,4 s

## Historia maszyn liczących

- ❖ 1945 r. John W. Mauchly i J. Presper Eckert - ENIAC,  
- 18.000 lamp, 90 m<sup>3</sup> objętości, 30 ton, 150 kW poboru mocy,  
5.000 działań na sekundę.  
Praktycznie liczydło - wykonane na zamówienie armii amerykańskiej do wyznaczania torów pocisków artyleryjskich i bomb.
- ❖ 1958 r. Jack Kilby z Texas Instruments zbudował pierwszy układ scalony, składający się z jednego tranzystora, trzech rezystorów i jednego kondensatora.
- ❖ 1972 r. firma INTEL Corporation wytwarza pierwszy mikroprocesor 4004, w którym było już 20.000 tranzystorów na jednym podłożu krzemowym.

## Historia maszyn liczących

- ❖ 1976 r. Steve Jobs i Stephen Wozniak zbudowali dostępny komercyjnie komputer i założyli firmę Apple Computer
- ❖ 1980 r. Seagate wprowadza pierwszy dysk twardy **5 MB** do komputerów osobistych
- ❖ 1981 r. IBM wprowadza na rynek pierwszy komputer osobisty PC z 16-bitowym procesorem **8088** , dla którego podstawowe oprogramowanie opracowała młoda firma Microsoft .
- ❖ 1982 r. Powstają: Compaq, Sun i Lotus.
- ❖ 1983 r. Bjarne Stroustrup opracowuje C++. Borland wprowadza Turbo Pascal.

## *Historia maszyn liczących*

- ❖ 1984 r. Microsoft prezentuje system operacyjny **Windows**.
- ❖ 1989 r. Anglik Tim Berners-Lee rozpoczyna badania nad uniwersalną wymianą danych w sieci Internet. Rok później powstaje protokół HTTP. Nazwa World Wide Web pojawia się w 1991 r.
- ❖ 1991 r. **Linus Thorvalds** opracowuje system operacyjny **Linux**.
- ❖ 2000 – INTEL wprowadza na rynek mikroprocesor Pentium IV

## Historia maszyn liczących

### ❖ 1945 – ENIAC

18.000 lamp

90 m<sup>3</sup> objętości

150 kW pobór mocy

### ❖ 1972 - mikroprocesor 4004

20.000 tranzystorów

25 cm<sup>3</sup> objętości ( 3,6 \*10<sup>6</sup> mniej )

10 W pobór mocy (1,5 \*10<sup>3</sup> mniej )

### ❖ 2000 – INTEL wprowadza na rynek mikroprocesor Pentium IV

- na płytce o powierzchni 42 cm<sup>2</sup> umieszczono 40 \*10<sup>6</sup> tranzystorów.

## Przechowywanie informacji

- ❖ Współczesne komputery przechowują informacje w postaci ciągu bitów.

**Bit** – *binary digit* – to cyfra 0 albo 1.

- ❖ Zapamiętanie bitu przez maszynę liczącą wymaga istnienia w niej urządzenia zdolnego do jednego z dwóch możliwych stanów.

przykłady :

- przełącznik – „włączony” lub „wyłączony”
- przekaźnik – „otwarty” lub „zamknięty”
- kondensator – „naładowany” lub „rozładowany”

☞ Brak zmian to brak informacji !

# Operacje logiczne

1 – prawda

0 – fałsz

- ❖ Operacje, które manipulują wartościami logicznymi prawda/fałsz (*true/false*) nazywa się operacjami logicznymi lub **boole'skimi** \*.

Operacja AND	0	0	1	1
	AND 0	AND 1	AND 0	AND 1
	<hr style="width: 100%;"/>	<hr style="width: 100%;"/>	<hr style="width: 100%;"/>	<hr style="width: 100%;"/>
	0	0	0	1
Operacja OR	0	0	1	1
	OR 0	OR 1	OR 0	OR 1
	<hr style="width: 100%;"/>	<hr style="width: 100%;"/>	<hr style="width: 100%;"/>	<hr style="width: 100%;"/>
	0	1	1	1
Operacja XOR	0	0	1	1
	XOR 0	XOR 1	XOR 0	XOR 1
	<hr style="width: 100%;"/>	<hr style="width: 100%;"/>	<hr style="width: 100%;"/>	<hr style="width: 100%;"/>
	0	1	1	0

