

# **Technika mikroprocesorowa I**

## **Wykład 1**

# Literatura do bieżącego wykładu:

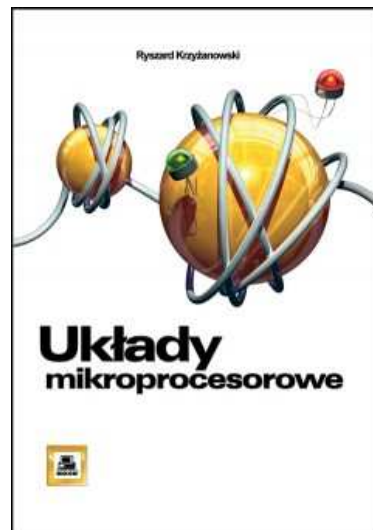
„Układy mikroprocesorowe. Przykłady rozwiązań” Autor: Bartłomiej Zieliński

„Układy mikroprocesorowe” Ryszard Krzyżanowski

„Mikroprocesor Z80” Jerzy Karczmarczuk

„Układy mikroprocesorowe Z80” Fedyna, Mizeracki

[pl.wikipedia.org](http://pl.wikipedia.org)



# **Mikroprocesor- definicje!!!**

*Mikroprocesor to synchroniczny automat sekwencyjny wykonujący dołączony z zewnątrz program.*

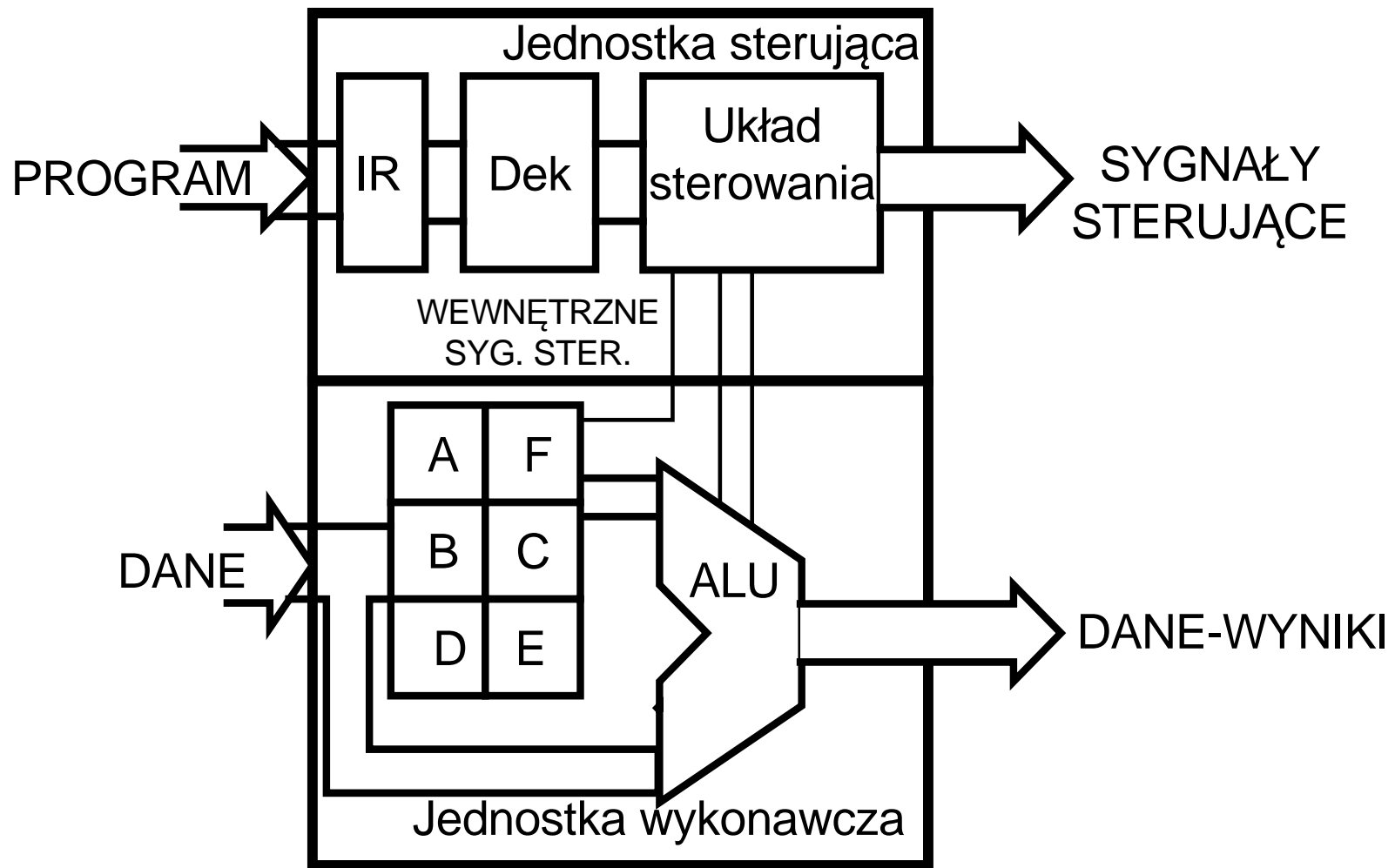
Definicja Wikipedia: Mikroprocesor – *układ cyfrowy wykonany jako pojedynczy układ scalony o wielkim stopniu integracji (LSI) zdolny do wykonywania operacji cyfrowych według dostarczonego ciągu instrukcji.*

Do działania mikroprocesora **niezbędny jest sygnał zegarowy**, który wyznacza szybkości jego pracy.

Program wykonywany przez mikroprocesor składa się z rozkazów (instrukcji).

Lista instrukcji jest stała, charakterystyczna dla danego mikroprocesora, uniwersalna lub zorientowana (np. na przetwarzanie obrazu, obliczenia numeryczne itp.)

# Budowa mikroprocesora



# Budowa mikroprocesora

Podstawowymi podzespołami mikroprocesora są:

- **Jednostka wykonawcza EU** (Execution Unit), która przetwarza informacje wykonując wszelkie operacje arytmetyczne i logiczne.

-**Jednostka sterująca CU** (Control Unit), która określa rodzaj wykonywanych operacji.

*W skład jednostki wykonawczej EU wchodzi:*

-jednostka **arytmetyczno-logiczna ALU**,

-zestaw współpracujących z nią **rejestrów**.

Informacją wejściową części wykonawczej są dane, zaś wyjściową wyniki.

*W skład jednostki sterującej CU wchodzi:*

- **rejestr rozkazów IR,**
- **dekoder rozkazów,**
- **układ sterowania.**

W rejestrze rozkazów przechowywany jest kod aktualnie wykonywanego rozkazu. Kody rozkazów pobierane są do rejestru rozkazów z pamięci. Po pobraniu z pamięci kod rozkazu jest dekodowany w dekodерze rozkazów, czyli jest określone, jaką operację będzie wykonywał mikroprocesor. Na tej podstawie układ sterowania wytwarza odpowiedni sygnał sterujący.

