

Informatyka 1 (EZ1E2008)

Politechnika Białostocka - Wydział Elektryczny
Elektrotechnika, semestr II, studia niestacjonarne I stopnia
Rok akademicki 2021/2022

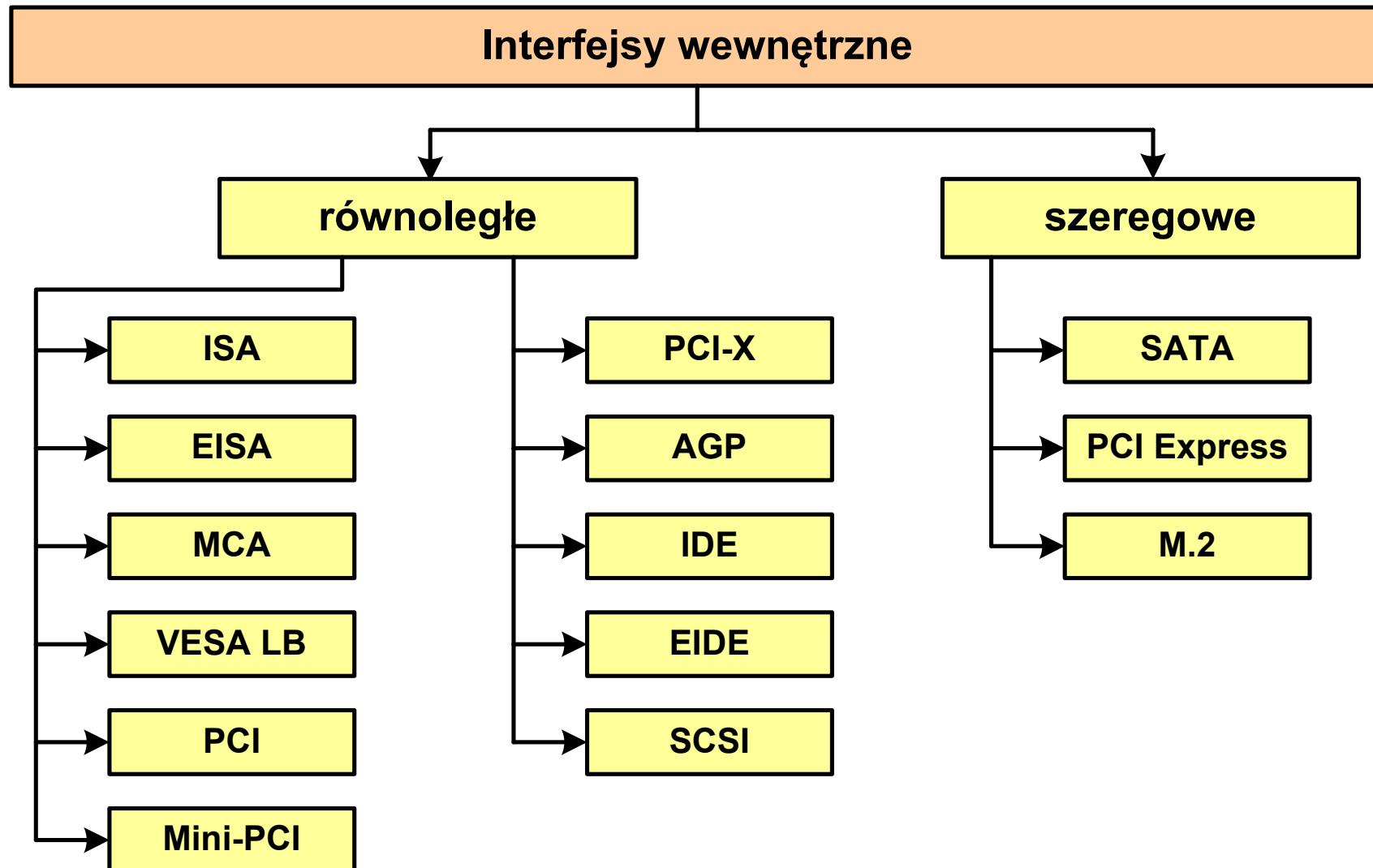
Wykład nr 7 (13.05.2022)

dr inż. Jarosław Forenc

Plan wykładu nr 7

- Budowa komputera
 - interfejsy wewnętrzne i zewnętrzne
- Język C - tablice jednowymiarowe (wektory)
 - deklaracja, odwołania do elementów, inicjalizacja tablicy
 - generator liczb pseudolosowych, operacje na wektorze
- Struktura i funkcjonowanie komputera
 - procesor, rozkazy, przerwania, magistrala

Interfejsy sprzętowe komputera



AGP

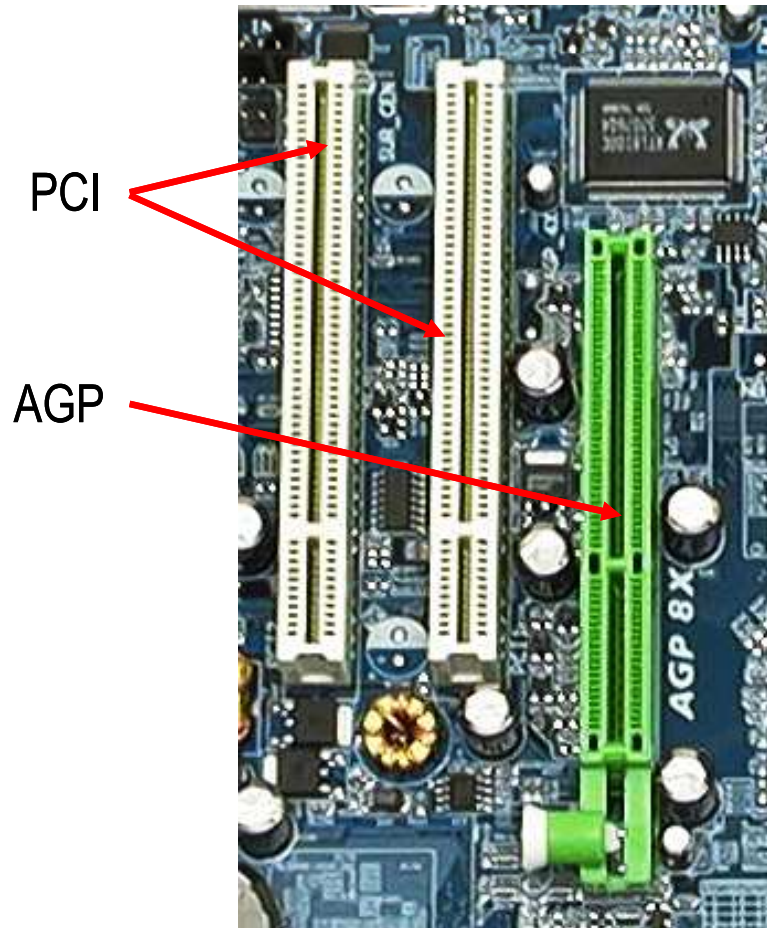
(wewnętrzny, równoległy)

- **AGP** - Accelerated / Advanced Graphics Port
- opracowana w 1996 r. przez firmę Intel
- 32-bitowa modyfikacja magistrali PCI zoptymalizowana do szybkiego przesyłania dużej ilości danych pomiędzy pamięcią operacyjną a kartą graficzną
- maksymalna moc pobierana przez kartę AGP to 35-40 W
- przy większym zapotrzebowaniu na energię doprowadza się dodatkowe zasilanie (złącze Molex)

Wersja	Rok	Napięcie	Mnożniki / Przepustowość
AGP 1.0	1996	3,3 V	1x - 267 MB/s, 2x - 533 MB/s
AGP 2.0	1998	1,5 V	1x - 267 MB/s, 2x - 533 MB/s, 4x - 1067 MB/s
AGP 3.0	2002	0,8 V	4x - 1067 MB/s, 8x - 2133 MB/s

AGP

(wewnętrzny, równoległy)



IDE (wewnętrzny, równoległy)

- **IDE** - Intelligent Drive Electronics, Integrated Device Electronics
- inne nazwy:
 - ATA - Advanced Technology Attachments
 - AT-BUS
 - PATA - Parallel ATA
- interfejs przeznaczony do komunikacji z dyskami twardymi
- w systemie tym, w przeciwieństwie do poprzedniego ST412/506, kontroler jest zintegrowany z dyskiem
- dyski komunikują się z szynami systemowymi za pośrednictwem host-adaptera umieszczonego na płycie głównej lub dodatkowej karcie rozszerzającej (starsze systemy)
- IDE dopuszczał obsługę do dwóch dysków twardych (Master i Slave) o maksymalnej pojemności 504 MB (dziesiętnie 528 MB)

IDE (wewnętrzny, równoległy)

- maksymalna długość przewodu łączącego dysk z host adapterem wynosiła 18 cali, czyli ok. 46 cm
- przewód ten miał trzy wtyki - kontroler, urządzenie Master i Slave
- żadne przewody nie były krzyżowane, dlatego fizyczna kolejność urządzeń na magistrali nie odgrywała żadnej roli



40-żyłowa
taśma IDE



EIDE

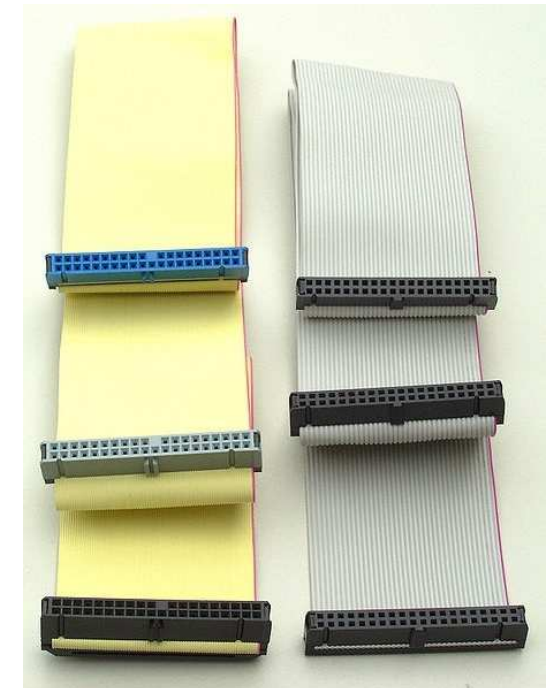
(wewnętrzny, równoległy)