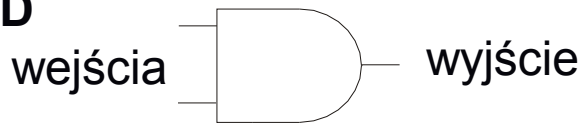
\* na cześć matematyka George'a Boola ( 1815 – 1864 )



# Operacje logiczne

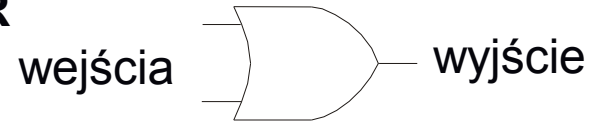
❖ **Bramki logiczne** – urządzenia, które na podstawie wejściowych tworzą wartość wyjściową zgodnie z określoną operacją logiczną

## AND



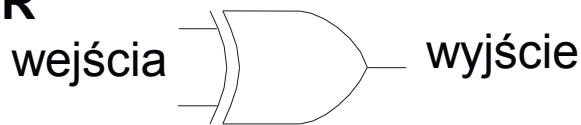
wejścia		wyjście
0	0	0
0	1	0
1	0	0
1	1	1

## OR



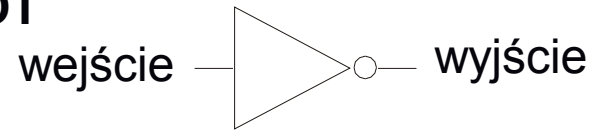
wejścia		wyjście
0	0	0
0	1	1
1	0	1
1	1	1

## XOR



wejścia		wyjście
0	0	0
0	1	1
1	0	1
1	1	0

## NOT



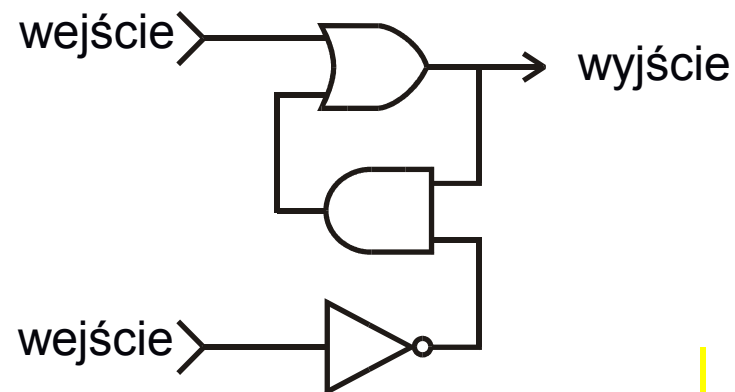
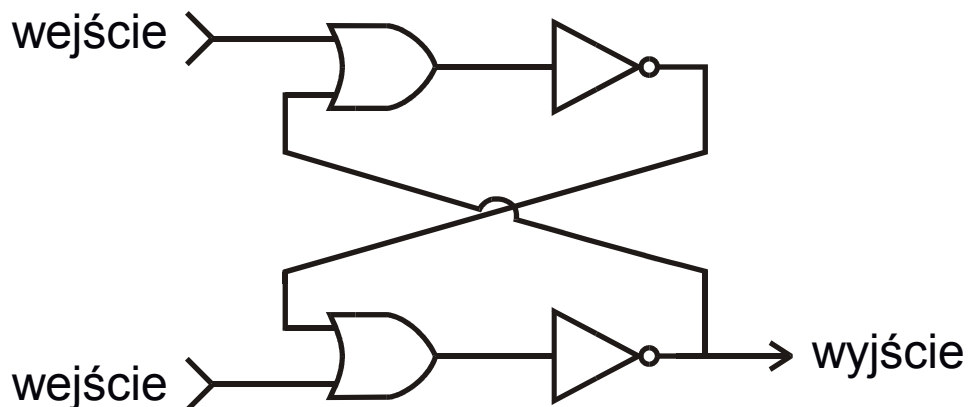
wejście	wyjście
0	1
1	0