**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).

**Rejestry w procesorze:**

**Akumulator A, ACC** - rejestr bezpośrednio współpracujący z ALU (stanowi źródło i rejestr wynikowy dla operacji arytmetycznych i logicznych)

**Wskaźnik stosu SP** - wskazuje koniec tzw. stosu.

**Licznik rozkazów PC** – wskazuje adres komórki pamięci programu z następnym rozkazem do wykonania

**Rejestr flag** - zawiera flagi (znaczniki bitowe) ustawiane w zależności od wyniku wykonanej operacji (np. nadmiar, zero, bit parzystości)

**Rejestry ogólnego przeznaczenia** – tzw. robocze (służą do przechowywania argumentów, adresów itp.)



# Rejestr znaczników

W technice mikroprocesorowej liczby zapisywane są zasadniczo w dwóch kodach:

-**Naturalnym binarnym (NB)**- liczby bez znaku.

-**Uzupełnień do 2 (U2)**- liczby ze znakiem.

Dla potrzeb działań na **liczbach dziesiętnych** nie ma wydzielonych rozkazów arytmetycznych, ale wprowadzono **rozkazy korekcji dziesiętnej**.

Na podstawie efektów obliczeń układ **ALU** ustawia lub kasuje określone **bity warunkowe w rejestrze znaczników**.

Rozróżnia się następujące bity warunkowe:

**C** – Przeniesienie lub pożyczka, bit dynamiczny, kod NB

**Z**- Zerowość, bit statyczny, kod NB i U2

**N**- Ujemność, bit statyczny, kod U2

**V**- Przepelnienie, bit dynamiczny, kod U2

Ponadto spotyka się jeszcze:

**H**- Przeniesienie połówkowe, kod NB

**P**- Parzystość (ilość jedynek).

Bity statyczne są ustawione bez konieczności przeprowadzania obliczeń.

## **Efektami działania rozkazu mogą być:**

- brak zmiany bitu,
- ustawienie lub skasowanie bitu w zależności od wyniku operacji,
- przyjęcie stałej wartości 0 lub 1,
- nieustalony stan bitu.

**NB**

**U2**

**1111 15**  
**1110 14**  
**1101 13**  
**1100 12**  
**1011 11**  
**1010 10**  
**1001 9**  
**1000 8**  
**0111 7**  
**0110 6**  
**0101 5**  
**0100 4**  
**0011 3**  
**0010 2**  
**0001 1**  
**0000 0**

**0111 7**  
**0110 6**  
**0101 5**  
**0100 4**  
**0011 3**  
**0010 2**  
**0001 1**  
**0000 0**  
**1111 -1**  
**1110 -2**  
**1101 -3**  
**1100 -4**  
**1011 -5**  
**1010 -6**  
**1001 -7**  
**1000 -8**

## **Kod Naturalny Binarny i Kod U2**

# Zamiana liczb dodatnich na liczby ujemne w kodzie U2

$$\begin{array}{r} 0101 \quad 5 \\ 1010 \\ 0001 \\ \hline 1011 \quad -5 \end{array}$$

Negacja  
wszystkich  
bitów  
Dodanie 1

$$\begin{array}{r} 0111 \quad 7 \\ 1000 \\ 0001 \\ \hline 1001 \quad -7 \end{array}$$

Negacja  
wszystkich  
bitów  
Dodanie 1

# Zamiana liczb ujemnych na liczby dodatnie w kodzie U2

$$\begin{array}{r} 1001 \quad -7 \\ 0110 \\ 0001 \\ \hline 0111 \quad 7 \end{array}$$

Negacja  
wszystkich  
bitów  
Dodanie 1

$$\begin{array}{r} 1000 \quad -8 \\ 0111 \\ 0001 \\ \hline 1000 \quad -8 \end{array}$$

Negacja  
wszystkich  
bitów  
Dodanie 1

Nie da się !!!

# Działanie bitów warunkowych

KOD NB

Dodawanie

0110	6
0011	3
<hr/>	
1001	9



**C=0, Z=0**

1011	11
1100	12
<hr/>	
0111	7



**C=1, Z=0**

0101	5
1011	11
<hr/>	
0000	0



**C=1, Z=1**

# Działanie bitów warunkowych

KOD NB

Odejmowanie

W przypadku odejmowania niezależnie od kodu odjemnik jest negowany i zwiększany o 1 po czym następuje dodawanie.

$$\begin{array}{r} \phantom{1} \phantom{0} \phantom{1} \phantom{1} \phantom{0} \phantom{0} \\ - \phantom{1} \phantom{0} \phantom{1} \phantom{1} \phantom{0} \phantom{0} \\ \hline \phantom{1} \phantom{0} \phantom{1} \phantom{1} \phantom{0} \phantom{0} \\ + \phantom{1} \phantom{0} \phantom{1} \phantom{1} \phantom{0} \phantom{0} \\ + \phantom{1} \phantom{0} \phantom{1} \phantom{0} \phantom{1} \phantom{0} \\ \hline 1 \phantom{0} \phantom{1} \phantom{1} \phantom{0} \phantom{0} \phantom{0} \\ \phantom{1} \phantom{0} \phantom{1} \phantom{1} \phantom{0} \phantom{0} \end{array}$$

$C=0, Z=0$

$$\begin{array}{r} \phantom{0} \phantom{1} \phantom{0} \phantom{1} \phantom{0} \phantom{0} \\ - \phantom{0} \phantom{1} \phantom{1} \phantom{1} \phantom{0} \phantom{0} \\ \hline \phantom{0} \phantom{1} \phantom{0} \phantom{1} \phantom{0} \phantom{0} \\ + \phantom{0} \phantom{1} \phantom{0} \phantom{1} \phantom{0} \phantom{0} \\ + \phantom{0} \phantom{0} \phantom{0} \phantom{1} \phantom{0} \phantom{0} \\ \hline 0 \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{0} \phantom{0} \\ \phantom{0} \phantom{1} \phantom{1} \phantom{1} \phantom{0} \phantom{0} \end{array}$$

$C=1, Z=0$



**UWAGA** w odejmowaniu bit C jest tzw. pożyczką, w związku z czym jest ustawiany wg negacji przeniesienia z najstarszego bitu

# Działanie bitów warunkowych

KOD U2

Dodawanie

$$\begin{array}{r} + \quad 0011 \quad 3 \\ \quad 0100 \quad 4 \\ \hline 0111 \quad 7 \end{array}$$

**C=0, N=0, V=0**

$$\begin{array}{r} + \quad 1101 \quad -3 \\ \quad 1100 \quad -4 \\ \hline 1001 \quad -7 \end{array}$$