- **EIDE** - Enhanced IDE
- EIDE miał usunąć ograniczenia standardu IDE, zapewniając przy tym pełną z nim zgodność
- opracowano różne wersja standardu EIDE:
 - ATA-2 (1994 r.)
 - ATA-3 (1996 r.)
 - ATA/ATAPI-4 (1997 r.) - możliwość podłączenia innych urządzeń niż dysk twardy - streamer, CD-ROM
 - ATA-ATAPI-5 (2000 r.)
 - ATA-ATAPI-6
- EIDE umożliwia obsługę dwóch host-adapterów (Primary, Secondary), czyli podłączenie do czterech urządzeń

EIDE

(wewnętrzny, równoległy)

- Problem ograniczenia pojemności dysków standardu IDE do 504 MB został rozwiązany na dwa sposoby:
 - adresowanie CHS (ang. Cylinder, Head, Sector)
 - adresowanie LBA (ang. Logical Block Addressing)
- Zwiększenie pasma przepustowego magistrali osiągnięto przez zastosowanie trybów pracy:
 - Ultra DMA/33 (Ultra-ATA) - przewód 40-żyłowy,
 - Ultra DMA/66 - 40 przewodów sygnałowych, ale przewód 80-żyłowy - każdy przewód sygnałowy oddzielony jest od sąsiada dodatkową linią masy, poszczególne wtyki przewodu opisane są i oznaczone różnymi kolorami: kontroler - niebieski, Master - czarny, Slave - szary,
 - Ultra ATA/100
 - Ultra ATA/133



SCSI

(wewnętrzny, równoległy)

- **SCSI** - Small Computer Systems Interface
- równoległa magistrala danych przeznaczona do przesyłania danych między urządzeniami (dyski twarde, skanery, drukarki, nagrywarki)
- wykorzystywana głównie w wysokiej klasy serwerach i stacjach roboczych
- magistrala wymaga zakończenia jej terminatorem



Wersja	Przepustowość	Rok
SCSI-1	5 MB/s	1986
SCSI-2 (Fast SCSI)	10 MB/s	1994
SCSI-2 (Wide SCSI)	20 MB/s	1994
SCSI-3 (Ultra SCSI)	20-40 MB/s	1996
Ultra2 SCSI	40-80 MB/s	1997
Ultra3 SCSI (Ultra 160 SCSI)	160 MB/s	1999
Ultra4 SCSI (Ultra 320 SCSI)	320 MB/s	2002
Ultra 640 SCSI	640 MB/s	2003

SCSI

(wewnętrzny, równoległy)



Kontroler SCSI



Kabel SCSI

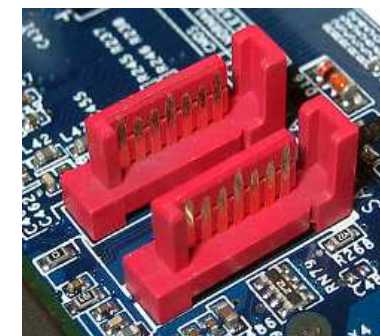


Skaner ze
złączem
SCSI

Serial ATA (wewnętrzny, szeregowy)

- **Serial ATA** - Serial Advanced Technology Attachment, SATA
- szeregową magistralą służącą do komunikacji Host Bus Adaptera z urządzeniami pamięci masowej (dyski twarde, napędy optyczne)
- zastąpiła równoległą magistralę ATA
- węższe i dłuższe (do 1 m) przewody niż w ATA
- 7-pinowa wtyczka sygnałowa
- 15-pinowa wtyczka zasilania

Generacja	Przepustowość
SATA I	1,5 Gbit/s (ok. 150 MB/s)
SATA II	3,0 Gbit/s (ok. 300 MB/s)
SATA III (3.0)	6,0 Gbit/s (ok. 600 MB/s)
SATA III (3.5)	6,0 Gbit/s (ok. 600 MB/s)



PCI Express

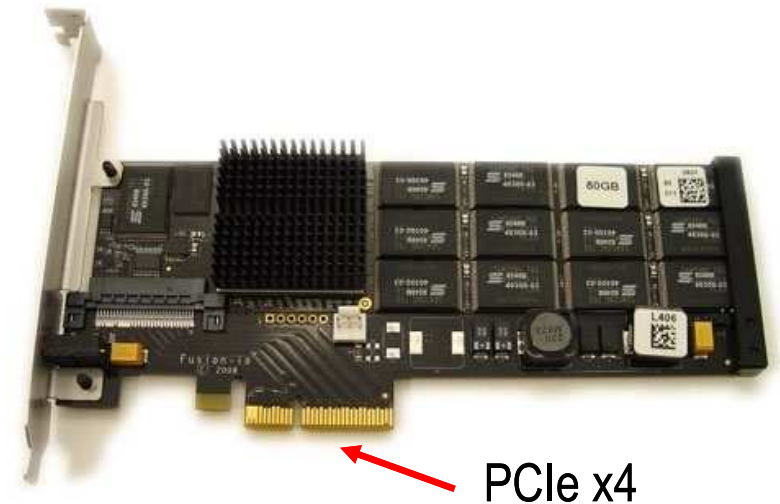
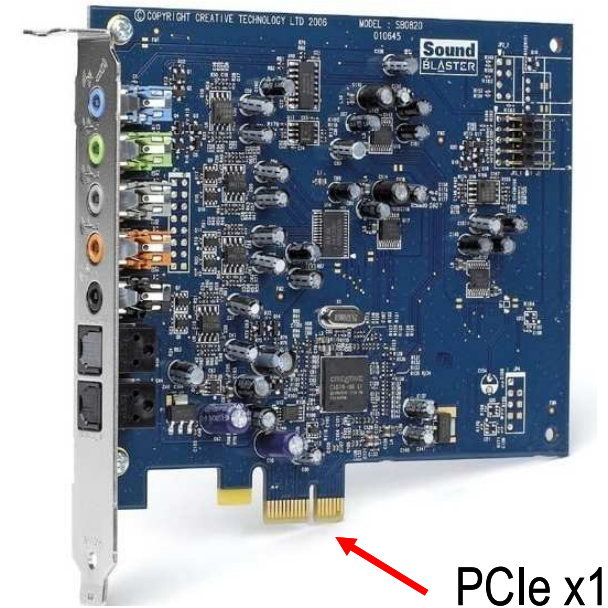
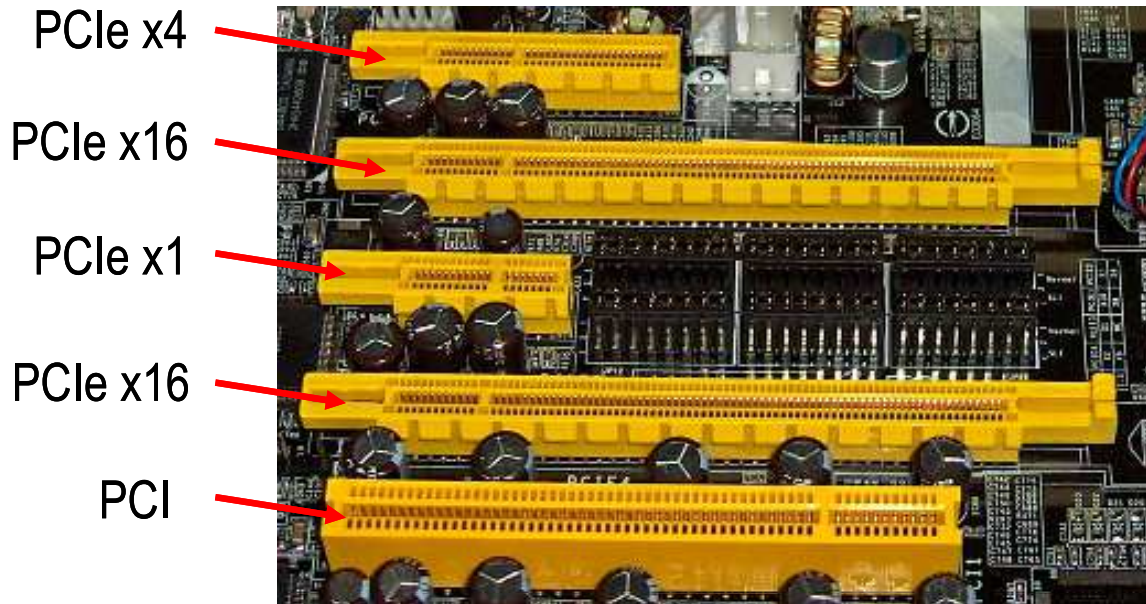
(wewnętrzny, szeregowy)

- **PCI Express** - Peripheral Component Interconnect Express, PCIe
- złącze przeznaczone do instalacji kart rozszerzeń na płycie głównej (graficzne, muzyczne, sieciowe, kontrolery IDE, SATA, USB)
- każde urządzenie jest połączone bezpośrednio z kontrolerem
- PCI Express zastąpił PCI i AGP
- jeśli podłączona karta wymaga więcej energii to jest zasilana przez dodatkowy przewód

Wersja	Rok	Przepustowość (x1)	Przepustowość (x16)
1.0	2003	0,250 GB/s	4,000 GB/s
2.0	2007	0,500 GB/s	8,000 GB/s
3.0	2010	0,985 GB/s	15,754 GB/s
4.0	2017	1,969 GB/s	31,508 GB/s
5.0	2019	3,938 GB/s	63,015 GB/s

