

# Architektura systemów komputerowych

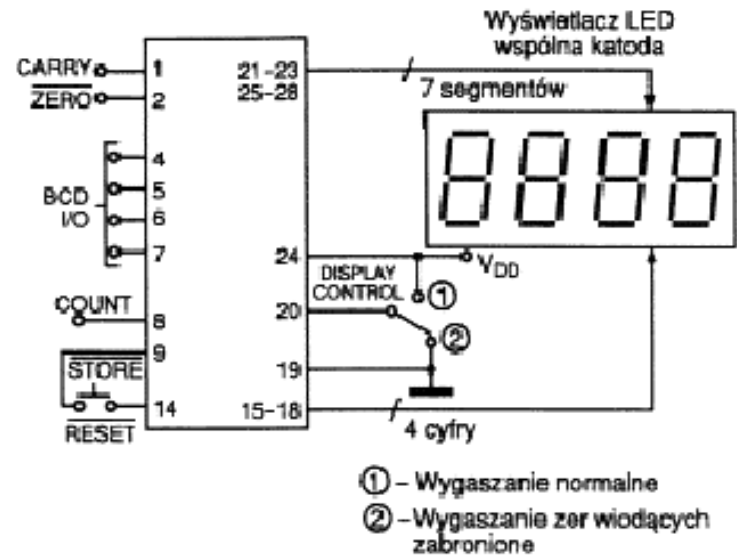
**Wykład I**

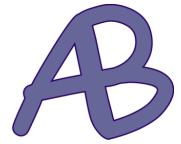
**Podstawowe pojęcia**

*dr Artur Bartoszewski*

# Część I

## Informacja cyfrowa

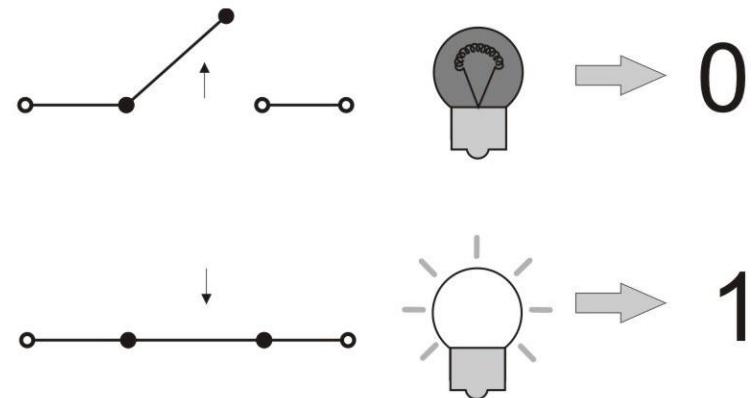


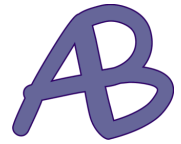


## Dlaczego system binarny?

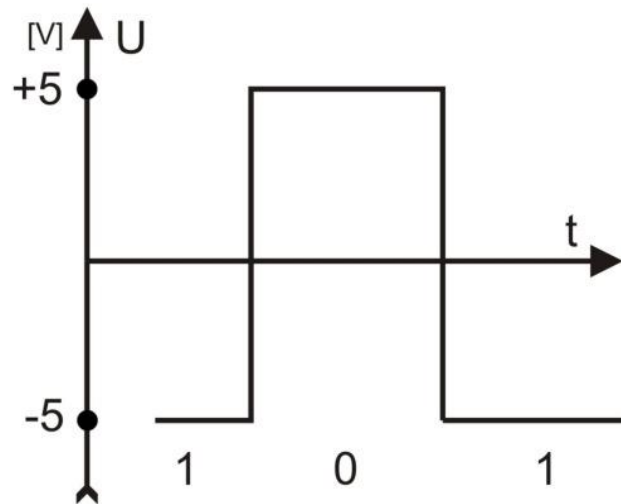
Dwójkowy system liczenia, choć nieintuicyjny dla człowieka, ma trzy cechy czyniące go idealnym z punktu widzenia elektroniki cyfrowej i informatyki. Są nimi:

- łatwość implementacji elektrycznej i elektronicznej,
- odporność na zakłócenia,
- możliwość interpretacji wartości 0 i 1 jako wartości logicznych „prawda” i „fałsz” (algebra Boole’a).





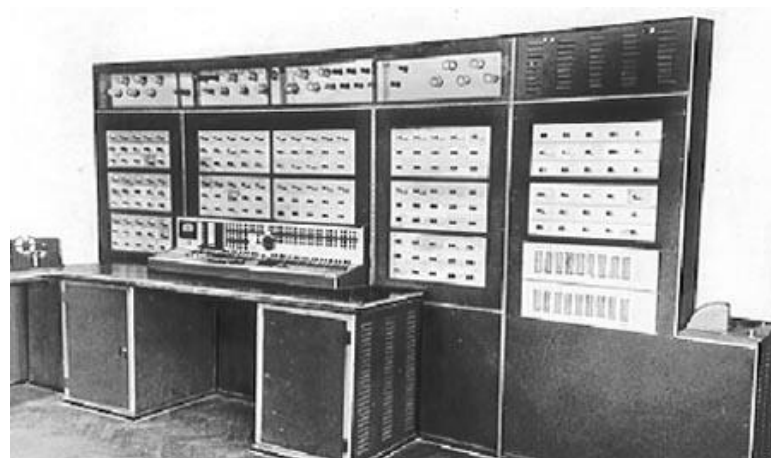
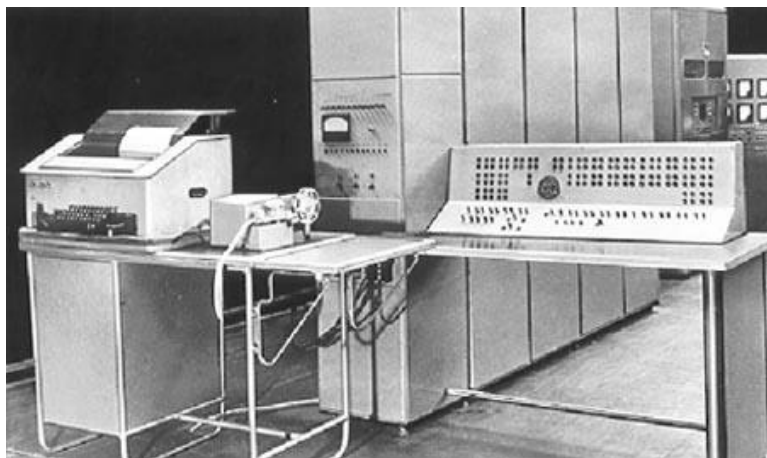
- ✓ Współczesne urządzenia techniki cyfrowej pracują na nieco innej zasadzie.
- ✓ Zero i jedynka reprezentowane są jako dwie różne wartości napięcia elektrycznego.
- ✓ Najczęściej są to wartości napięcia o przeciwnych znakach. Np. napięcie +5 woltów oznaczać może cyfrę zero, a -5 woltów jedynkę.



## CIEKAWOSTKA

Jedynym komputerem pracującym w systemie trójkowym był eksperymentalny radziecki Setun (rok 1959).

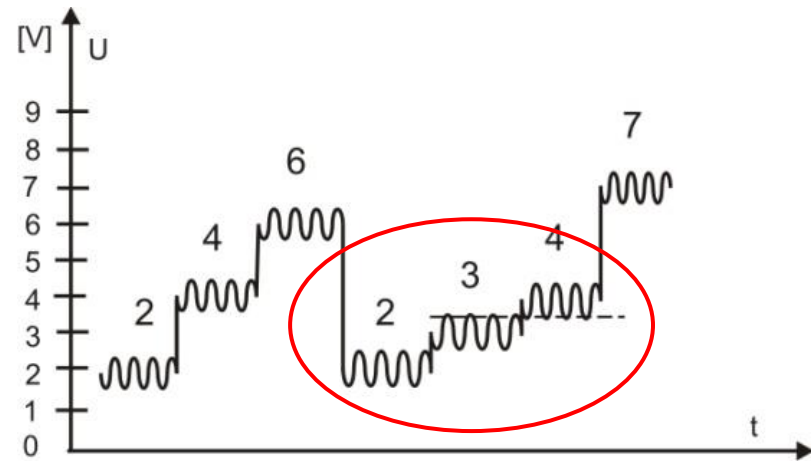
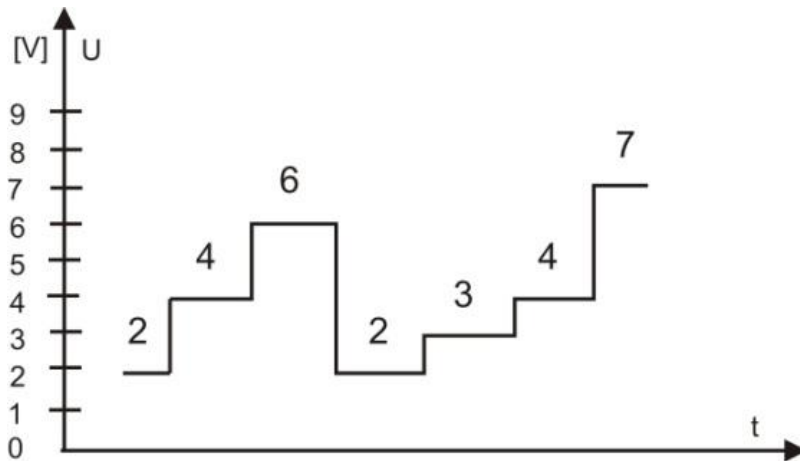
Elementem przechowującym informacje była tu para rdzeni magnetycznych. Każdy z nich mógł być namagnesowany w jednym z dwóch kierunków. Dawało to w sumie cztery możliwe stany, lecz wykorzystane były tylko trzy z nich (cyfry  $\{0, 1, 2\}$ ). Czwarty stan pełnił funkcję kontrolną.