**C=1, N=1, V=0**

$$\begin{array}{r} + \quad 0110 \quad 6 \\ \quad 0100 \quad 4 \\ \hline 1010 \quad -6 \end{array}$$

**C=0, N=1, V=1**

**Wynik niepoprawny**

# Działanie bitów warunkowych

KOD U2

Odejmowanie

$$\begin{array}{r} \text{—} \quad 1101 \quad -3 \\ \quad 1100 \quad -4 \\ \hline \text{+} \quad 1101 \\ \quad 0100 \\ \hline 0001 \quad 1 \end{array}$$

**C=0, N=0, V=0**

$$\begin{array}{r} \text{—} \quad 0011 \quad 3 \\ \quad 1100 \quad -4 \\ \hline \text{+} \quad 0101 \\ \quad 0100 \\ \hline 0111 \quad 7 \end{array}$$

**C=1, N=0, V=0**

$$\begin{array}{r} \text{—} \quad 0110 \quad 6 \\ \quad 1001 \quad -7 \\ \hline \text{+} \quad 0110 \\ \quad 0111 \\ \hline 1101 \quad -3 \end{array}$$

**C=1, N=1, V=1**

**Wynik niepoprawny**

# Pamięć programu i danych

**Pamięć programu** to element systemu mikroprocesorowego, w którym przechowywane są rozkazy wykonywane przez mikroprocesor i dane stałe. Najczęściej są to pamięci nieulotne typu ROM, EPROM, EEPROM.

**Pamięć danych** to element systemu mikroprocesorowego, w którym przechowywane są dane i wyniki w trakcie działania mikroprocesora. Najczęściej są to pamięci RAM, EEPROM, rzadziej DRAM.

Do adresowania kolejnych komórek pamięci służy magistrala adresowa mikroprocesora, do przesyłania danych, magistrala danych, do zapisu i odczytu pamięci- magistrala sterująca.

# Ważne pojęcia Techniki Mikroprocesorwej

**Stos:** wydzielony obszar pamięci służący do przechowywania danych, adresów powrotów z procedur, adresów powrotów z przerwania. Działa jak kolejka typu **FI-LO (First Input-Last Output)**. Adres wierzchołka stosu pokazuje wskaźnik stosu.

**Przerwanie:** sygnał zewnętrzny, wewnętrzny lub rozkaz, powodujący zarzucenie wykonywania programu po dokończeniu bieżącej instrukcji, zapamiętanie adresu powrotu do programu i przejście do wykonania tzw. procedury obsługi zakończonej odpowiednim rozkazem, powodującym powrót do programu właściwego.

**Ze względu na źródło, przerwania dzielimy na:**

**-sprzętowe:**

**-wewnętrzne (znaczniki),**

**-zewnętrzne (sygnały),**

**-programowe (rozkaz,).**

## **Ze względu na sposób przyjęcia:**

**-maskowalne** (aby zostały przyjęte odpowiedni bit maski musi to umożliwić),

**-niemaskowalne** (przyjmowane zawsze i bezwarunkowo).

**Cykl maszynowy**- pojedynczy cykl dostępu do pamięci lub urządzenia we-wy lub akceptacji przerwania. Składa się z kilku cykli zegarowych.

**Cykl instrukcji**- czas potrzebny do wykonania instrukcji od jej pobrania do wykonania. Składa się z jednego lub kilku cykli maszynowych.

**Instrukcja:** najmniejszy element języka programowania. Po napisaniu w języku najniższego rzędu (assemblerze) i po przetłumaczeniu na kod binarny może być wykonana przez mikroprocesor.

W skład instrukcji wchodzi zawsze **kod operacyjny instrukcji**. Mogą wejść także dodatkowe informacje typu:

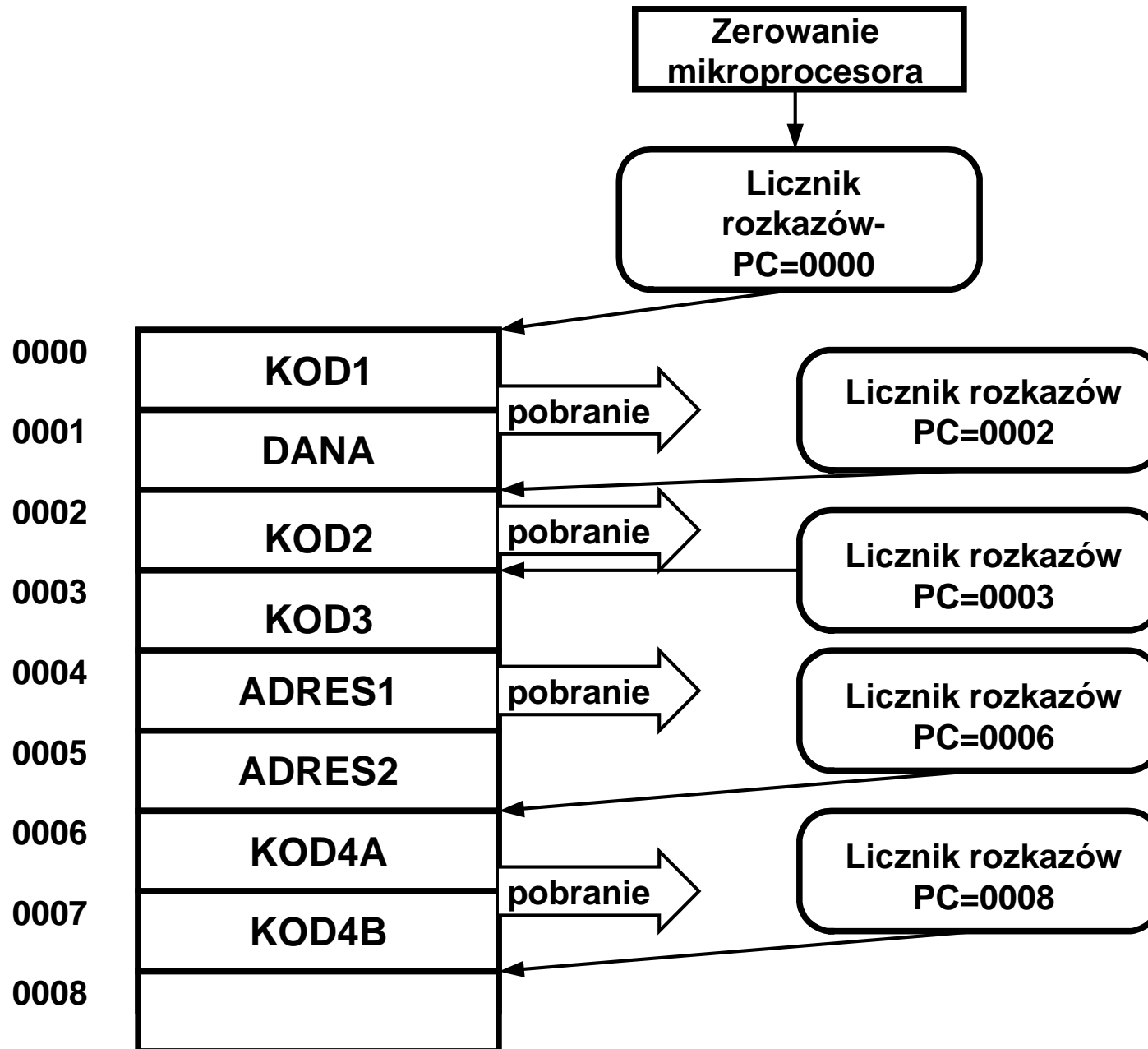
**-dana natychmiastowa,**

**-adres,**

**-dalsza część kodu instrukcji, itp..**

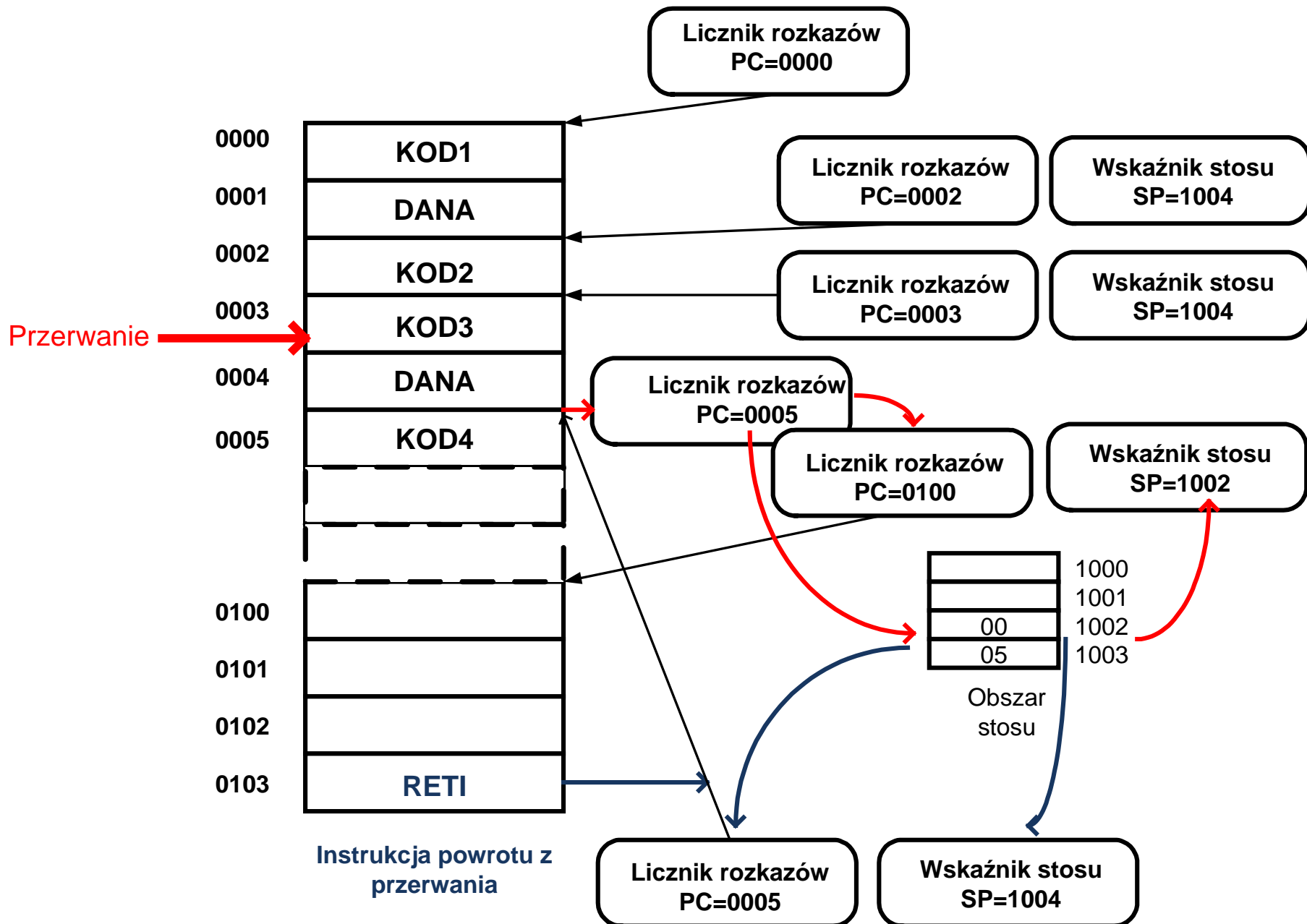
**Program:** jest to ciąg instrukcji wykonywanych przez mikroprocesor.

# Wykonywanie programu przez procesor





# Wykorzystanie stosu- obsługa przerwania



# Tryby adresacji w systemach mikroprocesorowych

**adresowanie natychmiastowe-** argument jest pobierany bezpośrednio z rozkazu,

**adresowanie rejestrowe-** operandy znajdują się w rejestrach wewnętrznych mikroprocesora,

**adresowanie bezpośrednio-** adres operandu znajduje się bezpośrednio w rozkazie,

**adresowanie pośrednie-** adres operandu znajduje się w rejestrze mikroprocesora,

**adresowanie bazowe-** adres rozkazu wskazuje rejestr bazowy,

**adresowanie indeksowe-** adres efektywny jest sumą zawartości adresu bazowego zawartego w rejestrze indeksowym i przesunięcia zawartego w kodzie rozkazu.