PCI Express

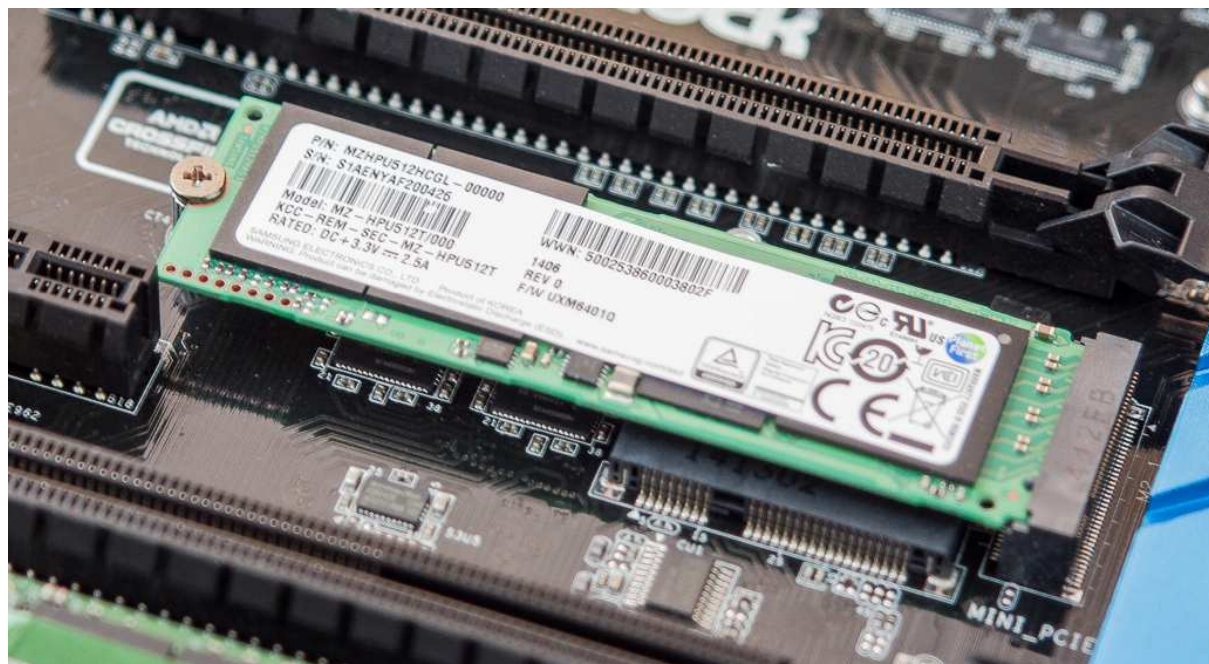
(wewnętrzny, szeregowy)



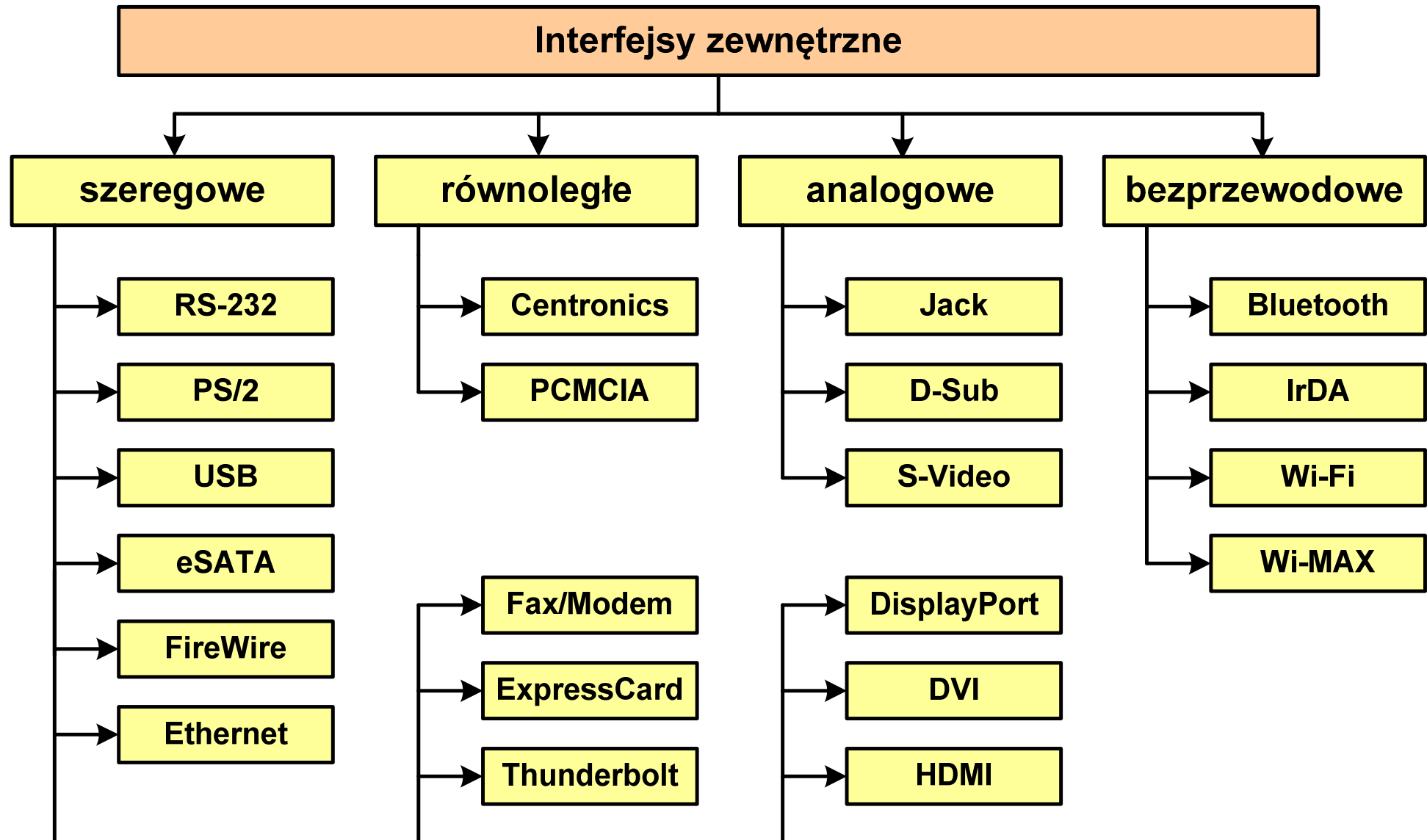
M.2

(wewnętrzny, szeregowy)

- inna nazwa: NGFF - Next Generation Form Factor
- złącze kart rozszerzeń zastępujące interfejs mSATA
- prędkość transmisji: 4 GB/s, 8 GB/s



Interfejsy sprzętowe komputera



RS-232

(zewnątrzny, szeregowy)

- **RS-232** (Recommended Standard 232)
- 1962 rok
- magistrala przeznaczona do szeregowej transmisji danych
- najbardziej popularna wersja standardu: RS-232C
- przepustowość: do 115,2 kbit/s
- długość magistrali: do ok. 15 m
- w architekturze PC przewidziano obecność do 4 portów COM (COM1-COM4)
- zastosowania: mysz komputerowa, modemy, telefony komórkowe, łączenie dwóch komputerów kablem, starsze drukarki, tunery satelitarne, programowanie układów logicznych
- obecnie zastąpiona przez USB

RS-232

(zewnątrzny, szeregowy)



DE-9 (gniazdo męskie)



DB-25 (gniazdo żeńskie)



DE-9 (wtyk żeński)

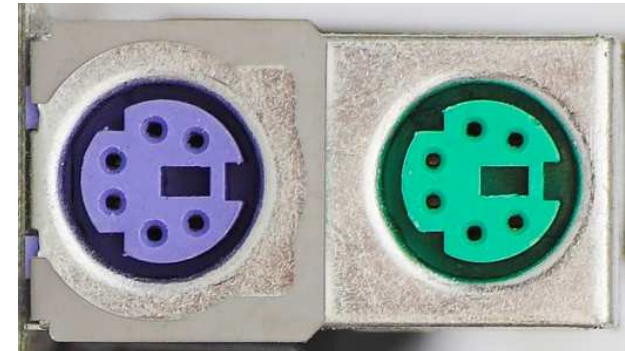


DB-25 (wtyk męski)

PS/2

(zewnątrzny, szeregowy)

- złącze używane do podłączenia klawiatury i myszy komputerowej
- IBM, 1987 rok
- zastąpiło złącze szeregowe myszy DE-9 i złącze klawiatury DIN
- przepustowość: 40 kB/s
- długość: 1,8 m
- zastąpione przez USB
- klawiatura - kolor fioletowy
- mysz - kolor zielony



6-pin Mini-DIN connector

USB

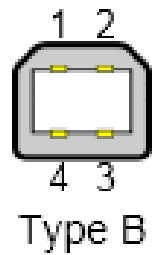
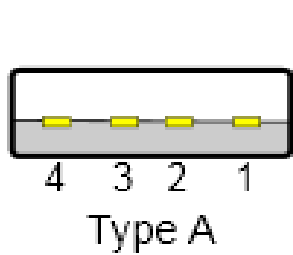
(zewnątrzny, szeregowy)

- **USB** (Universal Serial Bus)
- port komunikacyjny zastępujący stare porty szeregowy i równoległe
- zastosowanie: kamery i aparaty cyfrowe, telefony komórkowe, dyski, modemy, skanery, myszki, klawiatury, pen-drive'y, ...
- w systemie Windows obsługa USB od Windows 95 OSR2

Wersja	Przepustowość	Rok	Zasilanie	Przewód
USB 1.1 (Low Speed)	do 1,5 Mbit/s	1998	5 V, 500 mA	3 m
USB 1.1 (Full Speed)	do 12 Mbit/s	1998	5 V, 500 mA	5 m
USB 2.0 (Hi-Speed)	do 480 Mbit/s	2000	5 V, 500 mA	5 m
USB 3.0 (SuperSpeed)	do 4,8 Gbit/s	2008	5 V, 900 mA	3 m
USB 3.1 (SuperSpeed+)	do ok.10 Gbit/s	2014	5 V, 2 A	1 m
USB 3.2	do ok. 20 Gbit/s	2017	5 V	1 m