Wyprodukowano 50 takich komputerów, z czego 30 działało na uniwersytetach w całym ZSRR

# System binarny – odporność na zakłócenia

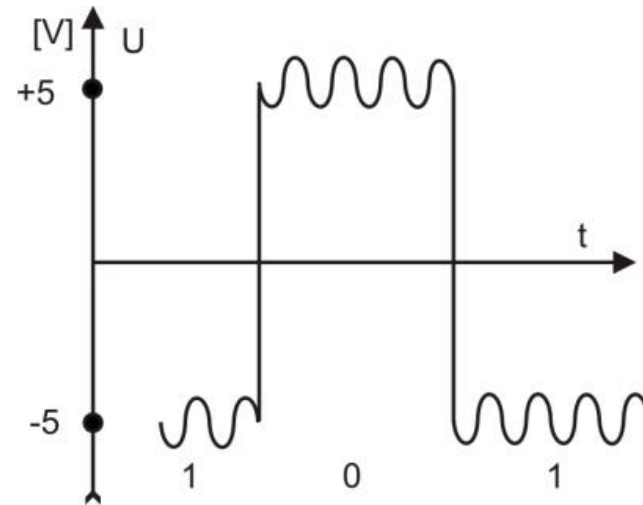
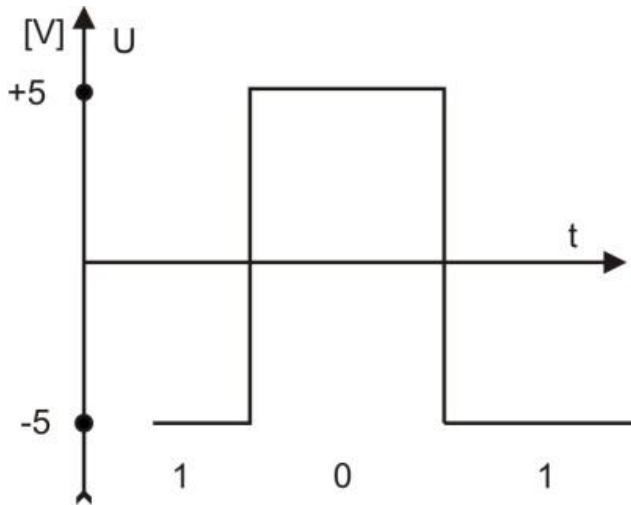
## Odporność na zakłócenia



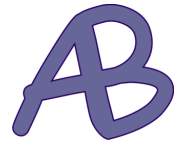
- ✓ Na rysunku widzimy cyfrowy **sygnał dziesiętny** – może on przyjmować 10 wartości odpowiadających cyfrom 0-9
- ✓ Wartość „3” może być zinterpretowana jako „2” lub „4” – zależnie od tego, w której dokładnie chwili nastąpi pomiar napięcia.

# System binarny – odporność na zakłócenia

## Odporność na zakłócenia



- ✓ Dla porównania rysunek przedstawia sygnał binarny o takiej samej rozpiętości (10V), na który nałożył się sygnał zakłócający o amplitudzie jednego wolta. Widzimy, że pomimo zakłóceń nie ma ryzyka błędnego zinterpretowania transmisji.
- ✓ Aby wprowadzić przekłamania do transmisji binarnej, sygnał zakłócający musiałby sięgnąć połowy różnicy napięć pomiędzy stanem „0” i „1”, czyli w omawianym tu przykładzie, pięciu wolt.

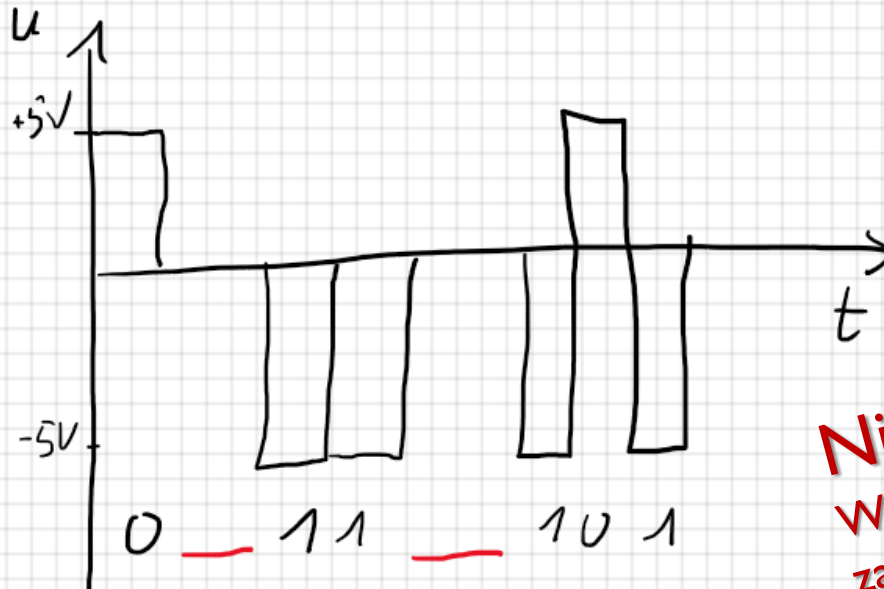


- ✓ Wewnątrz komputera informacja ma postać fizycznych sygnałów dwuwartościowych (np. dwa poziomy napięcia, dwa kierunki namagnesowania, dwie fazy przebiegu okresowego), którym można przypisać np. symbole 0 i 1.
- ✓ Każdy komunikat musi być przedstawiony jako łańcuch zerojedynkowy określonej długości (słowo).





0,3,5 -> 0 11 101



Przykład:

chcemy za pomocą sygnałów elektrycznych przesłać ciąg liczb {0, 3, 5}

**Nie działa**  
W technologii komputerowej nie możemy zakodować spacji – inaczej wyjdzie nam komputer „trójkowy”

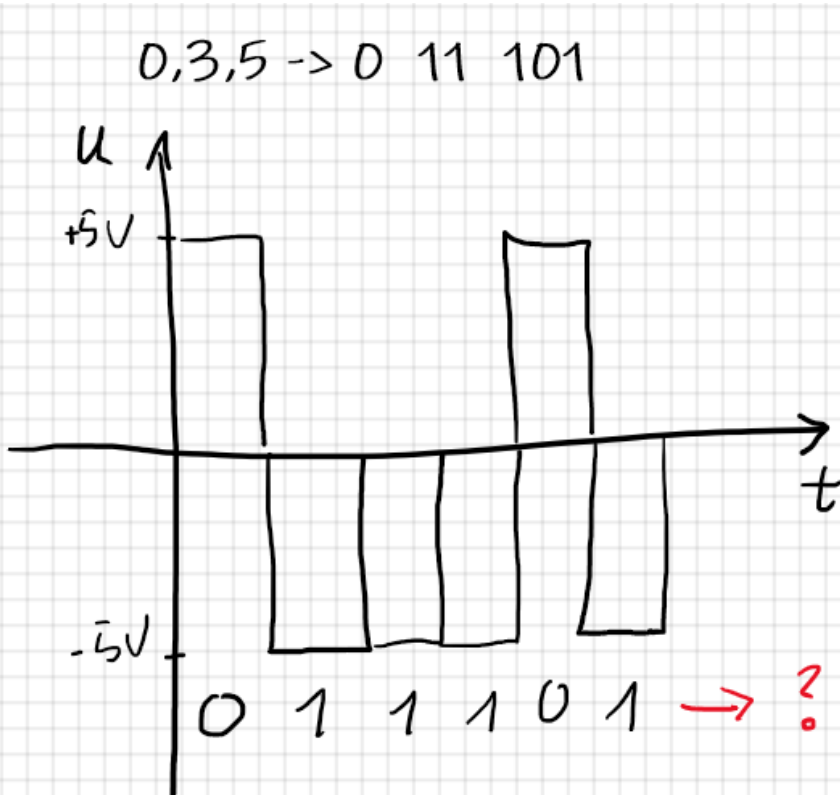
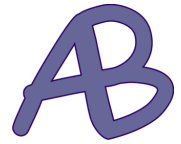
W tym zapisie występują 3 cyfry:

0 - zero

1 - jeden

- "odstęp", "spacja"

# Cyfrowe dane

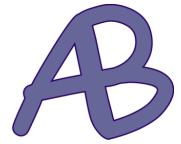


Przykład:

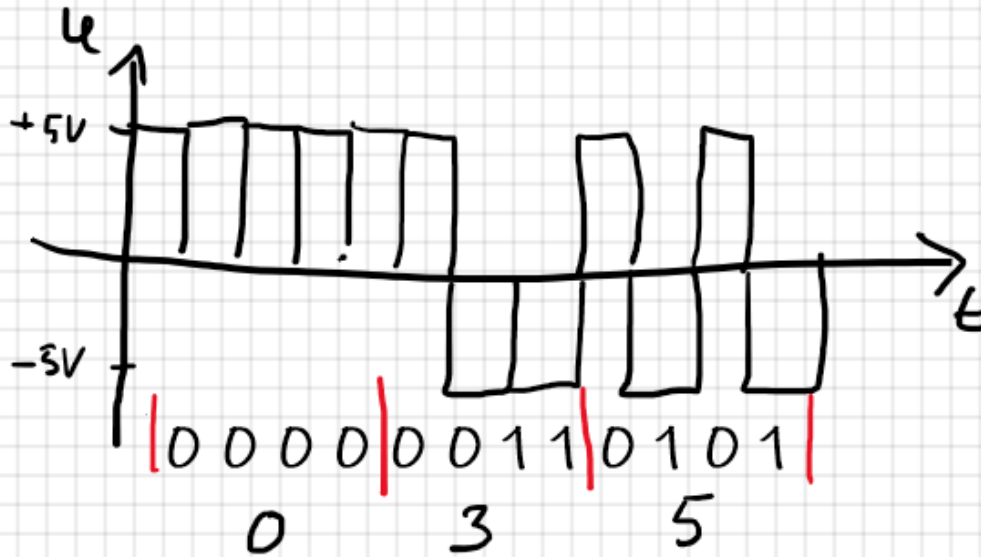
chcemy za pomocą sygnałów elektrycznych przesłać ciąg liczb {0, 3, 5}

**Nie działa**  
nie da się odczytać wyniku

Otrzymanego ciągu bitów nie da się jednoznacznie zinterpretować



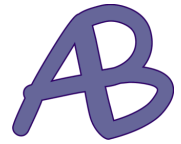
0,3,5  $\rightarrow$  0 11 101  $\rightarrow$  0000 0011 0101



Przykład:  
chcemy za pomocą  
sygnałów elektrycznych  
przesłać ciąg liczb {0,3,5}

OK

Każdą liczbę zapisujemy na określonej liczbie bitów (słowie binarnym)



**Definicja:**

**Informacją cyfrową** nazywamy informację przedstawioną (zakodowaną) w postaci słów cyfrowych.

**Definicja:**

**Słowem cyfrowym** (binarnym) nazywamy dowolny ciąg o **ściśle określonej długości** składający się z symboli 0 i/lub 1

# Słowo binarne



- ✓ W technice cyfrowej nie posługujemy się naturalnym kodem binarnym. Najmniejszą porcją informacji która może być przesłana, zapamiętana czy też przetworzona jest **SŁOWO**
- ✓ Rozmiar słowa zależy od architektury systemu.





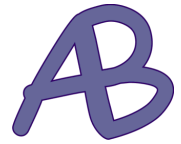
## Przyczyny zastosowania systemu binarnego

---

kb	Mb	Gb	Tb
kilobit	megabit	gigabit	terabit

**1 bajt = 8 bitów** (*ang. byte*)

kB	MB	GB	TB
kilobajt	megabajt	gigabajt	terabajt



## Ważniejsze potęgi dwójki

$$2^0 = 1$$

$$2^1 = 2$$

$$2^2 = 4$$

$$2^3 = 8$$

$$2^4 = 16$$

$$2^5 = 32$$

$$2^6 = 64$$

$$2^7 = 128$$

$$2^8 = 256 = 1 \text{ bajt}$$

$$2^{16} = 65.536$$

$$2^{24} = 16.777.216$$

**1 bajt = 8 bitów**

**10101111**

$$2^{10} \text{ bajtów} = 1 \text{ kB} \quad (1024)$$

$$2^{20} \text{ bajtów} = 1 \text{ MB} \quad (1024 * 1024)$$

$$2^{30} \text{ bajtów} = 1 \text{ GB} \quad (1024 * 1024 * 1024)$$



## System szesnastkowy (hexadecymalny)

---

Duże liczby binarne są nieczytelne.

0101001010010010000111100101010010101010110

Celem wprowadzenia systemu szesnastkowego jest skrócenie zapisu bez przeliczania na system dziesiętny.

Każde 4 bity da się przedstawić za pomocą 1 cyfry szesnastkowej – bez żadnego przeliczania.

hex	bin	dec
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15





# System szesnastkowy (hexadecymalny)

Przykład:

01010010100100100001111001010100101010101100

0101 0010 1001 0010 0001 1110 0101 0100 1010  
1010 1100

0101 0010 1001 0010 0001 1110 0101 0100 1010  
5 2 8 2 1 E 5 4 A  
1010 1100  
A C

52821E54AAC

hex	bin	dec
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

## Liczby zmiennopozycyjne



5 973 600 000 000 000 000 000 000 kg  
(Masa Ziemi)

$$5,9736 \cdot 10^{24}$$

$$5,9736 \text{ E}+24$$

**mantysa (precyzja)**

**cecha (wykładnik)**

Notacja naukowa pozwala na kodowanie  
bardzo dużych / małych liczb

## Liczby zmiennopozycyjne



5,625<sub>10</sub>

101,101

0,101101\*2<sup>3</sup>

Mantysa znormalizowana dla liczb binarnych należy do przedziału  $<\frac{1}{2}, 1)$ .

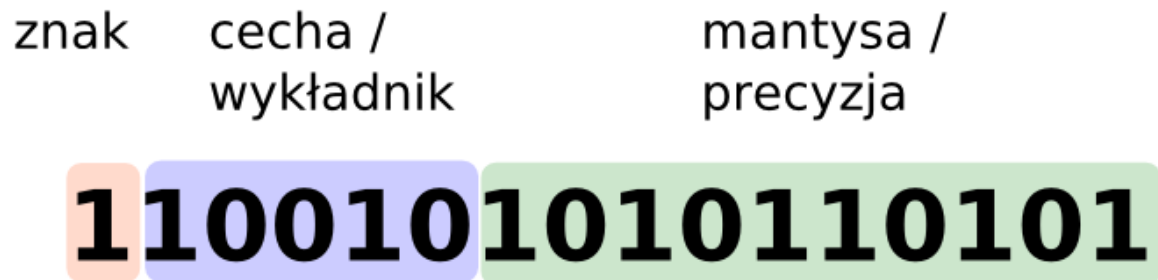
W praktyce oznacza to, że przecinek należy ustawić w taki sposób, aby liczba miała postać:

**0,1xxxxxx...**

Dzięki normalizacji zapis staje się jednoznaczny.

## Liczby zmiennopozycyjne

---



Standard IEEE 754

pojedyncza precyzja	1	8	23	(32 bity)
---------------------	---	---	----	-----------

podwójna precyzja	1	11	52	(64 bity)
-------------------	---	----	----	-----------

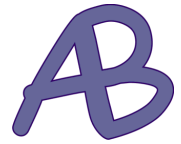


Przykład

**1000110110110101**

**1** **00011** **0110110101**

$$\begin{aligned} & - 0,110\ 110101 * 2^3 = \\ & = 110,110101 = 6,828125_{10} \end{aligned}$$



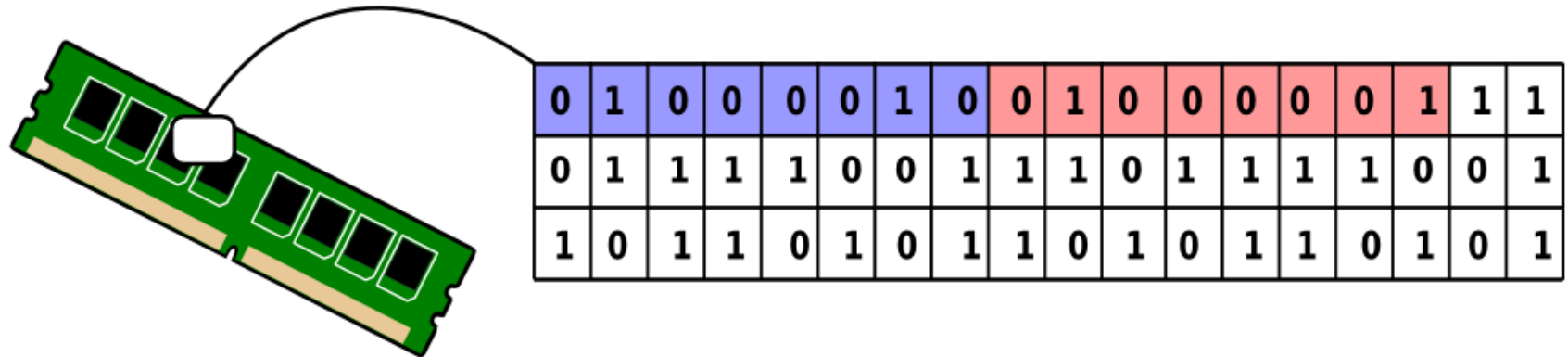
# Kod ASCII

American Standard Code for Information Interchange

Kod przypisujący 7-bitowe (128 kombinacji) ciągi do znaków.