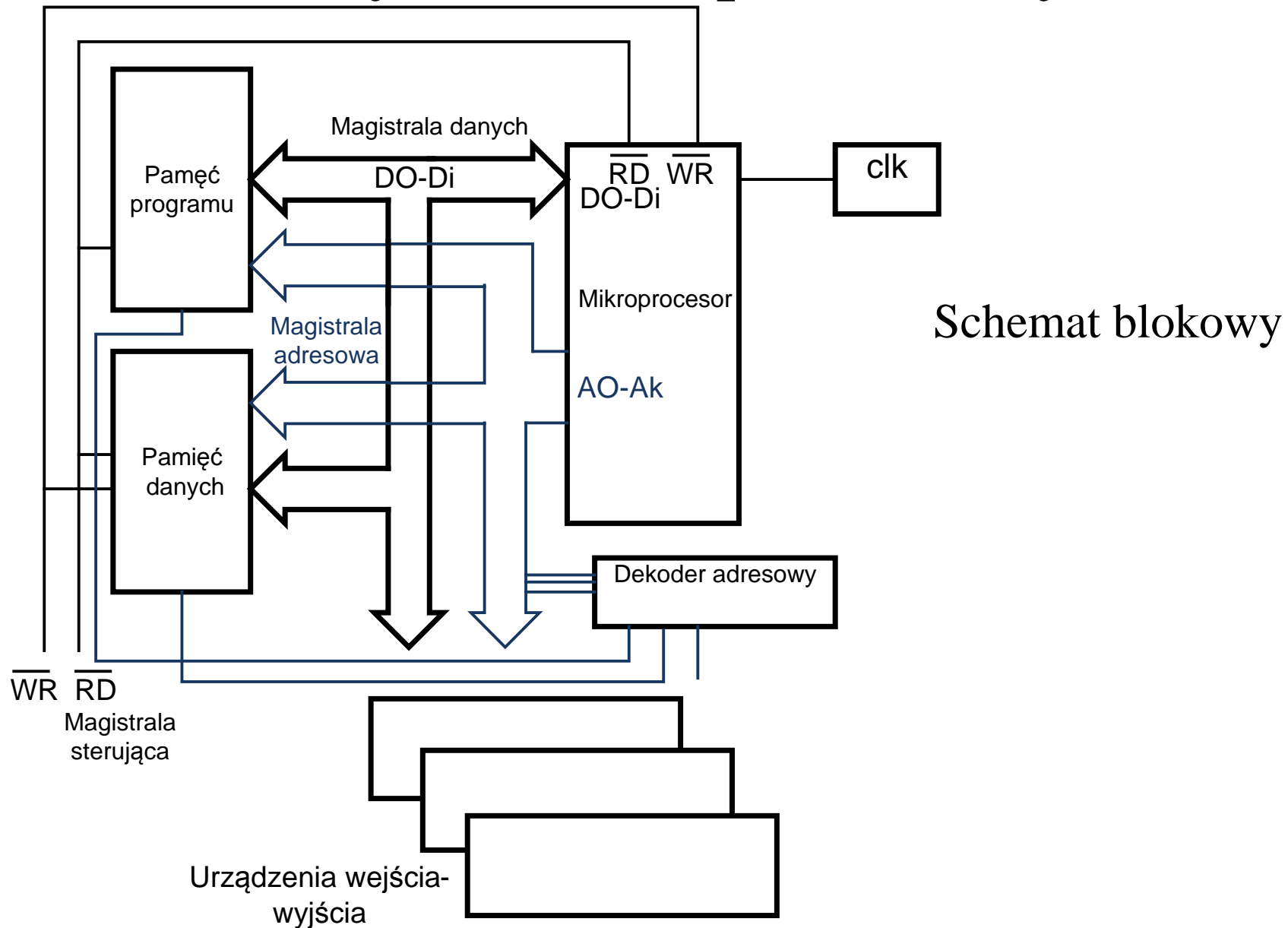
# System mikroprocesorowy

**System mikroprocesorowy:** Układ elektroniczny złożony z mikroprocesora wraz z niezbędnymi układami pamięci programu i danych, układami wejścia-wyjścia (zapewniającymi kontakt z użytkownikiem) oraz niezbędnymi układami sterującymi nosi nazwę systemu mikroprocesorowego.

**Mikrokontroler**– scalony system mikroprocesorowy, zrealizowany w postaci pojedynczego układu scalonego, który zawiera:

- jednostkę centralną (CPU),
- pamięć danych oraz na ogół pamięć programu,
- układy wejścia-wyjścia.

# System mikroprocesorowy



# Magistrale systemu mikroprocesorowego

**Magistrala danych-** służy do przesyłania danych, wyników oraz kodów instrukcji. Jest to magistrala dwukierunkowa, tzn. informacje zarówno wpływają do mikroprocesora, jak i są przez niego wysyłane do innych układów.

**Magistrala adresowa-** służy do adresowania komórek pamięci lub układów wejścia/wyjścia, z którymi chce się komunikować mikroprocesor. Jest to magistrala jednokierunkowa, tzn. adresy są generowane tylko przez mikroprocesor.

**Magistrala sterująca-** służy do sterowania pracą układów współpracujących z mikroprocesorem, sygnalizowania kierunku przesyłu danych oraz sygnalizowanie pewnych określonych stanów układów współpracujących.

# Architektury mikroprocesorów

*Architektura CISC* – ang. Complex Instruction Set Computers

*Architektura RISC* – ang. Reduced Instruction Set Computers

*Architektury:*

– **Von Neumanna**

– **Harvardzka**

– **Harvardzka zmodyfikowana**

## **Cechy architektury RISC:**

- 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- 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, które są następnie wykonywane przez RISC-owy blok wykonawczy.

## Cechy architektury CISC:

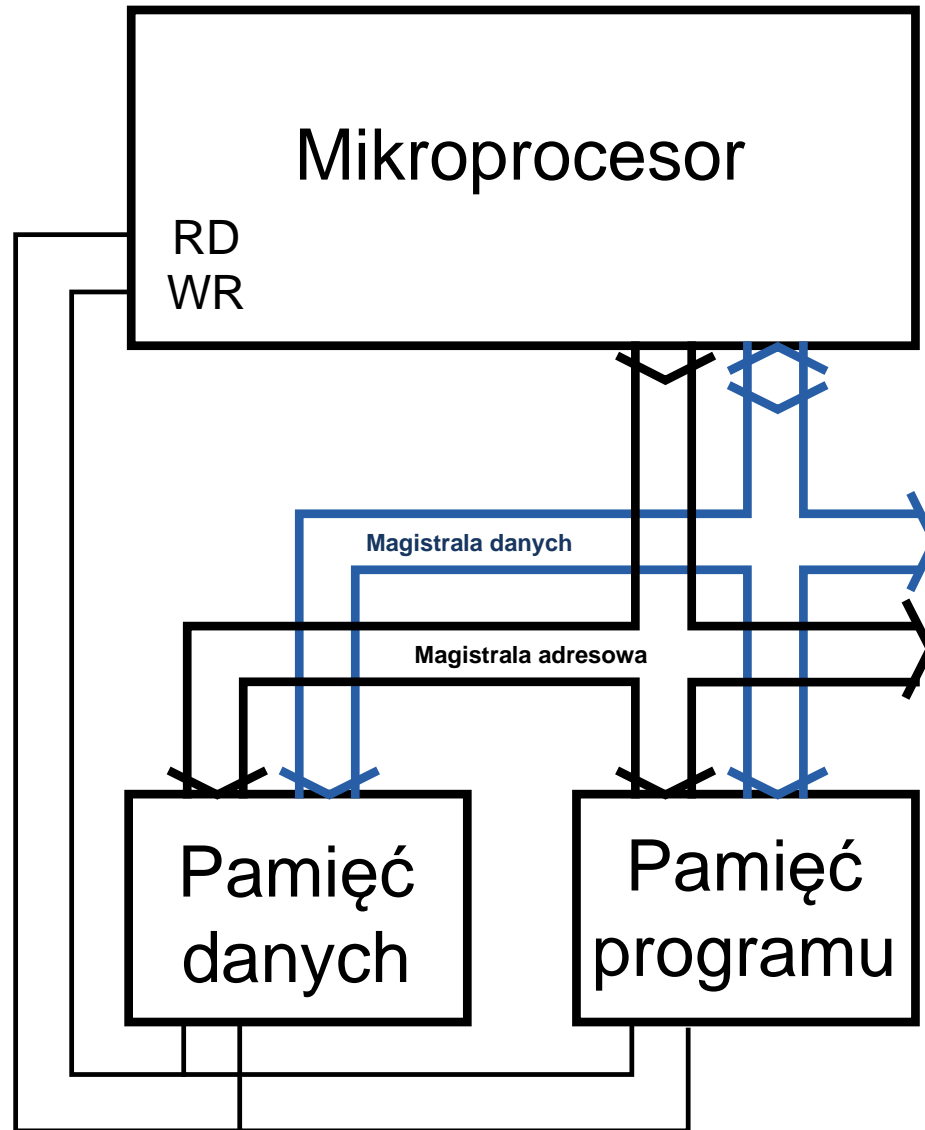
- występowanie złożonych, specjalistycznych rozkazów (instrukcji), które wymagają od kilku do kilkunastu cykli maszynowych (zmienna liczba cykli),
- szeroka gama trybów adresowania (skomplikowana konstrukcja dekodatorów adresu),
- stosunkowo długa lista rozkazów procesora.