## Bramki logiczne

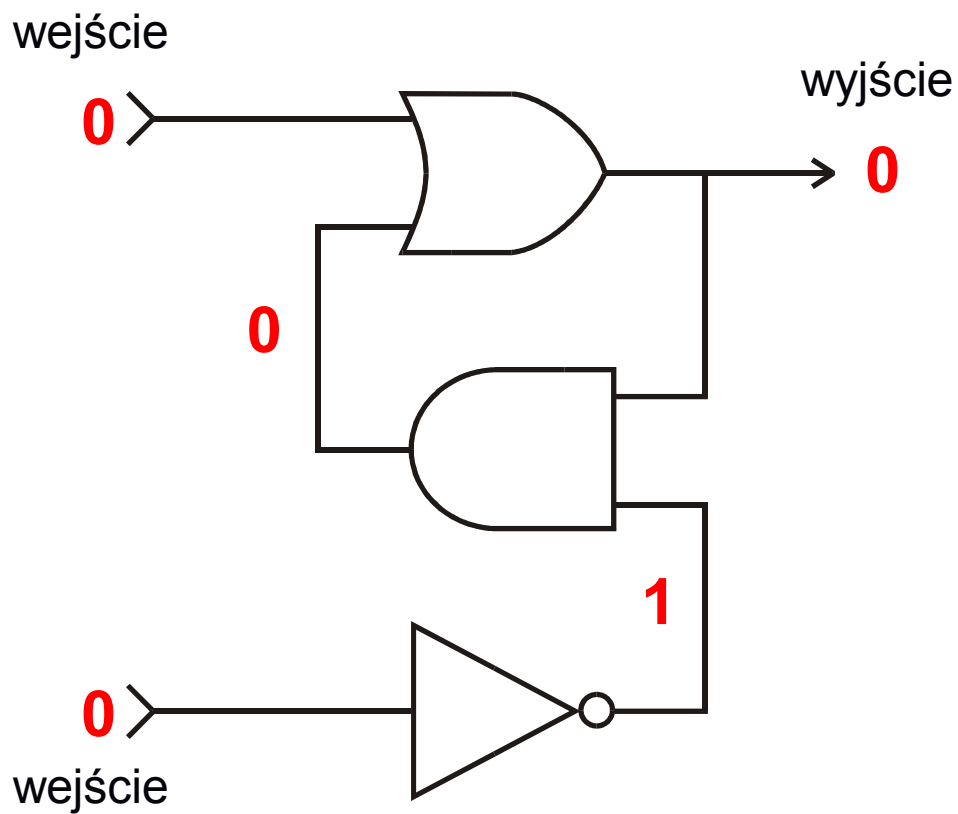
Bramki logiczne są podstawowymi elementami, z których buduje się komputery.

Zazwyczaj są układami elektronicznymi, w których cyfry 0 i 1 reprezentuje się przez różną poziomą napięcia.

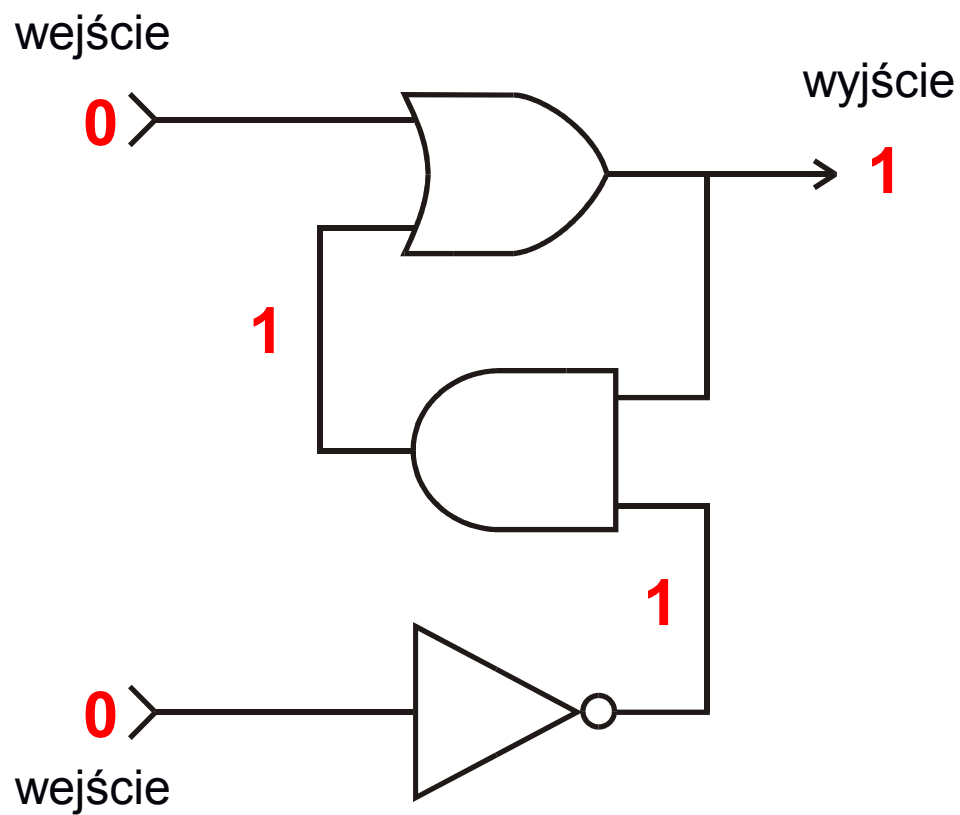
❖ **Przerzutniki** – układy idealne do zapamiętania jednego bitu.