A 01000001  
B 01000010

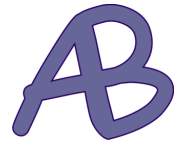
7 bitów



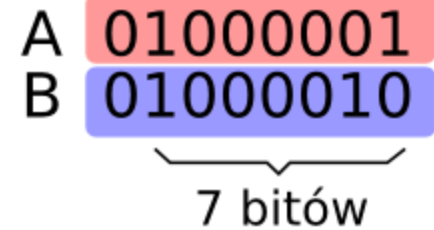


# Kod ASCII

1	-	⊙	30	-	▲	54	-	6	78	-	N	102	-	f	126	-	˜	150	-	ü	174	-	«	198	-	⌠	222	-	┘
2	-	⊗	31	-	▼	55	-	7	79	-	O	103	-	g	127	-	Δ	151	-	û	175	-	»	199	-	⌡	223	-	└
3	-	♥	32	-	!	56	-	8	80	-	P	104	-	h	128	-	Ç	152	-	Ÿ	176	-	⋮	200	-	⌢	224	-	α
4	-	♦	33	-	!"	57	-	9	81	-	Q	105	-	i	129	-	Û	153	-	ö	177	-	⋱	201	-	⌣	225	-	β
5	-	♣	34	-	"	58	-	:	82	-	R	106	-	j	130	-	ë	154	-	ü	178	-	⋰	202	-	⌤	226	-	Γ
6	-	♠	35	-	#	59	-	:	83	-	S	107	-	k	131	-	ä	155	-	ç	179	-	⋱	203	-	⌥	227	-	Π
11	-	♂	36	-	\$	60	-	<	84	-	T	108	-	l	132	-	ä	156	-	ê	180	-	⋱	204	-	⌦	228	-	Σ
12	-	♀	37	-	%	61	-	=	85	-	U	109	-	m	133	-	ä	157	-	¥	181	-	⋱	205	-	⌧	229	-	σ
14	-	♫	38	-	&	62	-	>	86	-	V	110	-	n	134	-	q	158	-	₯	182	-	⋱	206	-	⌨	230	-	μ
15	-	☼	39	-	'	63	-	?	87	-	W	111	-	o	135	-	ç	159	-	f	183	-	⋱	207	-	〈	231	-	γ
16	-	▶	40	-	(	64	-	@	88	-	X	112	-	p	136	-	ë	160	-	ž	184	-	⋱	208	-	〉	232	-	ÿ
17	-	◀	41	-	)	65	-	A	89	-	Y	113	-	q	137	-	ë	161	-	ž	185	-	⋱	209	-	⌫	233	-	Ÿ
18	-	↕	42	-	*	66	-	B	90	-	Z	114	-	r	138	-	ë	162	-	ó	186	-	⋱	210	-	⌬	234	-	Ź
19	-	!!!	43	-	+	67	-	C	91	-	[	115	-	s	139	-	ï	163	-	ó	187	-	⋱	211	-	⌭	235	-	δ
20	-	¶	44	-	,	68	-	D	92	-	\	116	-	t	140	-	î	164	-	ñ	188	-	⋱	212	-	⌮	236	-	∞
21	-	§	45	-	-	69	-	E	93	-	]	117	-	u	141	-	ć	165	-	Ń	189	-	⋱	213	-	⌯	237	-	ø
22	-	■	46	-	.	70	-	F	94	-	^	118	-	v	142	-	Ä	166	-	ž	190	-	⋱	214	-	⌰	238	-	Ε
23	-	⊕	47	-	/	71	-	G	95	-	⌘	119	-	w	143	-	Ä	167	-	ž	191	-	⋱	215	-	⌱	239	-	Π
24	-	↑	48	-	0	72	-	H	96	-	⌘	120	-	x	144	-	Ę	168	-	č	192	-	⋱	216	-	⌲	240	-	≡
25	-	↓	49	-	1	73	-	I	97	-	a	121	-	y	145	-	ę	169	-	ŕ	193	-	⋱	217	-	⌳	241	-	±
26	-	+	50	-	2	74	-	J	98	-	b	122	-	z	146	-	İ	170	-	ŕ	194	-	⋱	218	-	⌴	242	-	≥
27	-	+	51	-	3	75	-	K	99	-	c	123	-	{	147	-	ô	171	-	½	195	-	⋱	219	-	⌵	243	-	≤
28	-	L	52	-	4	76	-	L	100	-	d	124	-		148	-	ö	172	-	½	196	-	⋱	220	-	⌶	244	-	∫
29	-	+	53	-	5	77	-	M	101	-	e	125	-	}	149	-	Ć	173	-	i	197	-	⋱	221	-	⌷	245	-	.



## Regionalne strony kodowe



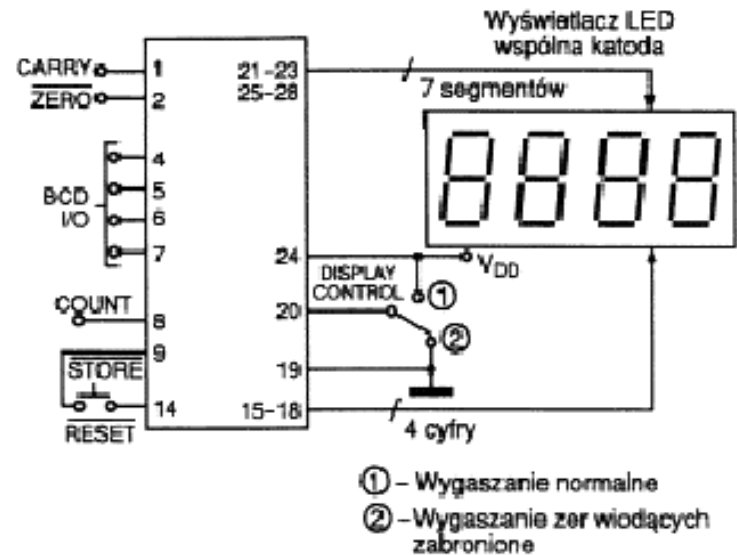
128 kombinacji wystarcza do zakodowania wszystkich liter i cyfr oraz kilkudziesięciu znaków drukowalnych (+ - =...) i niedrukowalnych znaków sterujących (np. nowy wiersz).

Rozbudowanie kodu do 8 bitów pozwala na przypisanie znaków narodowych (ąęäö...). Przykładowo Europa Centralna używa dla swoich alfabetów rozszerzenia iso-8859-2, a Europa Zachodnia iso-8895-1.



## Część 2

# Jak opisujemy budowę komputera – poziomy abstrakcji





## Jak opisujemy komputer?

---

Przy opisie komputera przyjmuje się zwykle trzy poziomy abstrakcji :

- *architektura,*
- *implementacja;*
- *realizacja.*

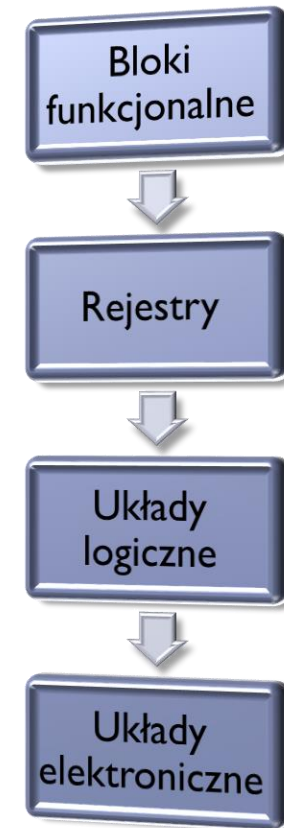
- ✓ Te same poziomy występują przy projektowaniu;
- ✓ wprawdzie są one wzajemnie zależne, lecz realizowane przez osobne zespoły i wymagające od twórców innej wiedzy, stosujące inne narzędzia i inaczej dokumentowane