USB

(zewnątrzny, szeregowy)



Type A



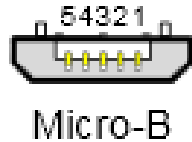
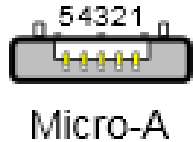
Type B



Mini-A

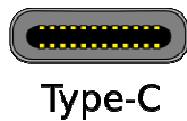


Mini-B



Micro-A

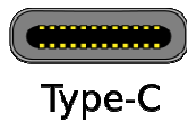
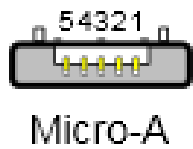
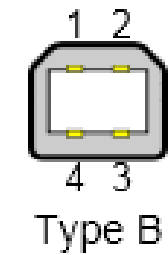
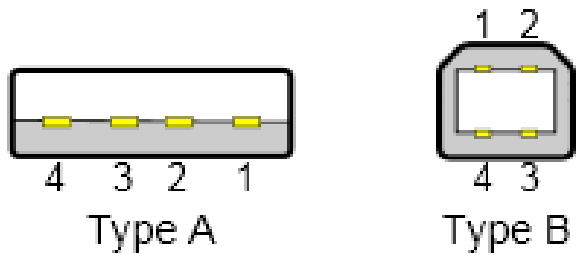
Micro-B



Type-C

USB

(zewnątrzny, szeregowy)



eSATA

(zewnątrzny, szeregowy)

- **eSATA** (external SATA) - 2004 rok
- zewnętrzny port SATA 3 Gbit/s przeznaczony do podłączania pamięci masowych zewnętrznych
- maksymalne przepustowości: 150 MB/s, 300 MB/s
- maksymalna długość kabla: 2 m



FireWire

(zewnątrzny, szeregowy)

- standard złącza szeregowego umożliwiający szybką komunikację i synchroniczne usługi w czasie rzeczywistym
- 1995 rok, dokument IEEE 1394
- przepustowość: 400/800/1600/3200 Mbit/s
- długość kabla: do 4,5 m
- złącze: IEEE-1394 (4, 6 lub 9 pinów)
- zastosowania: kamery i aparaty cyfrowe, skanery, drukarki



9-pin, 6-pin connectors



6-pin IEEE-1394 ports



4-pin connectors

Ethernet

(zewnątrzny, szeregowy)

- **BNC (Bayonet Neill-Concelman)** - złącze stosowane do łączenia sieci komputerowych zbudowanych z kabli koncentrycznych
- występuje w wersji 50 i 75-omowej



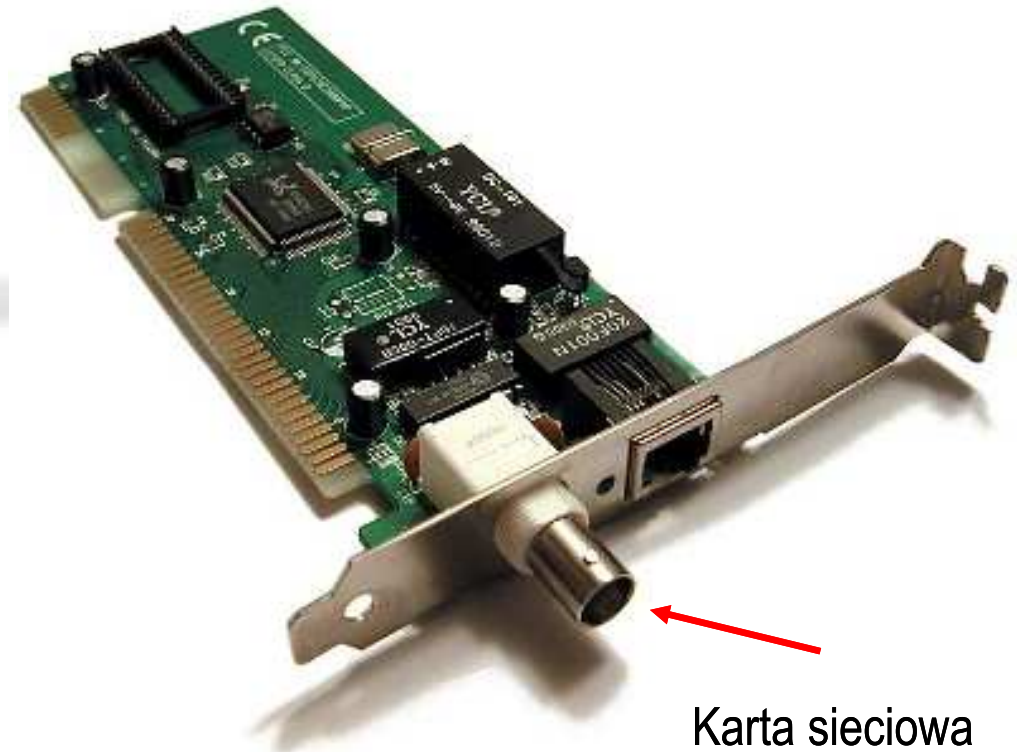
Złącze BNC



Trójnik



Terminator



Karta sieciowa
ze złączem BNC

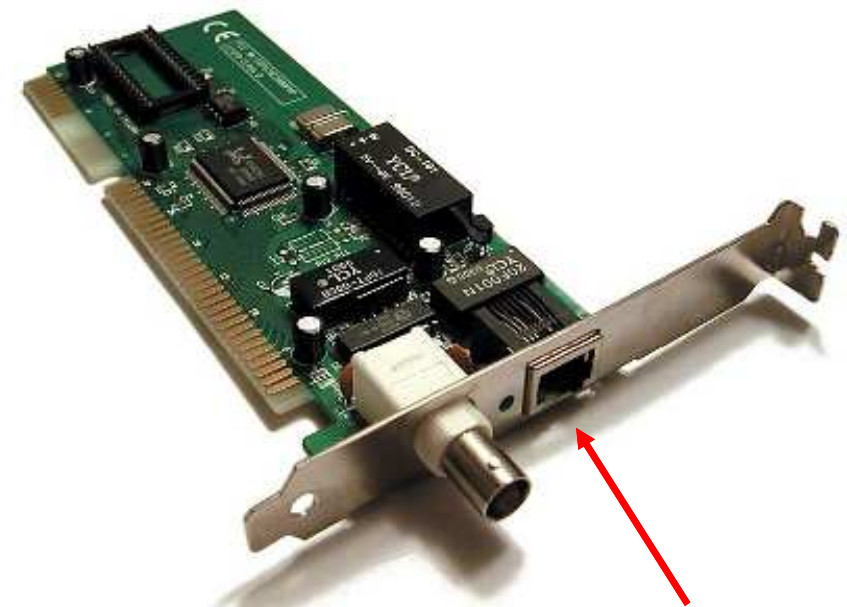
Ethernet

(zewnątrzny, szeregowy)

- **8P8C (8 Position 8 Contact)** - ośmiostykowe złącze wykorzystywane w sprzęcie komputerowym i telekomunikacyjnym
- nazywane RJ-45



Złącze 8P8C
na płycie głównej



Karta sieciowa
ze złączem 8P8C

Fax/Modem (RJ-11) (zewnątrzny, szeregowy)

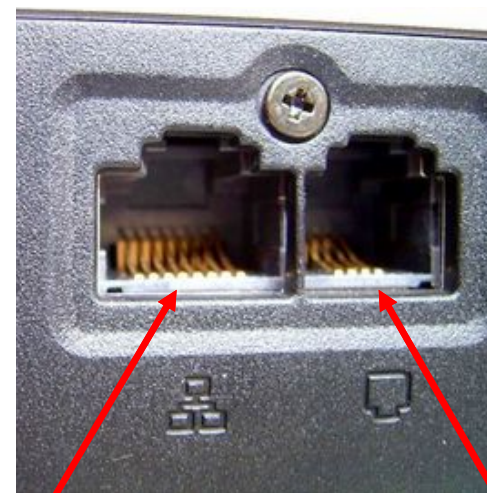
- **RJ-11 (Registered Jack - Type 11)** - złącze stosowane do podłączania sprzętu telekomunikacyjnego (linii telefonicznej)
- **6P2C (6 Position 2 Contact)** - sześciokrotny wtyk telefoniczny z dwoma stykami stosowany do zakończenia przewodów łączących sprzęt telekomunikacyjny



Wtyk RJ-11



Gniazdo RJ-11



RJ-45

RJ-11

Thunderbolt (zewnątrzny, szeregowy)

- interfejs do podłączania urządzeń zewnętrznych
- w założeniu ma zastąpić USB, FireWire, HDMI
- opracowanie - 2009 rok, pierwsze urządzenia - 2011 rok
- Intel, Apple Inc.
- przepustowość: 10 Gbit/s (Thunderbolt 1), 20 Gbit/s (Thunderbolt 2), 40 Gbit/s (Thunderbolt 3)



Złącze Thunderbolt w laptopie



Wtyczka
Thunderbolt

DisplayPort (zewnątrzny, szeregowy)

- **DisplayPort** - uniwersalny interfejs cyfrowy do przesyłania dźwięku i obrazu z prędkością 1,62 lub 2,7 Gb/s
- opracowany w 2006 roku
- dwukierunkowa wymiana informacji
- możliwa ochrona sygnału technologią DRM



Wtyk i gniazdo DisplayPort

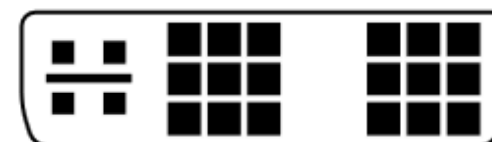


Gniazdo DisplayPort

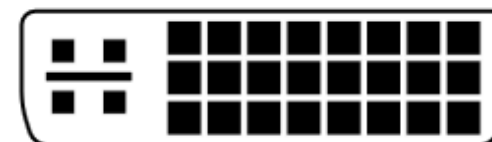
DVI

(zewnątrzny, szeregowy)