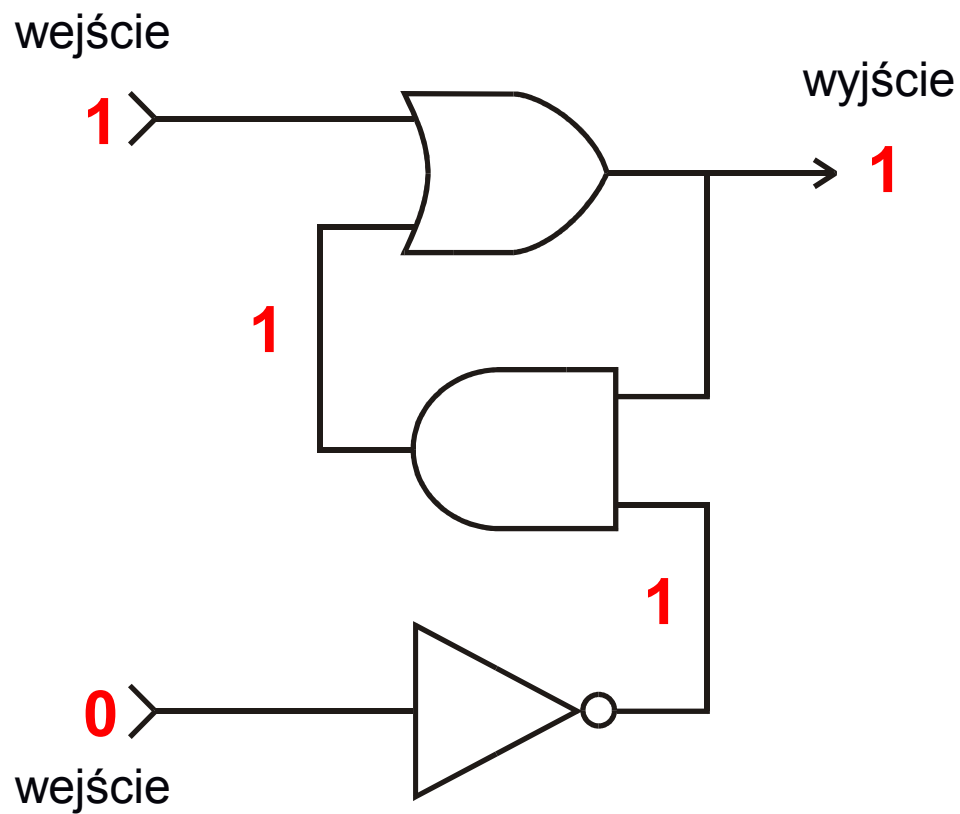
# Działanie przerzutnika



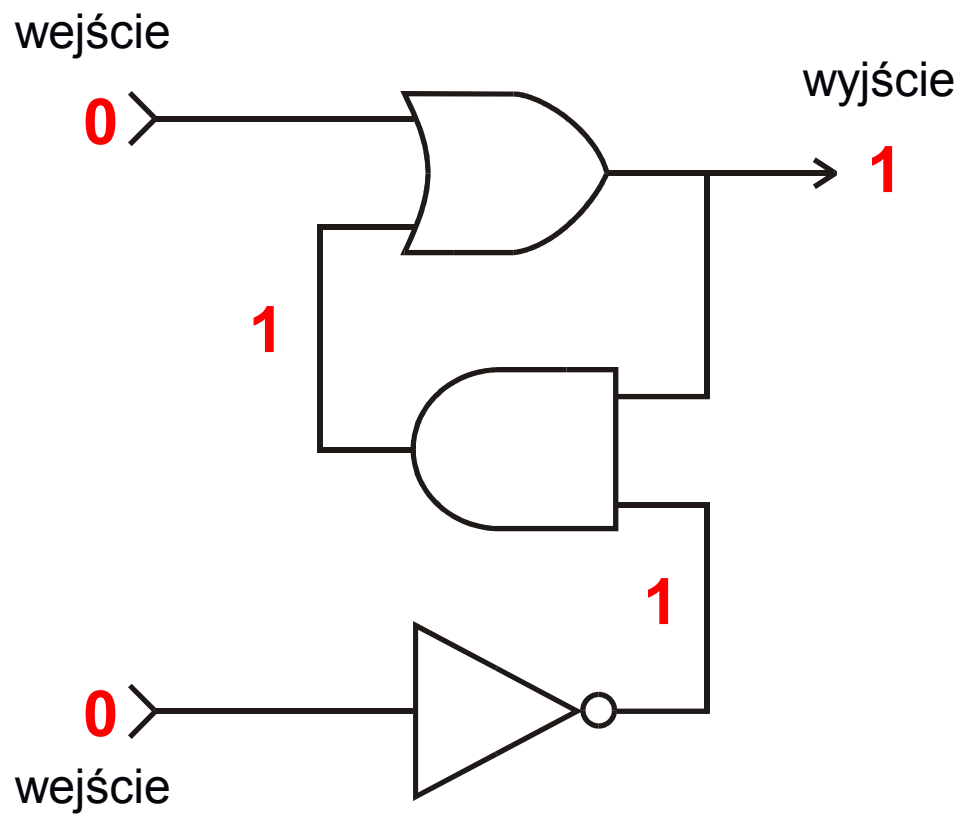
# Działanie przerzutnika



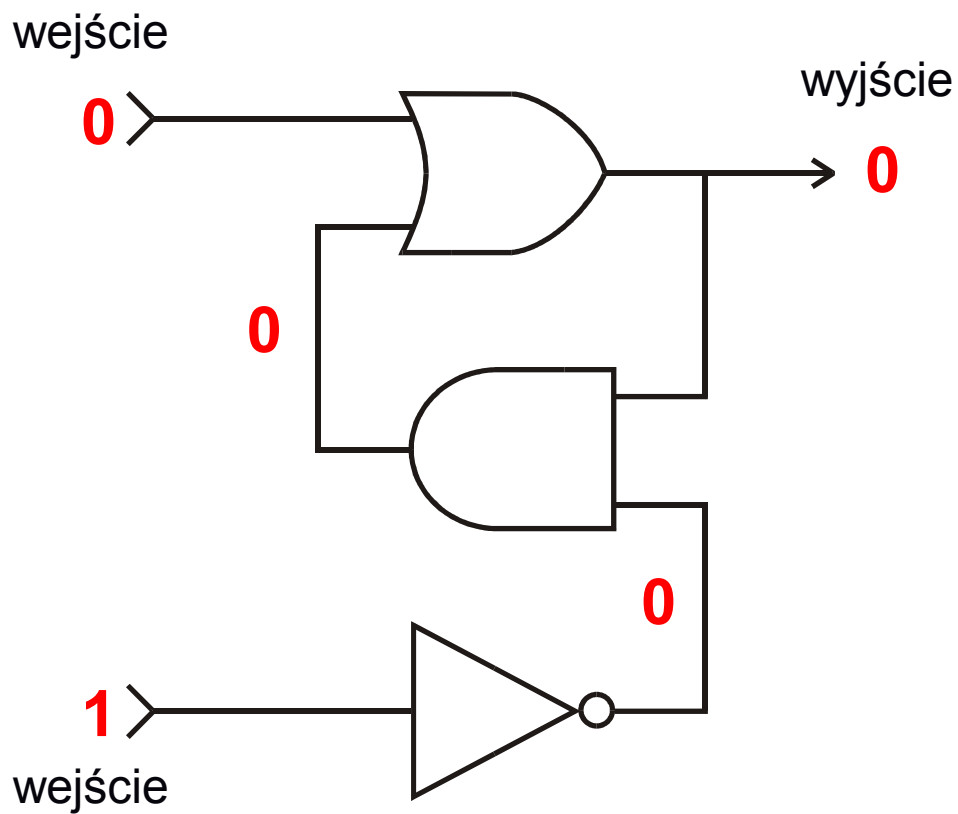
# Działanie przerzutnika



# Działanie przerzutnika



# Działanie przerzutnika



## Inne techniki przechowywania informacji

### ❖ Rdzenie – *core* – lata 60 ubiegłego stulecia;

małe pierścienie materiału magnetycznego z nawiniętym przewodem elektrycznym.

- kierunek namagnesowania oznaczał wartość przechowywanego bitu;
- ogromne rozmiary i duże zużycie energii;