## Opis komputera – poziomy abstrakcji

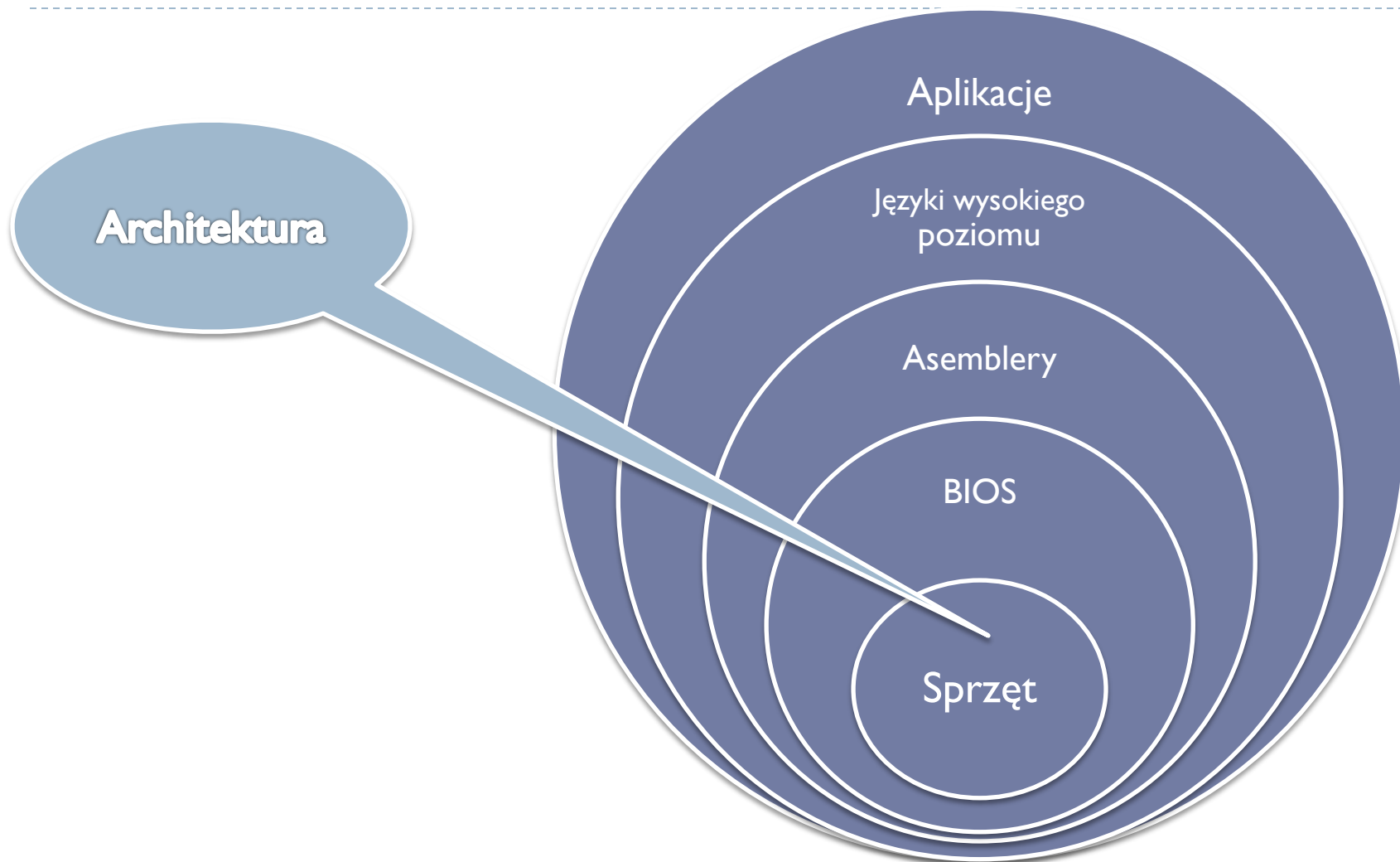
---

Poglądowo można powiedzieć, że:

- **architektura** opisuje zachowanie się z punktu widzenia użytkownika.
- **implementacja** - jaka jest jej *struktura* i *jak działa*,
- **realizacja** określa z czego maszyna jest zbudowana



# Opis komputera – poziomy abstrakcji

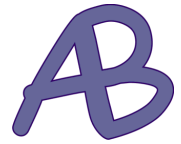




## Opis komputera – poziomy abstrakcji

---

- ✓ Z punktu widzenia użytkownika, komputer jest postrzegany poprzez język, w jakim się z nim komunikuje - jako pewna **maszyna wirtualna**.
- ✓ Może być nieistotne, jaka ta maszyna „jest naprawdę” - jaki jest procesor, jaki jest system operacyjny, jakie jest inne oprogramowanie.



## Standaryzacja technologii komputerowych

---

Technologia komputerowa jest rozwijana przez tysiące niezależnych i konkurujących ze sobą firm. Stanowi jednak spójną całość – urządzenia różnych producentów są ze sobą kompatybilne.

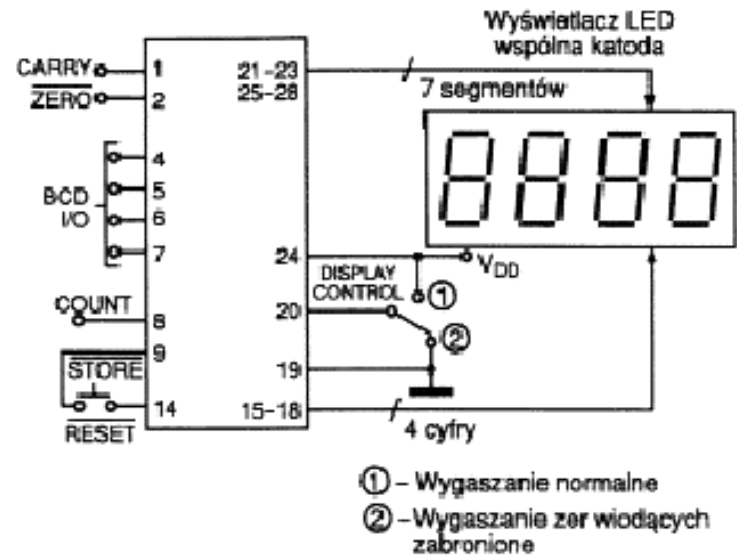
Ponieważ nie jest możliwe „centralne sterowanie” jej rozwojem, wypracowano metody negocjowania standardów.

**IEEE** (ang. Institute of Electrical and Electronics Engineers - **Instytut Inżynierów Elektryków i Elektroników**,) – organizacja typu non-profit skupiająca profesjonalistów. Powstała z konsolidacji grup AIEE oraz IRE w 1963 roku. Jednym z podstawowych jej zadań jest ustalanie standardów konstrukcji, pomiarów itp. dla urządzeń elektronicznych, w tym standardów dla urządzeń i formatów komputerowych.

**ISO Międzynarodowa Organizacja Normalizacyjna** (ang. International Organization for Standardization) – organizacja pozarządowa zrzeszająca krajowe organizacje normalizacyjne.

## Część 3

# Pojęcie komputera



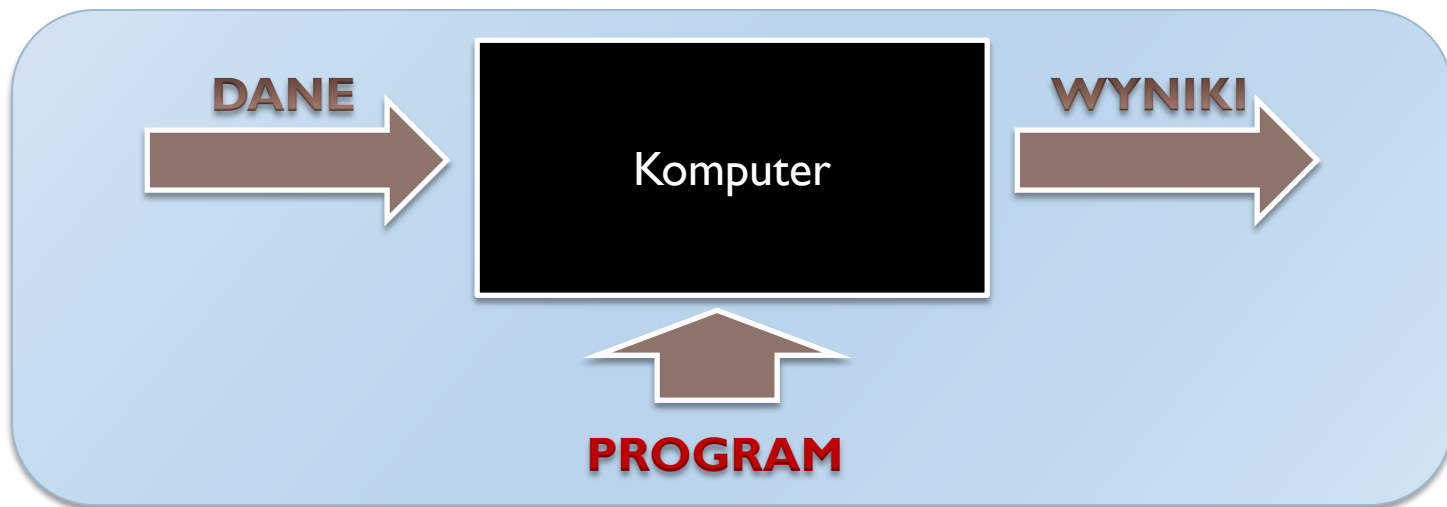
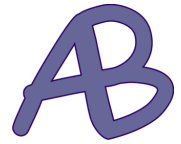


- **Układy cyfrowe** służą do przetwarzania informacji.
  - Do układu podajemy dane.
  - Układ wykonuje na danych określone operacje.
  - Układ zwraca wyniki.
- ✓ **Układ cyfrowy** to nie to samo co procesor czy nawet ALU.
- ✓ Pojęcie Układu cyfrowego obejmuje każde urządzenie elektroniki i automatyki spełniające powyższe funkcje.