- **DVI (Digital Visual Interface)** - standard złącza pomiędzy kartą graficzną a monitorem komputera
- wersje:
 - **DVI-I** - przesyła dane cyfrowe i analogowe
 - **DVI-D** - przesyła dane cyfrowe
 - **DVI-A** - przesyła dane analogowe



DVI-I (Single Link)



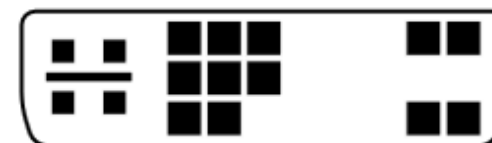
DVI-I (Dual Link)



DVI-D (Single Link)



DVI-D (Dual Link)



DVI-A



HDMI

(zewnątrzny, szeregowy)

- **HDMI (High Definition Multimedia Interface)** - interfejs do przesyłania cyfrowe, nieskompresowanego sygnału audio i wideo
- wrzesień 2003 r.
- wersje:
 - 1.0, 1.1, 1.2, 1.3, 1.4
 - 2.0, 2.0a, 2.0b (4096x2160p60)
 - 2.1 (2017 r., 48 Gb/s, 7680×4320p120)



IEEE 1284 (zewnątrzny, równoległy)

- port równoległy wykorzystywany do podłączenia urządzeń peryferyjnych (drukarki, skanery, plotery)
- nazywany **portem równoległym** lub **LPT** (Line Print Terminal)
- standard IEEE 1284 został opracowany w 1994 roku
- zapewnia kompatybilność z używanym w latach 70-tych jednokierunkowym portem **Centronics**
 - LPT1, I/O Port 0x378, IRQ7 + LPT2, I/O Port 0x278, IRQ5
- protokoły transmisji danych (wybrane):
 - **SPP** (Standard Parallel Port) - tryb kompatybilności z Centronics, możliwość transmisji dwukierunkowej, transfer do 150 kb/s, obsługa za pomocą przerwań
 - **EPP** (Enhanced Parallel Port) - sprzętowo ustalone parametry transmisji (automatycznie), brak kanału DMA
 - **ECP** (Extended Capability Port) - używa DMA, transfer do 2 Mb/s

IEEE 1284

(zewnątrzny, równoległy)



Port równoległy w laptopie



DB-25



Port równoległy
na płycie głównej

PCMCIA (zewnątrzny, równoległy)

- Personal Computer Memory Card International Association
- 1991 - standard interfejsu wejścia-wyjścia dla kart pamięci
- w kolejnych latach przekształcony w karty rozszerzeń, pełniące funkcje modemu, faksmodemu, karty sieciowej, Wi-Fi
- ustandaryzowane wymiary: 85,6 × 54 mm
- podział ze względu na wielkość:
 - **typ I** - grubość 3,3 mm; karty pamięci SRAM lub Flash
 - **typ II** - grubość 5,0 mm; karty rozszerzeń (modem, karta sieciowa)
 - **typ III** - grubość 10,5 mm; karty rozszerzeń (dysk twardy)
- podział ze względu na interfejs:
 - **PC Card 16** - interfejs magistrali ISA 16bit, zasilanie 5 V
 - **CardBus** - interfejs magistrali PCI 32bit, zasilanie 3-3,3 V

PCMCIA

(zewnątrzny, równoległy)



USB card
Type II



Wi-Fi card
Type II



gniazda
PCMCIA



Język C - tablica elementów

- **Tablica** - ciągły obszar pamięci, w którym umieszczone są elementy tego samego typu

wektor

5	3	-2	1	-4
---	---	----	---	----

macierz

a	c	d	m
p	d	q	l
a	t	x	v

1.2	2.5	2.0	10.0
-0.1	4.3	6.2	-5.1
0.0	12.2	4.1	-2.2

Język C - tablica jednowymiarowa

- **Tablica** - ciągły obszar pamięci, w którym umieszczone są elementy tego samego typu
- **Wektor** - tablica jednowymiarowa

5	3	-2	0	-4
---	---	----	---	----

- liczby całkowite

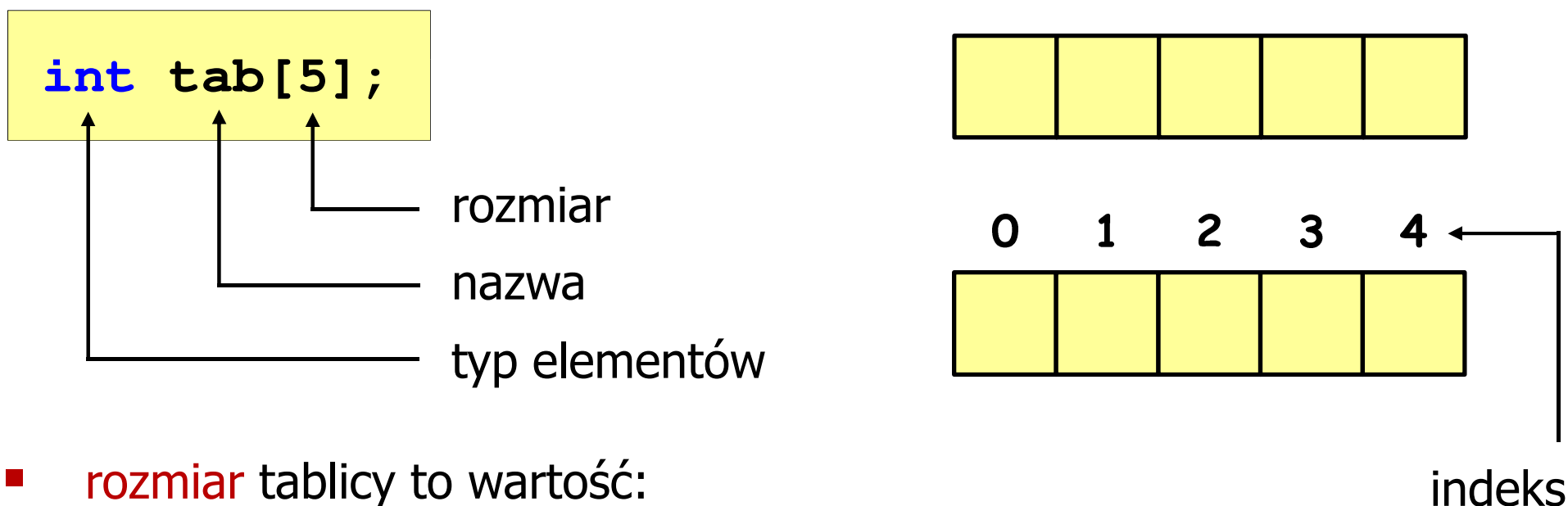
3.1	0.2	2.3	-1.3	1.5	1.1	-4.0
-----	-----	-----	------	-----	-----	------

- liczby rzeczywiste

a	Z	x	&	M	+
---	---	---	---	---	---

- znaki

Język C - deklaracja tablicy jednowymiarowej



- **rozmiar** tablicy to wartość:
 - całkowita, dodatnia
 - znana na etapie kompilacji programu
(stała liczbowa: **5**, `#define N 5`, `const int n = 5;`)

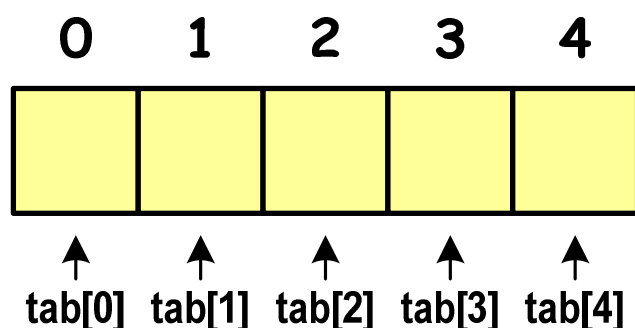
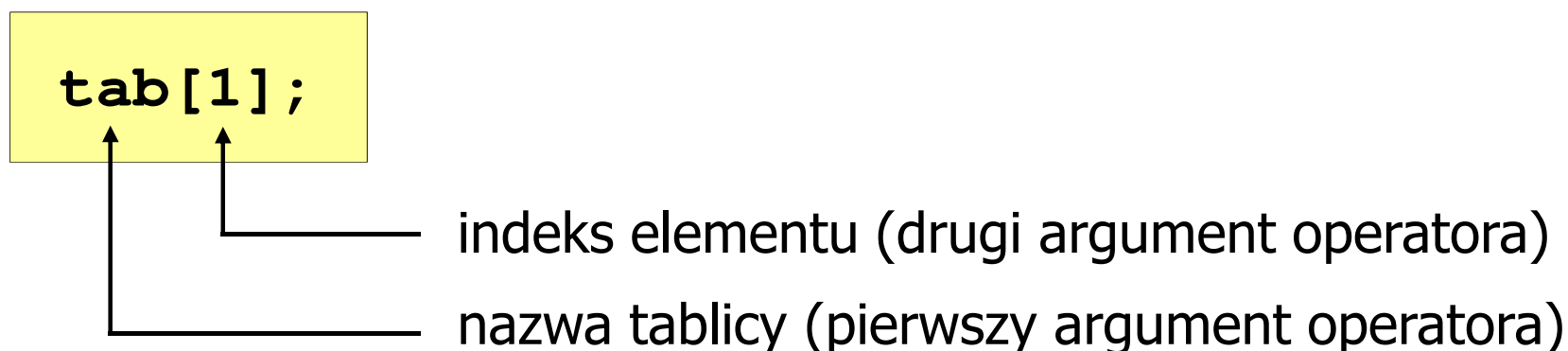
```
int tab[5];
```

```
int tab[N];
```

```
int tab[n];
```