- zdolność do przechowywania informacji po wyłączeniu komputera.

### ❖ Kondensatory

- wartość bitu reprezentowana jednym z dwóch możliwych stanów kondensatora – naładowany lub rozładowany.
- współczesna technika daje możliwość umieszczania na jednej płytce zwanej kością setek milionów maleńkich kondensatorów
- niewielkie ładunki mają skłonności do samoistnego zanikania dlatego też stosowany jest zabieg odświeżania ładunków – ***pamięć dynamiczna***.



# Reprezentacja informacji w komputerze

- ❖ Komputery przechowują informacje w postaci ciągu bitów.

10100100

111001101011011010100101

- ❖ **Notacja szesnastkowa** – *hexadecimal notation* – stosowana w celu uproszczenia reprezentacji ciągu bitów.

Ciąg bitów	reprezentacja szesnastkowa		
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

# Reprezentacja informacji w komputerze

W ten sposób można przedstawić :

- ciąg 8 bitowy 10100100 w postaci **A4**

- ciąg 16 bitowy 1110011010110110 w postaci **E6B6**

# Organizacja danych

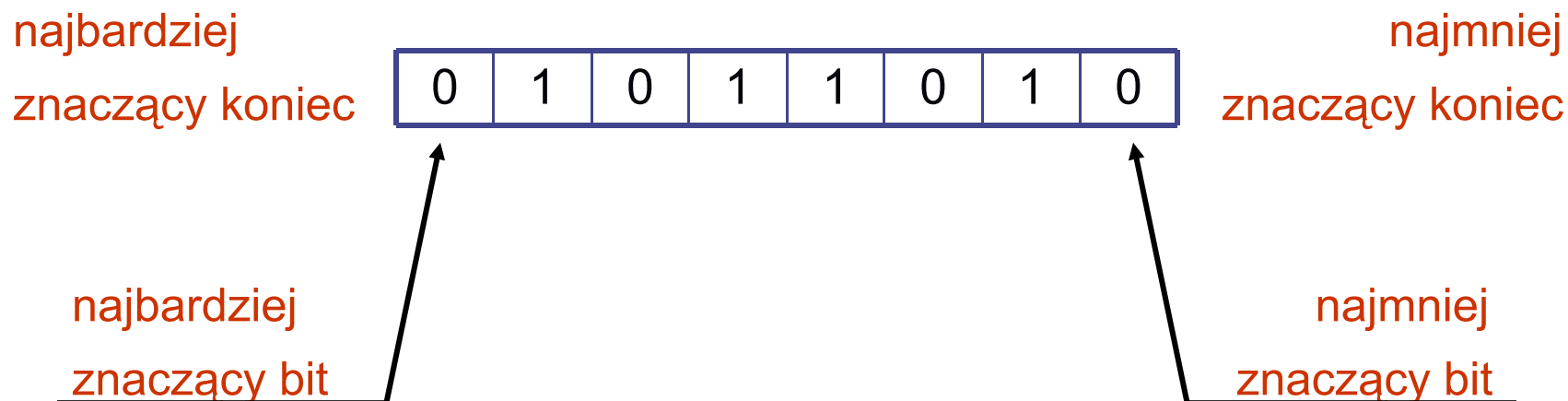
❖ **Pamięć główna** (operacyjna) – *main memory* – zestaw układów zdolnych do zapamiętania 1 bitu.