- ✓ Każdy komputer porównać można do **czarnej skrzynki**.
- ✓ W tym modelu widać, że zarówno „surowcem”, jak i „produktem” komputera jest informacja, przy czym - ze względu na olbrzymią różnorodność jej postaci - musi być ona przekształcana: na wejściu do postaci wymaganej przez maszynę, a na wyjściu do postaci czytelnej dla użytkownika



## Model von Neumana

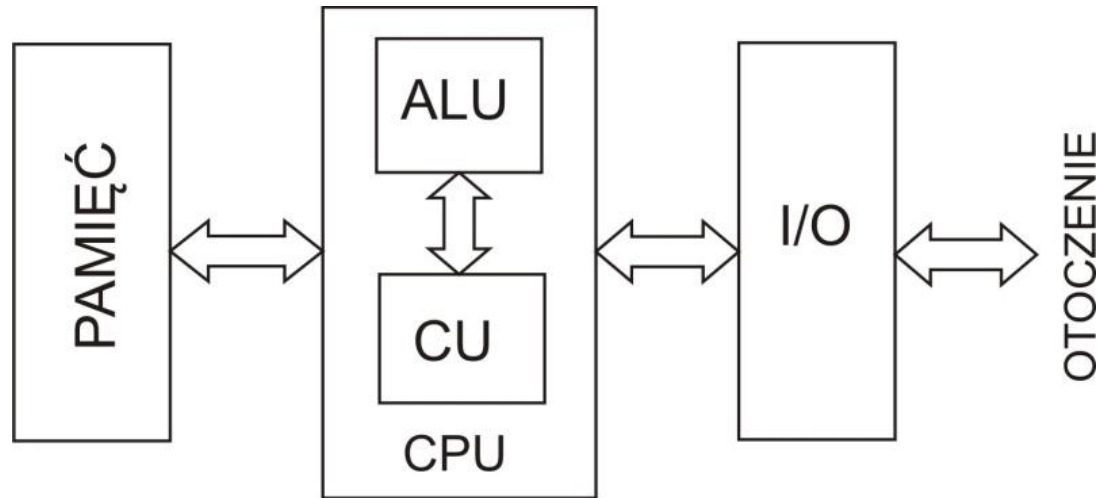
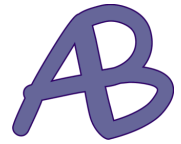
---

Podstawowym modelem prezentującym strukturę blokową każdego komputera, jest tzw. **model von Neumanna**. Został on zaproponowany w roku 1945 przez jednego z pionierów informatyki Johna von Neumanna.

W modelu tym wyróżniamy trzy podstawowe części:

- **procesor** - zawierający część sterującą (CU) oraz część arytmetyczno-logiczną (ALU),
- **pamięć** - przechowuje dane i program,
- **układy wejścia-wyjścia** - zapewniające komunikację z otoczeniem.

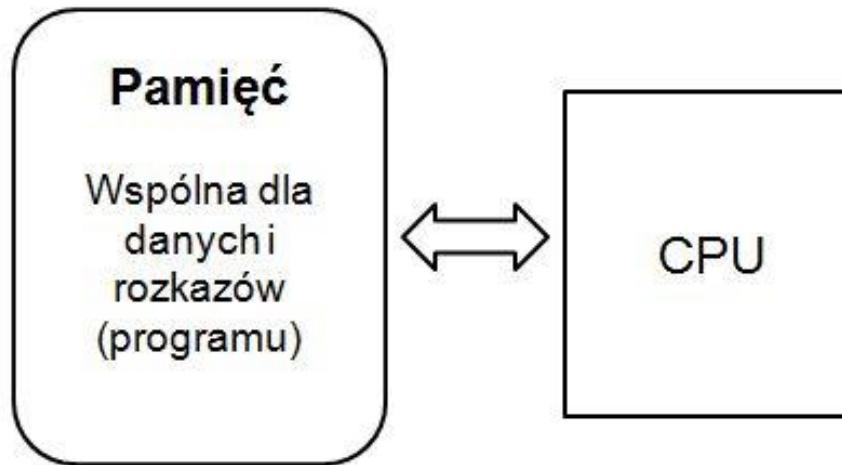
# Model von Neumana



ALU - jednostka arytmetyczno-logiczna  
CU - jednostka sterująca  
I/O - układy wejścia-wyjścia

Model ten jest implementacją czysto matematycznej konstrukcji zwanej maszyną Turinga.

## Model von Neumana

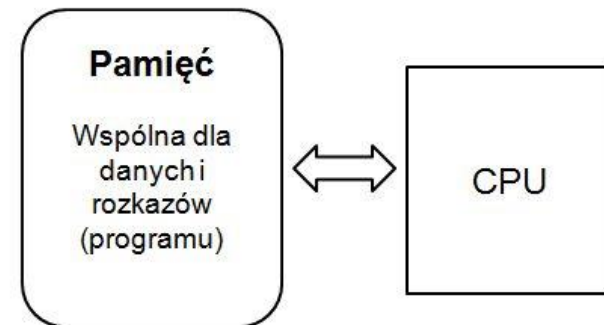


Podstawowym założeniem, stanowiącym część definicji komputera, jest **przechowywanie zarówno danych** (argumentów operacji i ich wyników), **jak i rozkazów** (informacji sterującej przetwarzaniem) **we wspólnej pamięci** .

## Model von Neumana

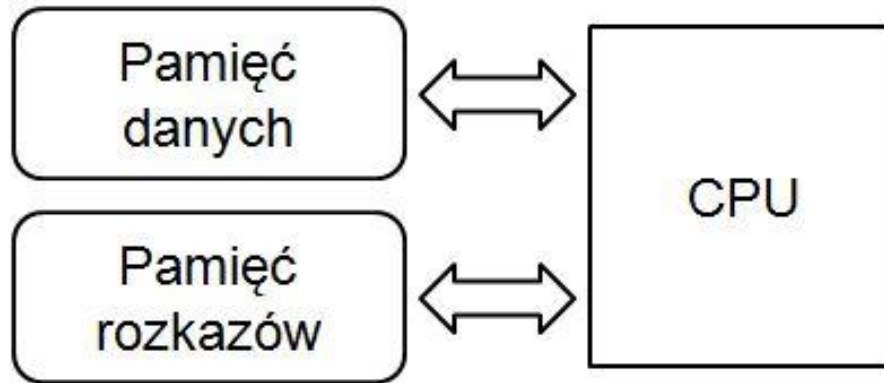
---

- ✓ **Jednolitość postaci informacji przetwarzanej i kodu sterującego umożliwia modyfikację programu w czasie jego wykonywania.**
- ✓ **Ta właśnie cecha decyduje o niebywalej uniwersalności komputerów**
- ✓ **Komputer wykonać może dowolny algorytm, a dostosowanie go do nowego zadania wymaga wyłącznie wprowadzenia do pamięci operacyjnej nowego zestawu instrukcji (wczytania nowego programu).**



## Model Harwardzki

---



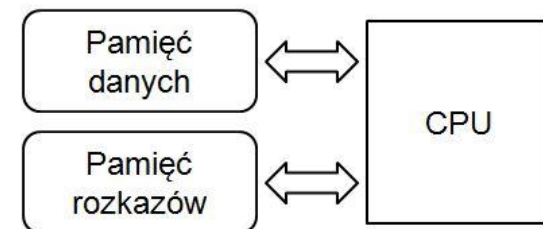
Drugim możliwym modelem komputera jest **model harwardzki**.