### 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.



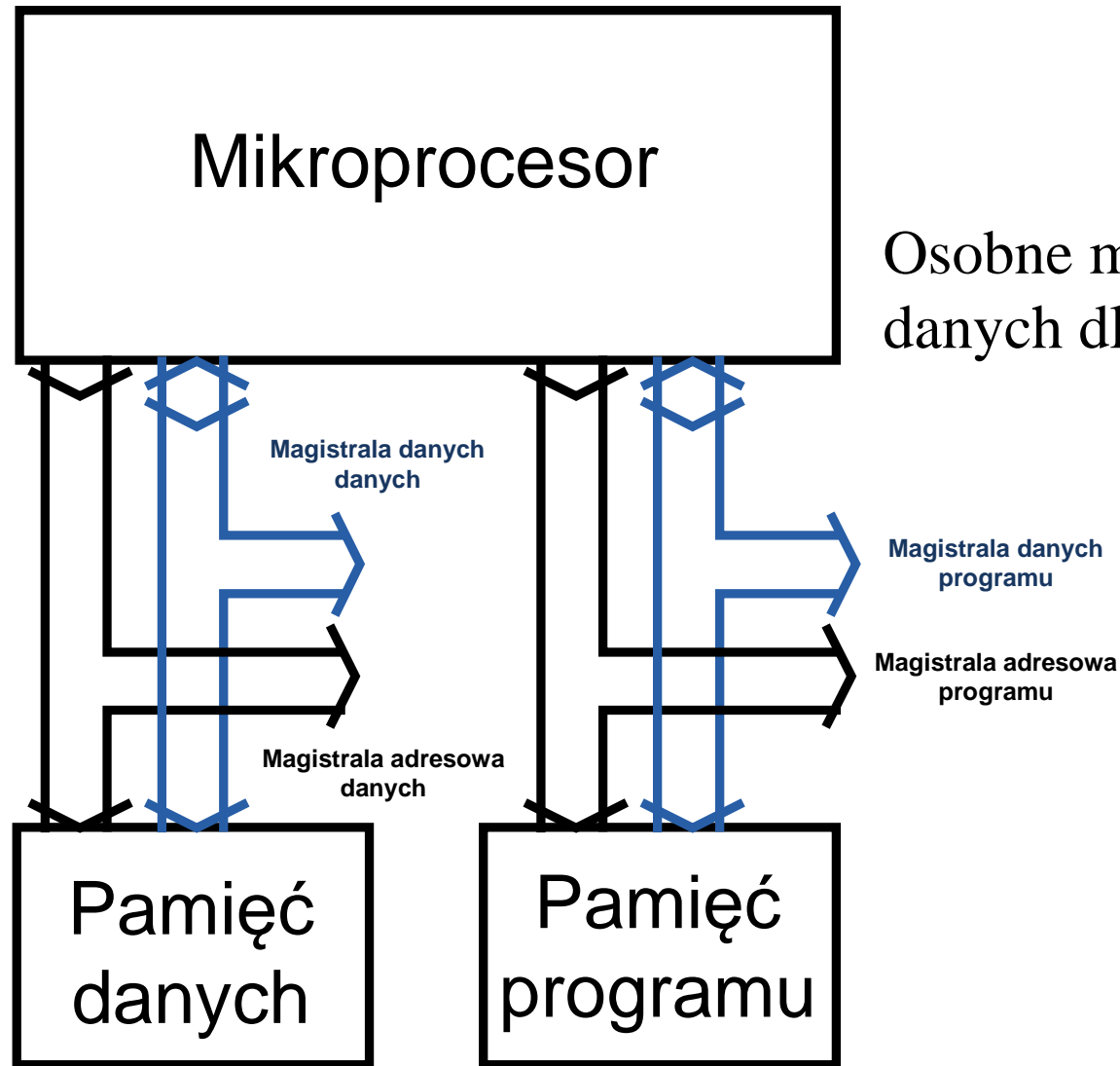
# Architektura Von Neumanna



Wspólna przestrzeń adresowa dla pamięci kodu i danych. Wspólne magistrale:

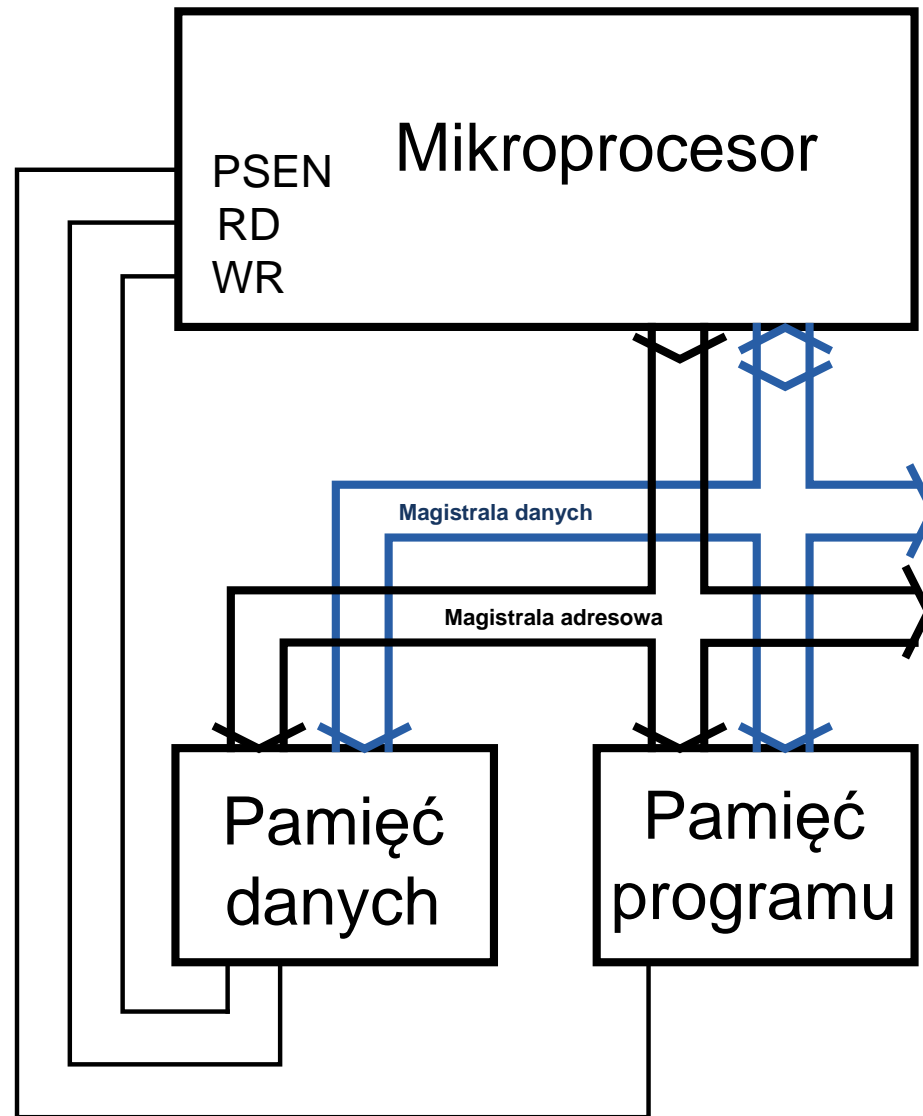
- danych,
- adresowa,
- sterująca.

# Architektura Harvardzka



Osobne magistrale adresowe i danych dla programu i danych.

# Architektura Harvardzka, zmodyfikowana



Wspólna magistrala danych i adresowa, ale pamięci rozdzielone dzięki osobnym sygnałom sterującym.

# Historia Mikroprocesorów

## Intel 4004

Wprowadzony na rynek 15 listopada 1971

Cechny:

- zegar 740 kHz,
- szyna danych: 4-bitowa,
- pamięć adresowalna danych 640 bajtów,
- pamięć programu 4 kilobajty.

**Pierwszy mikroprocesor na świecie używany w kalkulatorach Busicom**

# Intel 4004

Dane techniczne:

- osobna pamięć dla programu i danych (tzw. "architektura harwardzka"),
- dostępne 46 instrukcji,
- 16 czterobitowych rejestrów,
- stos 3-poziomowy.

# Intel 8080

Został wyprodukowany w kwietniu 1974.

Cechy mikroprocesora:

- zegar 2 MHz,
- szyna danych 8-bitowa,
- liczba tranzystorów 6000,
- pamięć jest adresowana 16-bitową szyną adresową,
- jest on uniwersalną jednostką centralną złożoną z jednostki arytmetyczno-logicznej, rejestrów roboczych i układu sterowania,
- słowo 8-bitowe,
- realizuje 72 instrukcje,

- 8 rejestrów programowych dostępnych dla programisty,
- wymagał 3-ech napięć zasilające: +5V, +12V, -5V (włączanych w określonej kolejności),
- posiadał ubogi zestaw trybów adresowania,
- istniała konieczność stosowania dodatkowych układów: zegar i sterownik magistrali.

# Mikroprocesor Z80

Firma **Zilog** została założona przez byłych pracowników firmy Intel. Opracowali oni projekt mikroprocesora opartego na Intel 8080. Nowy układ o nazwie Z80 wszedł do sprzedaży w lipcu 1976 roku.

Dużą zasługę w sukcesie nowego mikroprocesora odegrała w Z80 zgodność programowa z 8080 – systemy oparte na Z80 bez większych problemów mogły korzystać istniejącej już bazy oprogramowania dla 8080.

**Zaletami Z80** w porównaniu do 8080 były:

- pojedyncze napięcie zasilające,
- jednofazowy zegar,
- brak konieczności stosowania dodatkowych kontrolerów magistrali,
- rozszerzona lista rozkazów (m.in. rozkazy arytmetyczne 16-to bitowe),