Język C - odwołania do elementów tablicy

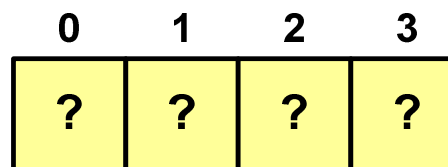
[] - dwuargumentowy operator indeksowania



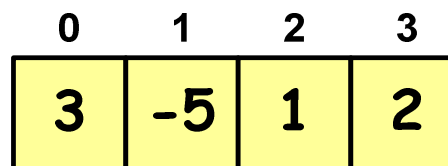
- indeks:
 - stała liczbowa, np. 0, 1, 10
 - nazwa zmiennej, np. i, idx
 - wyrażenie, np. $i*j+5$

Język C - odwołania do elementów tablicy

```
int tab[4];
```



```
tab[0] = 3;  
tab[1] = -5;  
tab[2] = 1;  
tab[3] = 2;
```



- Każdy element tablicy traktowany jest jak zmienna typu `int`

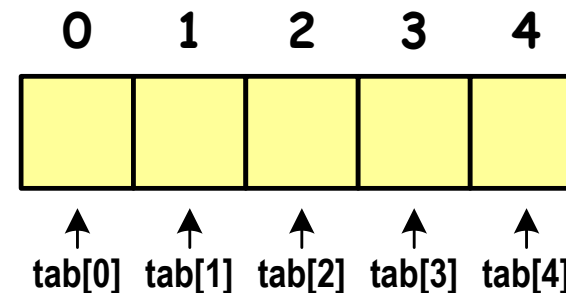
```
printf("%d", tab[0]);
```

```
scanf("%d", &tab[1]);
```


Język C - odwołania do elementów tablicy

- Przy odwołaniach do elementów tablicy kompilator nie sprawdza poprawności indeksów

```
int tab[5];  
tab[5] = 10;
```



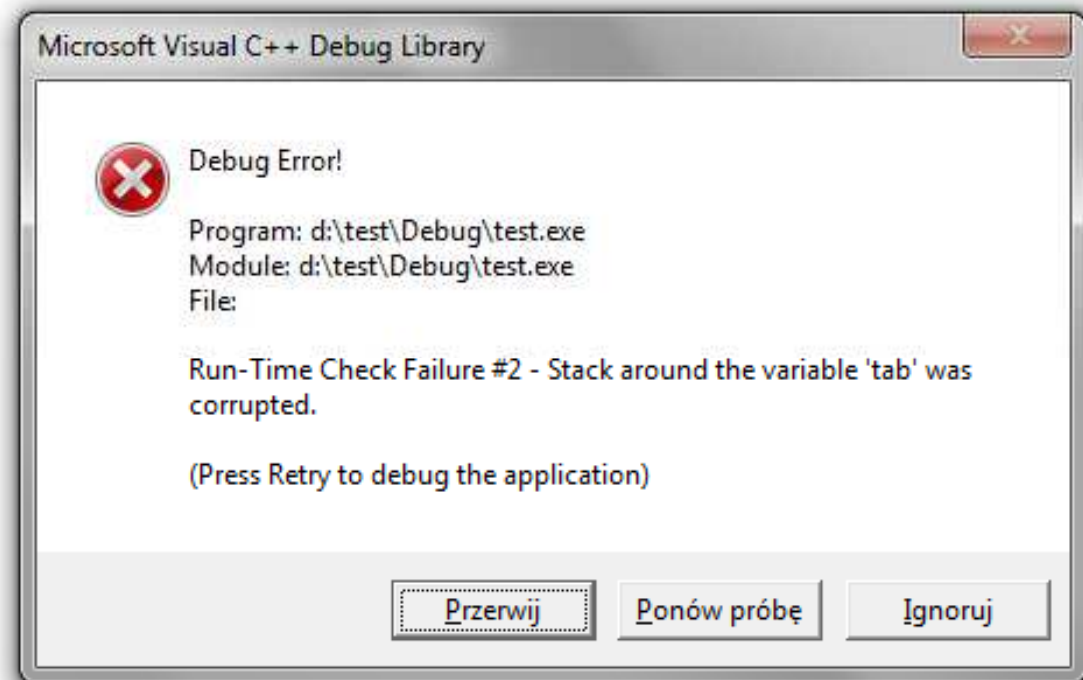
- **błąd!!!** - nie istnieje element **tab[5]**

- Kompilator nie zasygnalizuje błędu
- Program wykona operację
- Środowisko programistyczne może zasygnalizować problem

Język C - odwołania do elementów tablicy

- Przy odwołaniach do elementów tablicy kompilator nie sprawdza poprawności indeksów

```
int tab[5];  
tab[5] = 10;
```



Język C - inicjalizacja tablicy jednowymiarowej

```
int tab[5] = {1, 2, 3, 4, 5};
```

0	1	2	3	4
1	2	3	4	5

```
int tab[5] = {1, 2, 3};
```

0	1	2	3	4
1	2	3	0	0

```
int tab[5] = {1, 2, 3, 4, 5, 6};
```

- błąd kompilacji

```
int tab[] = {1, 2, 3, 4, 5};
```

0	1	2	3	4
1	2	3	4	5

Język C - inicjalizacja tablicy jednowymiarowej

```
int tab[5] = {0};
```

0	1	2	3	4
0	0	0	0	0

```
int tab[5] = {};
```

0	1	2	3	4
0	0	0	0	0

Język C - odwołania do elementów tablicy

- Zapisanie wartości **1** do wszystkich elementów tablicy

```
int tab[5];
```

```
tab[0] = 1;
```

```
tab[1] = 1;
```

```
tab[2] = 1;
```

```
tab[3] = 1;
```

```
tab[4] = 1;
```

0	1	2	3	4
1	1	1	1	1

```
int tab[5], i;
```

```
for (i=0; i<5; i++)
```

```
    tab[i] = 1;
```

Przykład: operacje na dużej ilości danych (tablica)

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    double U[5] = { 5.0, 10.0, 15.0, 20.0, 25.0 };
```

```
    double I[5] = { 0.16, 0.21, 0.27, 0.33, 0.36 };
```

```
    double R[5];
```

```
    int i;
```

```
    for (i=0; i<5; i++)  
        R[i] = U[i]/I[i];
```

```
    for (i=0; i<5; i++)  
        printf("R%d = %f\n", i+1, R[i]);
```

```
    return 0;
```

```
}
```

R1 = 31.250000

R2 = 47.619048

R3 = 55.555556

R4 = 60.606061

R5 = 69.444444

	0	1	2	3	4
U	5.0	10.0	15.0	20.0	25.0
I	0.16	0.21	0.27	0.33	0.36
R	31.25	47.62	55.56	60.61	69.44

Język C - generator liczb pseudolosowych

- `rand()` - zwraca liczbę pseudolosową - zakres: `0 ... RAND_MAX`
(`0 ... 32767`)
- `srand()` - inicjalizuje generator liczb pseudolosowych
- Plik nagłówkowy: `stdlib.h` (`time.h`)

```
int x, y, z;
srand((unsigned int) time(NULL));
x = rand();           // zakres <0, 32767>
y = rand() % 100;     // zakres <0, 99>
z = rand() % (b-a+1) - a; // zakres <a, b>
```

Przykład: operacje na wektorze

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
```

```
#define N 10
```

```
int main(void)
{
```

```
    int tab[N], i;
```

```
    /* generowanie elementów tablicy */
```

```
    srand((unsigned int) time(NULL));
```

```
    for (i=0; i<N; i++)
        tab[i] = rand() % 20;
```

0	1	2	3	4	5	6	7	8	9
11	12	14	9	6	11	6	18	9	10

Przykład: operacje na wektorze

```
/* wyświetlenie elementów tablicy */  
  
printf("Elementy tablicy:\n");  
for (i=0; i<N; i++)  
    printf("%d  ", tab[i]);  
printf("\n");
```

Elementy tablicy:

11 12 14 9 6 11 6 18 9 10

0	1	2	3	4	5	6	7	8	9
11	12	14	9	6	11	6	18	9	10

N = 10

Przykład: operacje na wektorze

```
/* wyświetlenie elementów w odwrotnej kolejności */  
  
printf("Elementy w odwrotnej kolejności:\n");  
for (i=N-1; i>=0; i--)  
    printf("%d  ", tab[i]);  
printf("\n");
```

```
Elementy w odwrotnej kolejności:  
10  9  18  6  11  6  9  14  12  11
```

0	1	2	3	4	5	6	7	8	9
11	12	14	9	6	11	6	18	9	10

N = 10

Przykład: operacje na wektorze

```
/* wyszukanie elementu o najmniejszej wartości */  
  
int min;  
  
min = tab[0];  
for (i=1; i<N; i++)  
    if (tab[i]<min)  
        min = tab[i];  
printf("Wartosc elementu najmniejszego: %d\n",min);
```

Wartosc elementu najmniejszego: 6

0	1	2	3	4	5	6	7	8	9
11	12	14	9	6	11	6	18	9	10

N = 10

Przykład: operacje na wektorze

```
/* indeksy elementów o najmniejszej wartości */  
  
printf("Indeksy elementu najmniejszego: ");  
for (i=0; i<N; i++)  
    if (tab[i]==min)  
        printf("%d ", i);  
printf("\n");
```

Indeksy elementu najmniejszego: 4 6

0	1	2	3	4	5	6	7	8	9
11	12	14	9	6	11	6	18	9	10

N = 10

Przykład: operacje na wektorze

```
/* suma i średnia arytmetyczna elementów tablicy */  
  
int suma = 0;  
float srednia;  
  
for (i=0; i<N; i++)  
    suma = suma + tab[i];  
srednia = (float) suma/N;  
printf("Suma: %d, srednia: %g\n", suma, srednia);
```

Suma: 106, srednia: 10.6

0	1	2	3	4	5	6	7	8	9
11	12	14	9	6	11	6	18	9	10

N = 10

Przykład: operacje na wektorze

```
/* liczba parzystych elementów tablicy */  
  
int ile = 0;  
  
for (i=0; i<N; i++)  
    if (tab[i]%2==0)  
        ile++;  
printf("Liczba parzystych elementów: %d\n",ile);
```