- Układy zorganizowane są w jednostki zwane **komórkami** lub **słowami**.
- Zazwyczaj w jednej komórce mieści się 8 bitów.
- Ciąg 8 bitów to **bajt**.
- Każda komórka posiada unikatowy adres.
- Dostęp do każdej komórki jest **niezależny** od dostępu do komórek pozostałych.



# Organizacja danych

❖ Bity umieszczone w komórce traktowane są w ustalony sposób.



☞ Ważną konsekwencją uporządkowania zarówno komórek w pamięci jak i bitów w komórce jest to, że cały zbiór bitów w pamięci stanowi jeden długi szereg.

## Jednostki wielkości pamięci

❖ 1 bit	1 b			
❖ 1 bajt	1 B	1 B = 8b		
❖ 1 kilobajt	1 kB	1 kB = 2 <sup>10</sup> B	(1 024 B )	~ 10 <sup>3</sup>
❖ 1 megabajt	1 MB	1 MB = 2 <sup>20</sup> B	(1 048 576 B)	~ 10 <sup>6</sup>
❖ 1 gigabajt	1 GB	1 GB = 2 <sup>30</sup> B	(1 073 741 824 B)	~ 10 <sup>9</sup>
❖ 1 terabajt	1 TB	1 TB = 2 <sup>40</sup> B	(11 099 511 627 776 B)	~ 10 <sup>12</sup>

☞ Jeden bajt może reprezentować 256 różnych wartości, które mogą odpowiadać **różnym** zapisywanym informacjom.

☞ **różnym** dzięki odpowiedniemu **kodowaniu informacji**.

## Kodowanie tekstu

- ❖ **ASCII** – *American Standard Code for Information Interchange* – wprowadzony przez *American National Standards Institute* (ANSI).
- ❖ Do reprezentacji małych i wielkich liter, znaków przestankowych, cyfr od 0 do 9 oraz informacji sterujących zastosowano ciągi 7-bitowe.
- ❖ Obecnie ASCII rozszerza się do 8 bitów wprowadzając 0 na najbardziej znaczącej pozycji ciągu 8-bitowego

0 – 31 – kody sterujące

32 – 47 – znaki pomocnicze ( ,!,+,-, ...)

48 – 57- cyfry 0 – 9

58 – 64 – znaki pomocnicze (:,;,,>,,=,?,@)

65 – 90 – A, B, ... , Z

91 – 96 – znaki specjalne

97 – 122 – a, b, ... , z

123 – 126 – znaki specjalne

127 – DEL

## Kodowanie liczb

### ❖ System pozycyjny wagowy

$$L = \sum_{i=-m}^{n-1} a_i p^i =$$

$$= a_{-m} p^{-m} + \dots + a_{-1} p^{-1} + a_0 p^0 + a_1 p^1 + \dots + a_{n-1} p^{n-1}$$

gdzie:

$p \geq 2$  - podstawa systemu liczbowego

$a_i$  - cyfra  $i$ -tej pozycji,  $a_i \in \{0, 1, \dots, p-1\}$

$n$  - ilość cyfr części całkowitej liczby  $L$

$m$  - ilość cyfr części ułamkowej liczby  $L$

## Kodowanie liczb

### ❖ System dziesiętny

$$p = 10, \quad a_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

- dla liczby 1204,42

$$L = \sum_{i=-m}^{n-1} a_i p^i$$

można zapisać w postaci

$$\begin{aligned} L &= 2 \cdot 10^{-2} + 4 \cdot 10^{-1} + 4 \cdot 10^0 + 0 \cdot 10^1 + 2 \cdot 10^2 + 1 \cdot 10^3 = \\ &= 0,02 + 0,4 + 4 + 0 + 200 + 1000 = 1204,42 \end{aligned}$$



## Kodowanie liczb

### ❖ System piątkowy

$$p = 5, \quad a_i \in \{0, 1, 2, 3, 4\}$$

liczba  $(1204,42)_5$

$$L = \sum_{i=-m}^{n-1} a_i p^i$$

przedstawia wartość

$$\begin{aligned} L &= 2 \cdot 5^{-2} + 4 \cdot 5^{-1} + 4 \cdot 5^0 + 0 \cdot 5^1 + 2 \cdot 5^2 + 1 \cdot 5^3 = \\ &= 0,08 + 0,8 + 4 + 0 + 50 + 125 = 179,88 \end{aligned}$$