W architekturze harwardzkiej pamięć podzielona jest na dwie odrębne części – pamięć danych i pamięć programu

## Model Harwardzki

---

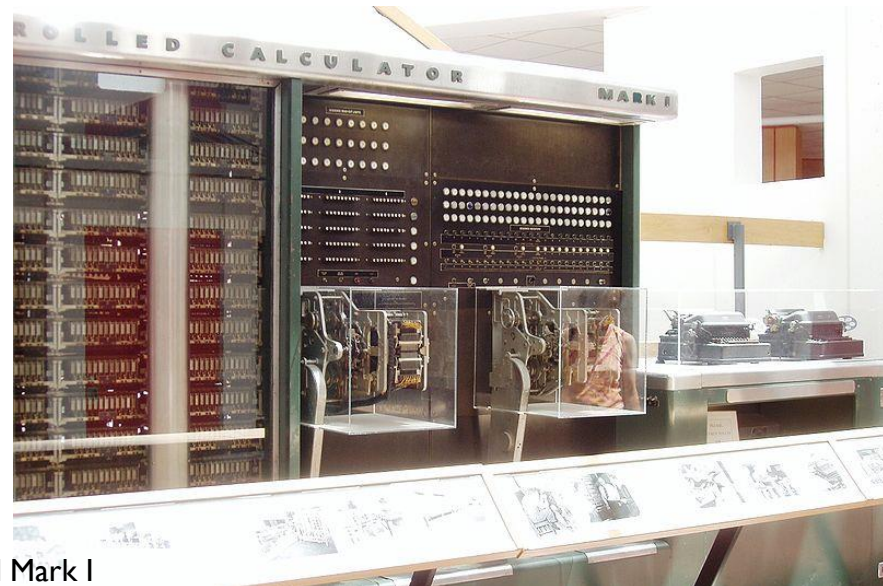
- ✓ Komputer taki jest, mówiąc najprościej, komputerem zaprojektowanym do wykonywania jednego programu.
- ✓ Program ten zapisany jest na stałe w pamięci RAM.
- ✓ **Zaletą tego rozwiązania jest duża wydajność.**
- ✓ W technologii tej wykonywane są obecnie procesory sygnału (DSP) wykorzystywane między innymi w kamerach i aparatach cyfrowych oraz mikrokontrolery jednokładowe, obecne między innymi w nowoczesnych samochodach.





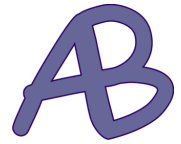
## CIEKAWOSTKA

Najstarsze komputery (komputery generacji zerowej i pierwsze komputery generacji pierwszej) zaprojektowane były w oparciu o model harwardzki. Algorytm przetwarzania danych, według którego pracowały, wynikał bezpośrednio z ich budowy. Zmiana programu wymagała więc fizycznego przebudowania komputera.



Przełącznikowy komputer Harvard Mark I

# Magistrala

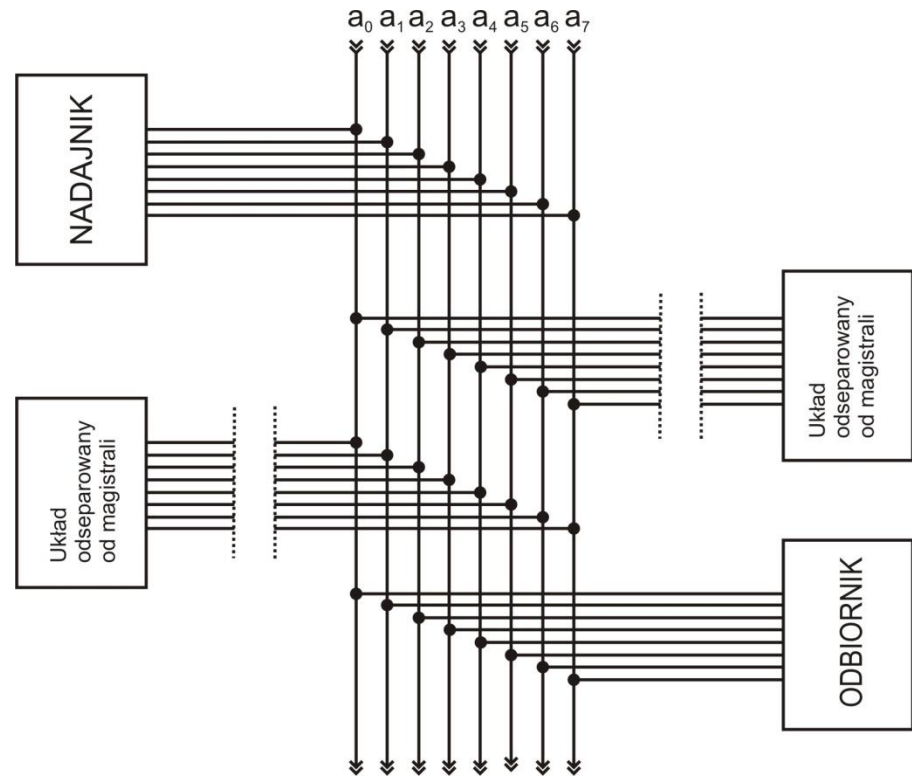


Główne podzespoły komputera (CPU, pamięć operacyjna i układy we/wy).  
Komunikują się za pośrednictwem **magistral**.

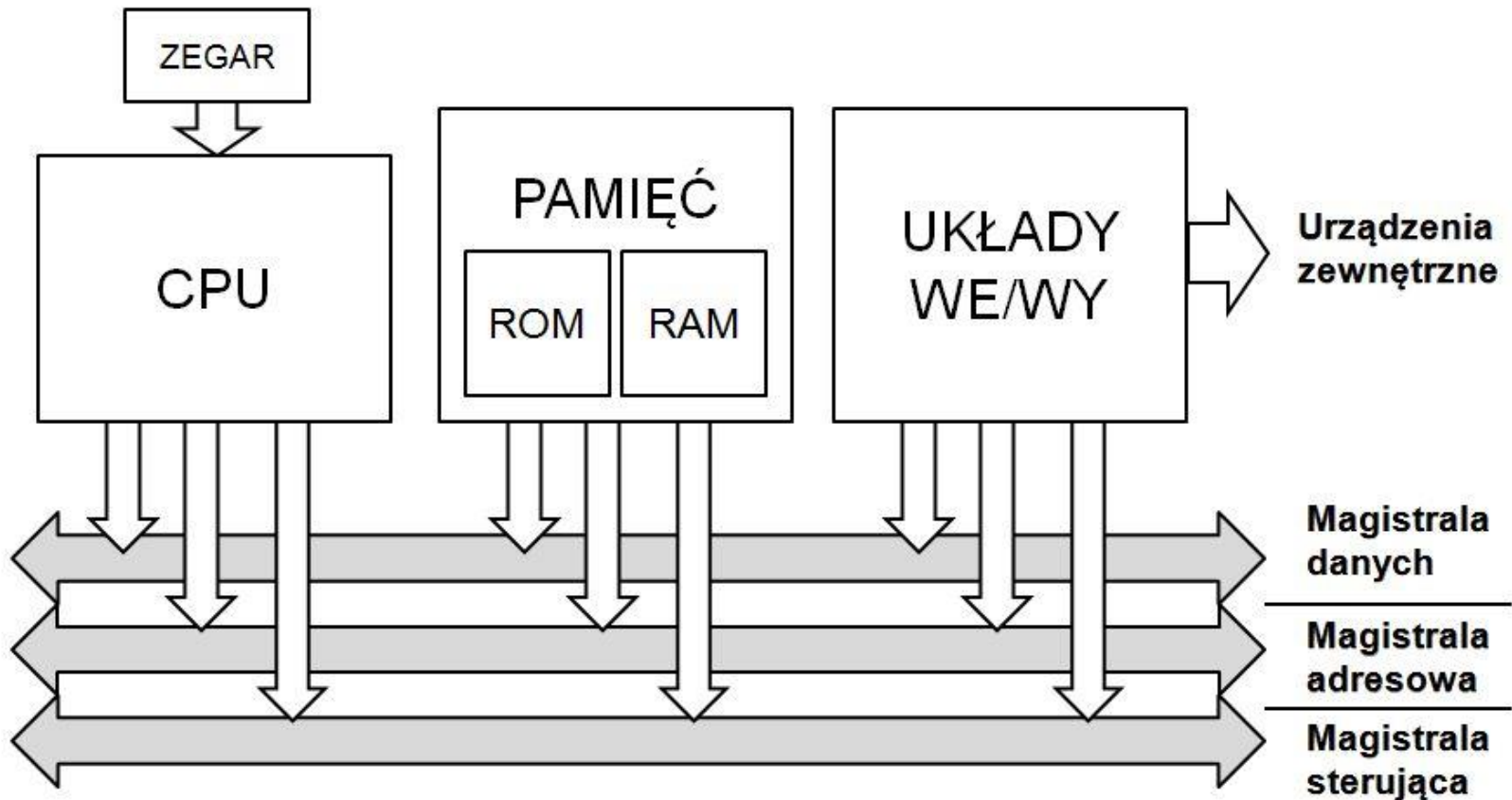
## Definicja:

**Magistralą** nazywamy zestaw linii oraz układów przełączających, łączących dwa lub więcej układów mogących być nadajnikami lub odbiornikami informacji.