Liczba parzystych elementów: 6

0	1	2	3	4	5	6	7	8	9
11	12	14	9	6	11	6	18	9	10

N = 10

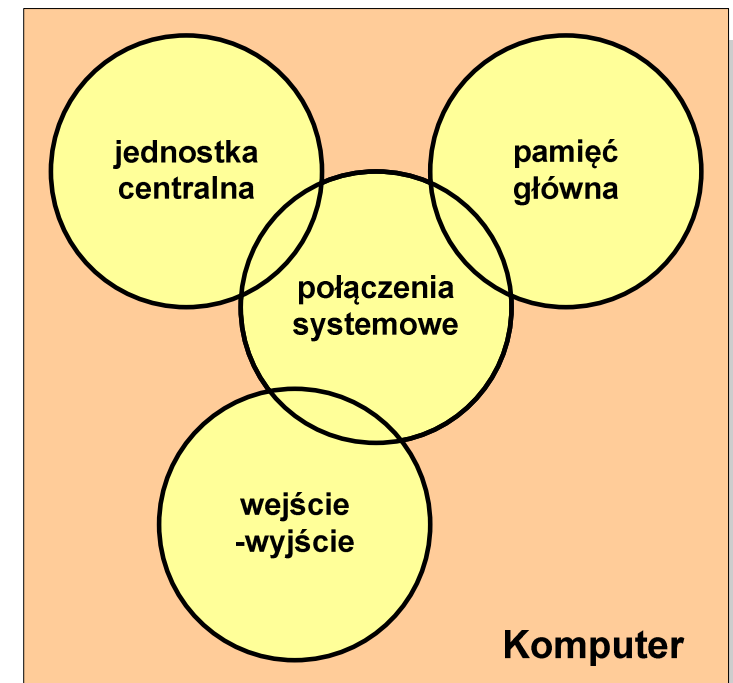
Przykład: operacje na wektorze

```
/* sortowanie elementów tablicy */  
  
int tmp, j;  
  
for (i=0; i<N-1; i++)  
    for (j=i+1; j<N; j++)  
        if (tab[i]>tab[j])  
        {  
            tmp = tab[i];  
            tab[i] = tab[j];  
            tab[j] = tmp;  
        }
```



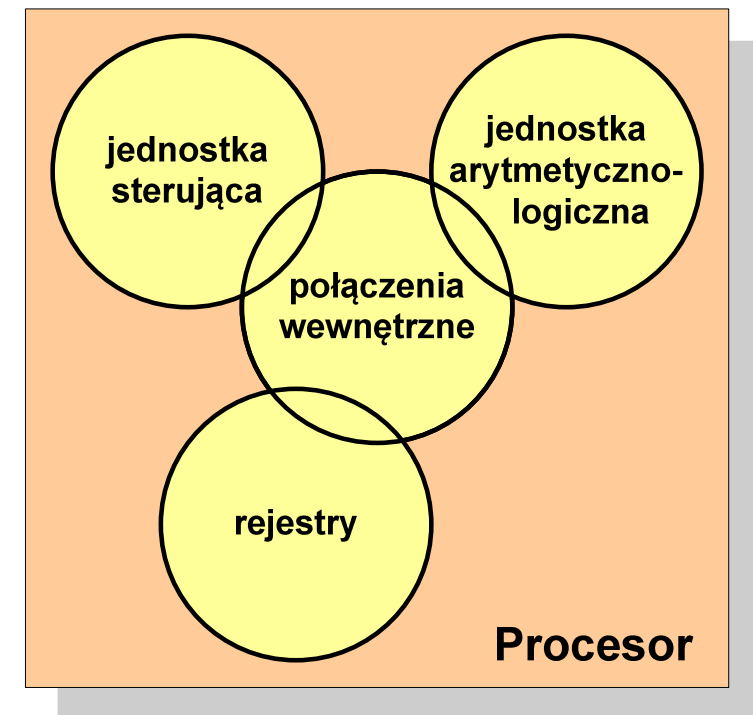
Ogólna struktura systemu komputerowego

- Komputer tworzą cztery główne składniki:
 - **procesor** (jednostka centralna, CPU)
- steruje działaniem komputera
i realizuje przetwarzanie danych
 - **pamięć główna** - przechowuje dane
 - **wejście-wyjście** - przenosi dane
między komputerem a jego
otoczeniem zewnętrznym
 - **połączenia systemu** - mechanizmy
zapewniające komunikację między
składnikami systemu



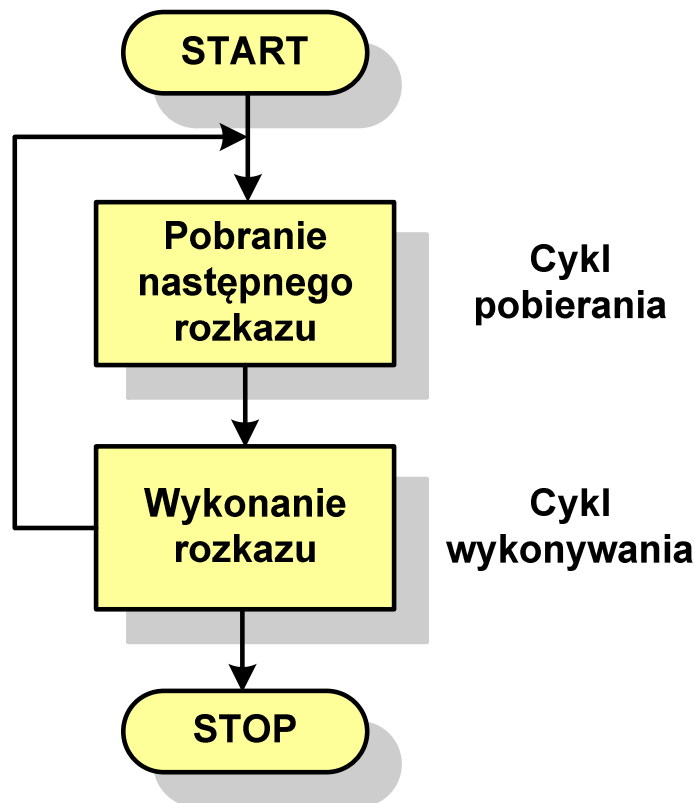
Ogólna struktura procesora

- Główne składniki strukturalne procesora to:
 - **jednostka sterująca** - steruje działaniem procesora i pośrednio całego komputera
 - **jednostka arytmetyczno-logiczna (ALU)** - realizuje przetwarzanie danych przez komputer
 - **rejstry** - realizują wewnętrzne przechowywanie danych w procesorze
 - **połączenia procesora** - wszystkie mechanizmy zapewniające komunikację między jednostką sterującą, ALU i rejestrami.



Działanie komputera

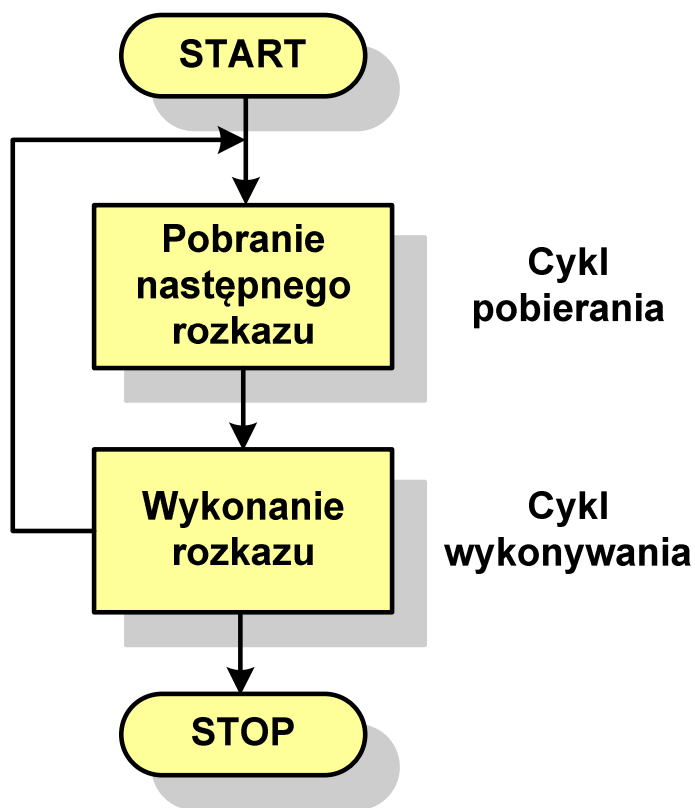
- Podstawowe zadanie komputera to wykonywanie programu
- Program składa się z rozkazów przechowywanych w pamięci
- Rozkazy są przetwarzane w dwu krokach:



- Cykl pobierania (ang. fetch):
 - odczytanie rozkazu z pamięci
 - licznik rozkazów (PC) lub wskaźnik instrukcji (IP) określa, który rozkaz ma być pobrany
 - jeśli procesor nie otrzyma innego polecenia, to inkrementuje licznik PC po każdym pobraniu rozkazu.