## Kodowanie liczb

### ❖ System dwójkowy - binarny

$$p = 2, \quad a_i \in \{0, 1\}$$

liczba (całkowita) reprezentowana przez ciąg bitów : 10100101

zgodnie z zależnością : 
$$L = \sum_{i=-m}^{n-1} a_i p^i$$

przedstawia wartość :

$$\begin{aligned} L &= 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 + 0 \cdot 2^4 + 1 \cdot 2^5 + 0 \cdot 2^6 + 1 \cdot 2^7 = \\ &= 1 + 0 + 4 + 0 + 0 + 32 + 0 + 128 = 165 \end{aligned}$$

## Ułamki w zapisie binarnym

❖ **Liczba ułamkowa** wyrażona w zapisie binarnym jako :

1 1 1 0 0 · 1 0 1 1

zgodnie z zależnością :

$$L = \sum_{i=-m}^{n-1} a_i p^i$$

przedstawia liczbę :

$$L = 1 \cdot 2^{-4} + 1 \cdot 2^{-3} + 0 \cdot 2^{-2} + 1 \cdot 2^{-1} + 0 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4 =$$

$$= 0,0625 + 0,125 + 0 + 0,5 + 0 + 0 + 4 + 8 + 16 = 28,6875$$

☞ kropka pozycyjna ( *radix point* ) oddziela część całkowitą liczby od części ułamkowej.

# Konwersja liczby dziesiętnej

- ❖ Algorytm zamiany całkowitej liczby dziesiętnej  $L$  na liczbę w systemie o podstawie  $p$ .

$$\begin{array}{rccccccc}
 L & : & p & = & L_1 & \text{reszta} & R_1 \\
 L_1 & : & p & = & L_2 & \text{reszta} & R_2 \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 L_{k-1} & : & p & = & 0 & \text{reszta} & R_k
 \end{array}$$

$$(L)_p = R_k R_{k-1} \dots R_1$$

## Konwersja liczby dziesiętnej

Przykład zamiany liczby 333 na liczbę w systemie piątkowym .

$$\begin{array}{rclclcl}
 333 & : & 5 & = & 66 & \text{reszta} & 3 \\
 66 & : & 5 & = & 13 & \text{reszta} & 1 \\
 13 & : & 5 & = & 2 & \text{reszta} & 3 \\
 2 & : & 5 & = & 0 & \text{reszta} & 2
 \end{array}$$

$$333 = (2313)_5$$

## Konwersja liczby dziesiętnej

Przykład zamiany liczby 333 na liczbę w systemie binarnym.

333	:	2	=	166	reszta	1
166	:	2	=	83	reszta	0
83	:	2	=	41	reszta	1
41	:	2	=	20	reszta	1
20	:	2	=	10	reszta	0
10	:	2	=	5	reszta	0
5	:	2	=	2	reszta	1
2	:	2	=	1	reszta	0
1	:	2	=	0	reszta	1

$$333 = ( 101001101 )_2$$

# Konwersja ułamka

- ❖ Algorytm zamiany ułamka dziesiętnego  $U$  na ułamek w systemie o podstawie  $p$ .

$$\begin{array}{rccccccc}
 U & * & p & = & L_1 & \text{cz. całkowita} & C_1 \\
 U_1 & * & p & = & L_2 & \text{cz. całkowita} & C_2 \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 U_{k-1} & * & p & = & 0 & \text{cz. całkowita} & C_k
 \end{array}$$

$$(U)_p = C_1 \dots C_{k-1} C_k$$

## Konwersja ułamka

Przykład zamiany ułamka 0,3104 na ułamek w systemie piątkowym .

0,3104	*	5	=	1,552	cz. całkowita	1
0,552	*	5	=	2,76	cz. całkowita	2
0,76	*	5	=	3,8	cz. całkowita	3
0,8	*	5	=	4,0	cz. całkowita	4

$$0,3104 = ( 0,1234 )_5$$



## Konwersja ułamka

Ale w przypadku ułamka 0,25 ...

0,25	*	5	=	1,25	cz. całkowita	1
0,25	*	5	=	1,25	cz. całkowita	1
0,25	*	5	=	1,25	cz. całkowita	1
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
0,25	*	5	=	1,25	cz. całkowita	1

$$0,25 = ( 0,111111(1) )_5$$