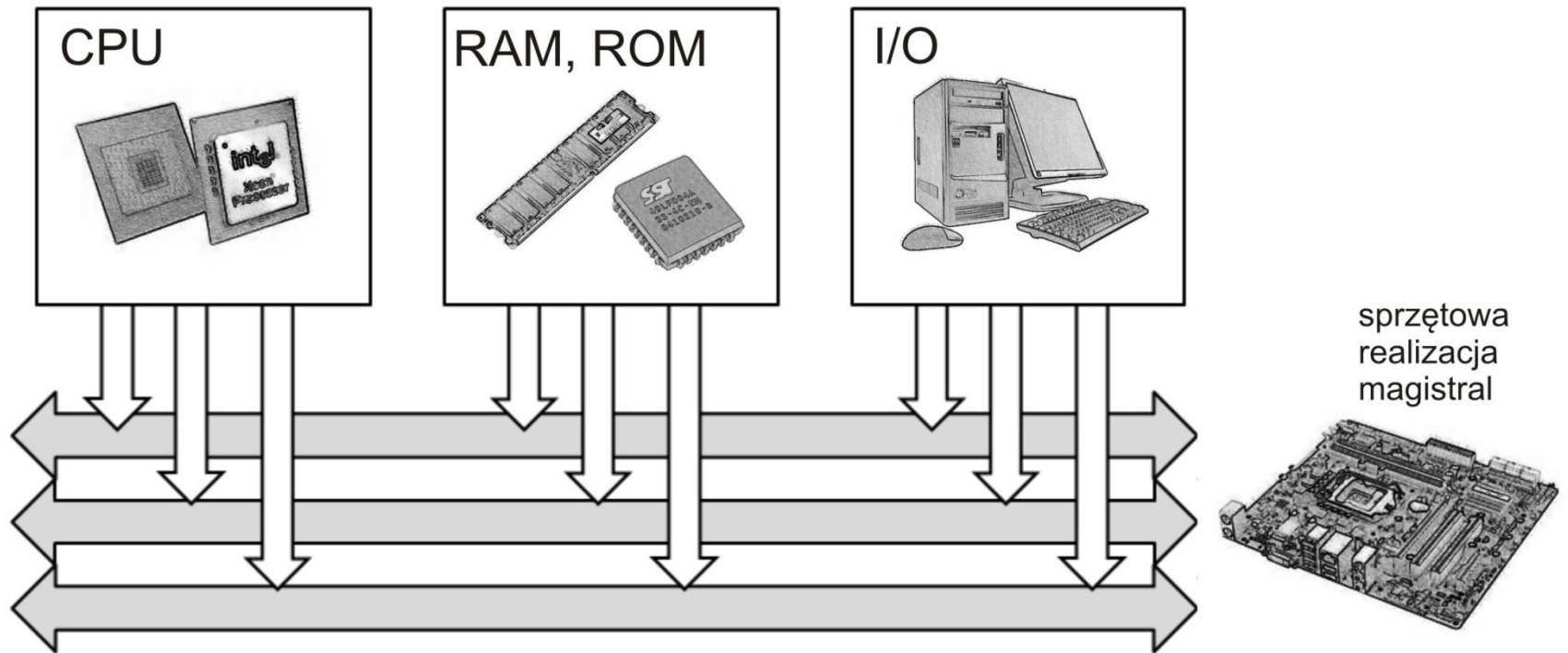
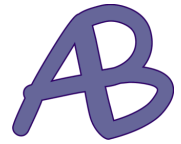
**Przesyłanie informacji** zachodzi zawsze pomiędzy dokładnie jednym układem będącym **nadajnikiem** a dokładnie jednym układem będącym **odbiornikiem**, przy pozostałych układach odseparowanych od linii przesyłających.



# NAJPROSTRSZY MODEL KOMPUTERA



# NAJPROSTRSZY MODEL KOMPUTERA



# NAJPROSTRSZY MODEL KOMPUTERA

## CPU

(mikroprocesor zegar i sterownik magistrali)

W skład **CPU** wchodzi:

- mikroprocesor,
- zegar
- oraz opcjonalnie sterownik magistrali oraz koprocesor matematyczny.

**Zegar systemowy** tworzy przebiegi czasowe synchronizujące pracę wszystkich elementów systemu.

**Sterownik magistrali** jest specjalizowanym układem który na podstawie informacji otrzymanych z mikroprocesora sygnały sterujące pracą pamięci układów we/wy i innych.

# NAJPROSTRSZY MODEL KOMPUTERA



## CPU

(mikroprocesor zegar i sterownik magistrali)

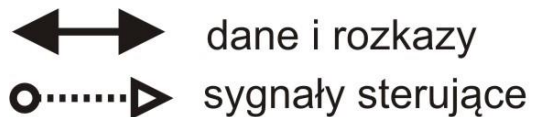
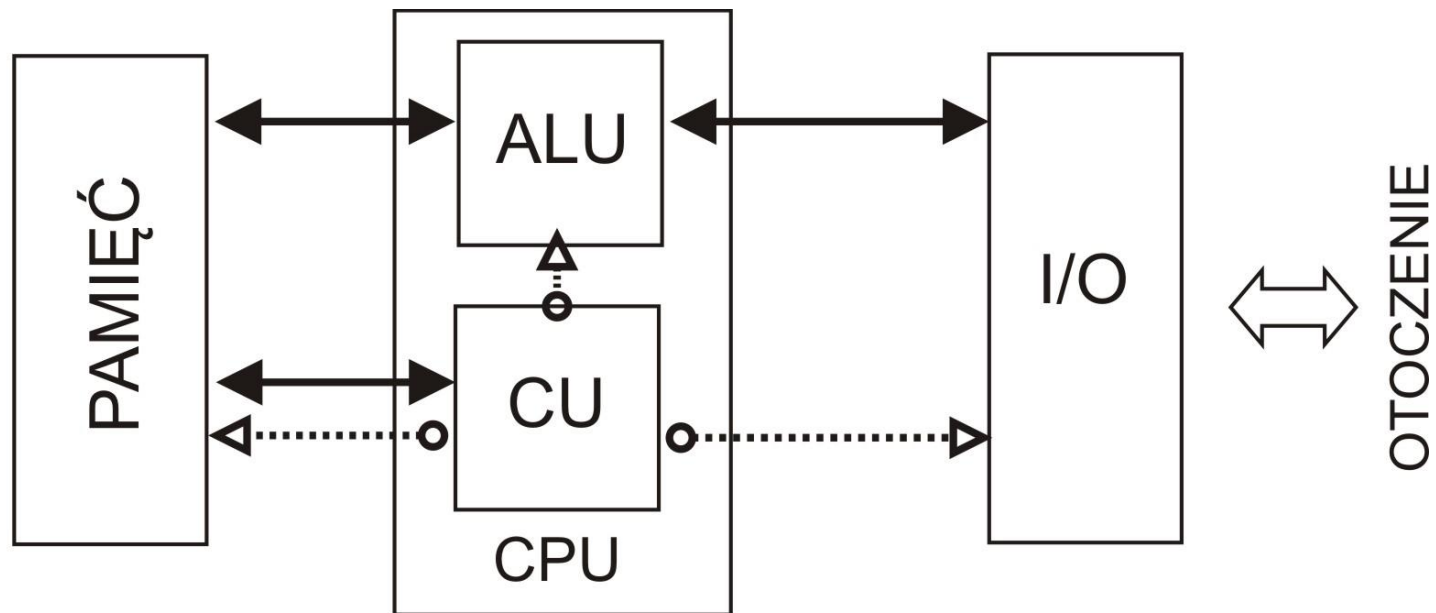
**Zadaniem CPU** oprócz przetwarzania informacji jest także sterowanie pracą pozostałych układów systemu.

**Wszystkie działania i operacje** w systemie są sterowane lub zainicjowane przez procesor.

**Działanie** jest skutkiem ciągu instrukcji dostarczonych do mikroprocesora czyli programu.

# NAJPROSTRSZY MODEL KOMPUTERA

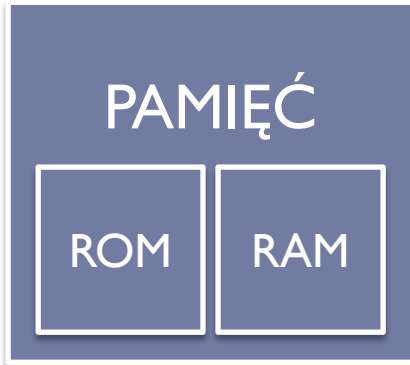
**Wszystkie działania i operacje** w systemie są sterowane lub zainicjowane przez procesor.



ALU - jednostka arytmetyczno-logiczna  
CU - jednostka sterująca  
I/O - układy wejścia-wyjścia

# NAJPROSTRSZY MODEL KOMPUTERA

---



**Program** musi być przechowywany w pamięci o krótkim czasie dostępu (pamięć półprzewodnikowa ROM). Pamięci masowa nie nadają się - mają zbyt długi czas dostępu.

**W każdym systemie komputerowym** musi istnieć pamięć stała (RAM), która przechowuje instrukcję (niewielki fragment kodu), od której procesor startuje (wykonuje ją po załączeniu zasilania).





## Literatura:

---

Metzger Piotr - Anatomia PC, wydanie XI, Helion 2007

Wojtuszkiewicz Krzysztof - Urządzenia techniki komputerowej, część I: Jak działa komputer, MIKOM, Warszawa 2000

Wojtuszkiewicz Krzysztof - Urządzenia techniki komputerowej, część II: Urządzenia peryferyjne i interfejsy, MIKOM, Warszawa 2000

Komorowski Witold - Krótki kurs architektury i organizacji komputerów, MIKOM Warszawa 2004

Gook Michael - Interfejsy sprzętowe komputerów PC, Helion, 2005