Działanie komputera

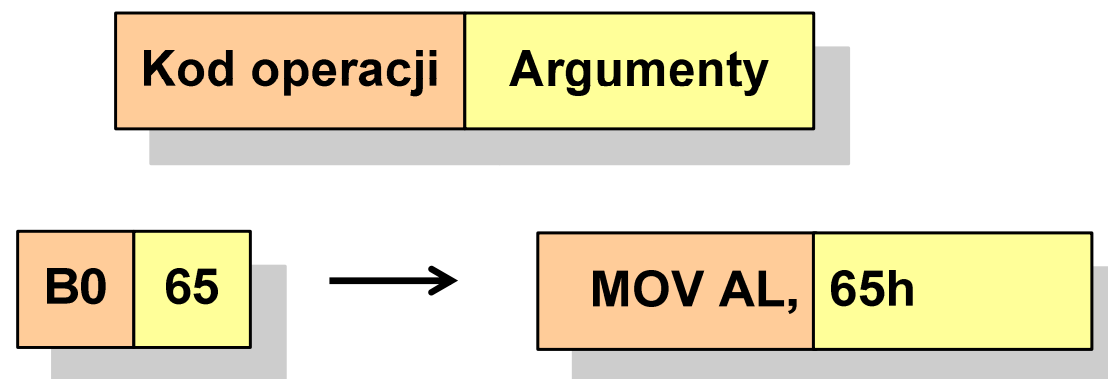
- Podstawowe zadanie komputera to wykonywanie programu
- Program składa się z rozkazów przechowywanych w pamięci
- Rozkazy są przetwarzane w dwu krokach:



- Cykl wykonywania (ang. execution):
 - pobrany rozkaz jest umieszczany w rejestrze rozkazu (IR)
 - rozkaz określa działania, które ma podjąć procesor
 - procesor interpretuje rozkaz i przeprowadza wymagane operacje.

Działanie komputera

- Rozkaz:
 - przechowywany jest w postaci **binarnej**
 - ma określony **format**
 - używa określonego **trybu adresowania**
- **Format** - sposób rozmieszczenia informacji w kodzie rozkazu
- Rozkaz zawiera:
 - **kod operacji** (rodzaj wykonywanej operacji)
 - **argumenty** (lub adresy argumentów) wykonywanych operacji



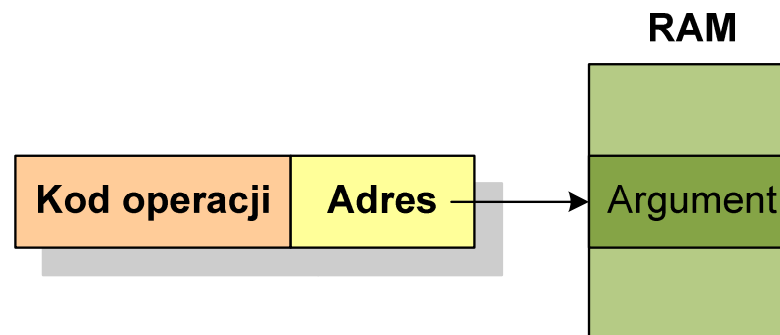
Działanie komputera

- **Tryb adresowania** - sposób określania miejsca przechowywania argumentów rozkazu (operandów)
- Przykładowe rodzaje adresowania:

- **natychmiastowe** - argument znajduje się w kodzie rozkazu



- **bezpośrednie** - kod rozkazu zawiera adres komórki pamięci, w której znajduje się argument



- **rejestrowe** - kod rozkazu zawiera oznaczenie rejestru, w którym znajduje się argument



Program w asemblerze

```
.model SMALL
.286
.stack 100h
.code
    start:
        jmp begin

    handler:
        pusha
        push ds
        pop ds
        popa
        iret

    begin:
        mov ax,0000h
        mov ds,ax
        mov di,0070h
        lea ax,handler
```

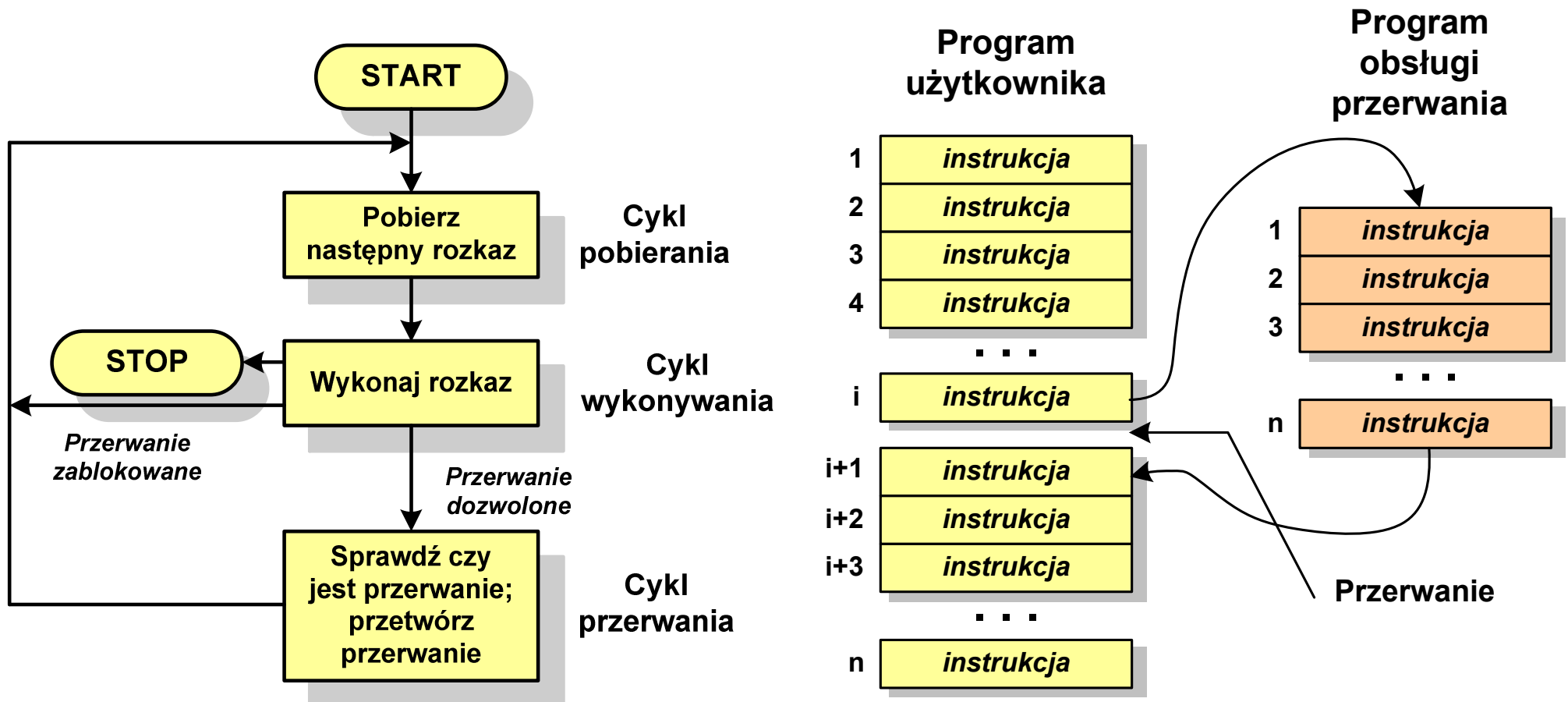
```
cli
mov [di],ax
mov [di+2],cs
sti
mov ax,3100h
mov dx,(offset begin - offset handler)
inc dx
int 21h
end
start
```

Działanie komputera - przerwania

- Wykonywanie kolejnych rozkazów przez procesor może zostać przerwane poprzez wystąpienie tzw. **przerwania** (**interrupt**)
- Przerwanie jest to **sygnał** pochodzący od sprzętu lub oprogramowania informujący procesor o wystąpieniu jakiegoś zdarzenia (np. wciśnięcie klawisza na klawiaturze)
- Bez przerwania procesor musiałby ciągle kontrolować wszystkie urządzenia zewnętrzne, np. klawiatura, port szeregowy
- Każde przerwanie posiada procedurę obsługi przerwania, która jest wykonywana w momencie jego wystąpienia
- Adresy procedur obsługi przerwania zapisane są w tablicy wektorów przerwania

Działanie komputera - przerwania

- Implementacja przerwania wymaga dodania cyklu przerwania do cyklu rozkazu



Rodzaje przerwania

■ Sprzętowe

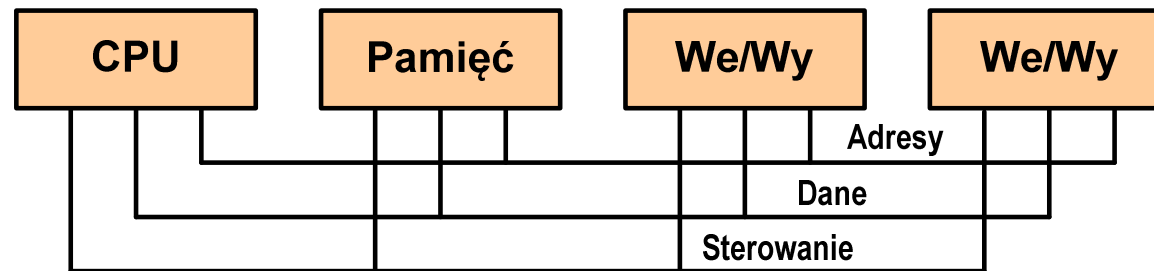
- **zewnętrzne** - sygnały pochodzące z urządzeń zewnętrznych i służące do komunikacji z nimi, np. 08H - zegar, 09h - klawiatura
- **wewnętrzne** - wywoływane przez procesor w celu zasygnalizowania sytuacji wyjątkowych (faults, traps, aborts)

■ Programowe

- instrukcje programu wywołują przerwanie - tym samym wykonywana jest procedura obsługi przerwania
- służą głównie do komunikacji z systemem operacyjnym (DOS - 21h, Windows - 2h, Linux - 80h)

Magistrala

- Najczęściej stosowana struktura połączeń to **magistrala**, składająca się z wielu linii komunikacyjnych, którym przypisane jest określone znaczenie i określona funkcja



- **linie danych (szyna danych)** - przenoszą dane między modułami systemu, liczba linii określa szerokość szyny danych (8, 16, 32, 64 bity)
- **linie adresowe** - służą do określania źródła i miejsca przeznaczenia danych przesyłanych magistralą; liczba linii adresowych określa maksymalną możliwą pojemność pamięci systemu
- **linie sterowania** - służą do sterowania dostępem do linii danych i linii adresowych

Koniec wykładu nr 7

Dziękuję za uwagę!