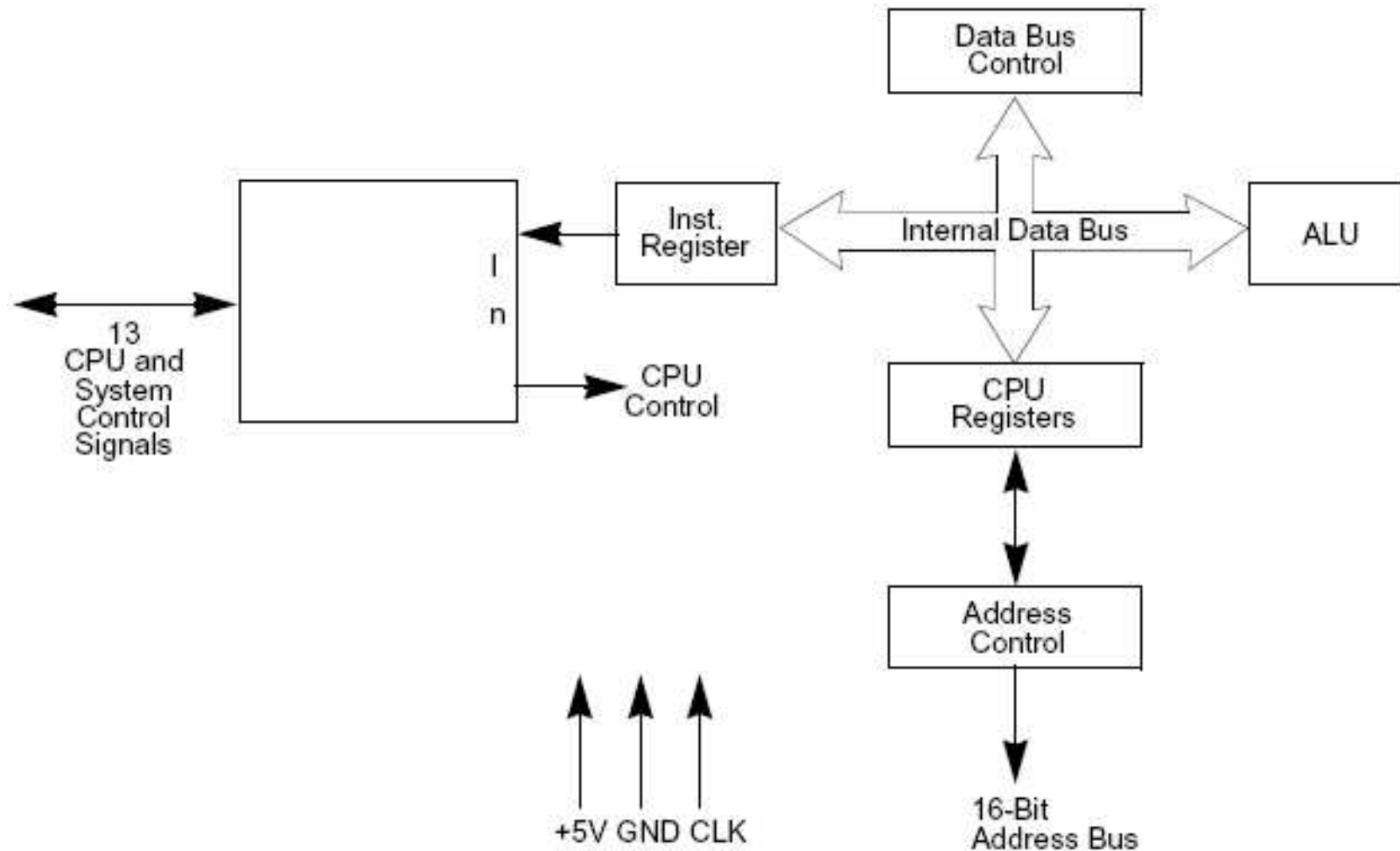


# Cechy mikroprocesora Z80

- 8-bitowa magistrala danych,
- 16-bitowa magistrala adresowa – możliwość zaadresowania 64kB pamięci RAM i obszaru 256B przestrzeni in/out,
- zasilanie i poziomy logiczne zgodne ze standardem TTL (za wyłączeniem zegara taktującego),
- dodatkowe rozkazy (w porównaniu z 8080, m.in. adresacja indeksowa),
- wszystkie sygnały sterujące i obie magistrale dostępne wprost (brak multipleksowania),
- wbudowany układ odświeżania pamięci dynamicznej;
- 158 rozkazów, w tym 78 zgodnych z mikroprocesorem Intel 8080 (pełna wsteczna kompatybilność z 8080);

- zestaw rozkazów operujących na 16-bitowych danych (rejstry można „sklejać” parami);
- zegar od 2MHz do 8 MHz (produkowana w NRD (Republika Demokratyczna Niemiec)- wersja 1MHz,
- duży jak dla procesora 8-bitowego zestaw rejestrów wewnętrznych ogólnego przeznaczenia wraz z zestawem alternatywnych rejestrów.

# Mikroprocesor Z80 (ZILOG)



Schemat blokowy [www.zilog.com]

# Zestaw rejestrów

Main Register Set		Alternate Register Set	
Accumulator	Flags	Accumulator	Flags
A	F	A'	F'
B	C	B'	B'
D	E	D'	E'
H	L	H'	L'

General Purpose Registers

Interrupt Vector I	Memory Refresh R
Index Register	IX
Index Register	IY
Stack Pointer	SP
Program Counter	PC

Special Purpose Registers

**A- akumulator**, rejestr współpracujący z jednostką arytmetyczno-logiczną ALU. Jest źródłem argumentów oraz rejestrem wynikowym przy operacjach arytmetycznych 8-mio bitowych i logicznych. Jest rejestrem uprzywilejowanym pod względem trybu ilości trybów adresacji.

**F- rejestr znaczników**, zawiera bity warunkowe ustawiane lub kasowane w czasie działania programu.

<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
S	Z	X	N	X	P/V	N	C

Symbol	Field Name
C	Carry Flag
N	Add/Subtract
P/V	Parity/Overflow Flag
H	Half Carry Flag
Z	Zero Flag
S	Sign Flag
X	Not Used

[[www.zilog.com](http://www.zilog.com)]

**C**- przeniesienie lub pożyczka,

**N**- ostatnio wykonywaną instrukcją było odejmowanie lub dodawanie (informacja ta jest potrzebna do korekcji dziesiętnej po tych operacjach)

**P/V**- parzystość/przepełnienie

*Po operacjach arytmetycznych* wskazuje czy nastąpiło przepełnienie w przypadku działania na kodzie U2.

*Po operacjach logicznych* wskazuje na parzystą ilość jedynek w akumulatorze.

**H**- przeniesienie połówkowe, informuje o wystąpieniu przeniesienia pomiędzy 3 i 4 bitem bajta (do korekcji dziesiętnej).

**Z**- zerowość, jest ustawiany na jeden jeśli w wyniku operacji arytmetycznej logicznej lub porównania wynik w akumulatorze jest równy 0

**S**- znak, jest kopią 7 bitu akumulatora, oznacza liczbę ujemną w kodzie U2

**Rejestry B i C.** Można połączyć je w parę 16-to bitową BC. Są to rejestry ogólnego przeznaczenia wykorzystywane również jako liczniki operacji blokowych i do tworzenia pętli liczących. Rejestr C służy do adresacji przestrzeni we-wy.

**Rejestry D i E.** Można połączyć je w parę 16-to bitową DE. Są to rejestry ogólnego przeznaczenia.

**Rejestry H i L.** Można połączyć je w parę 16-to bitową HL. Są to rejestry ogólnego przeznaczenia. Ponadto para rejestrów HL jest używana jako rejestr adresowy przestrzeni pamięci oraz jako akumulator przy operacjach arytmetycznych 16-to bitowych.

**Rejestry A' i F'** – rejestry lustrzane do A i F. Istnieje możliwość wymiany zawartości tych rejestrów np. na czas wykonywania obsługi przerwania.

**Rejestry B', C', D', E' H',L'**- rejestry lustrzane do B,C,D,E,H,L. Istnieje możliwość wymiany zawartości tych rejestrów np. na czas wykonywania obsługi przerwania.

**Rejestry IX, IY-** rejestry 16-to bitowe indeksowe, służące do adresacji obszaru pamięci. W rejestrach tych znajduje się adres bazowy. Przy adresacji indeksowej, adres efektywny jest sumą adresu bazowego zawartego w Ix lub IY i 8-mio bitowego przesunięcia w kodzie U2, podanego za kodem rozkazu przesłania. Kody operacyjne instrukcji wykorzystujących rejestry indeksowe są dwubajtowe.

**Rejestr SP-** 16-to bitowy wskaźnik stosu. Przy składaniu na stos wskaźnik jest zmniejszany a następnie następuje zapis na stos. Po rozpoczęciu programu należy go ustawić w zależności od dysponowanej pamięci.

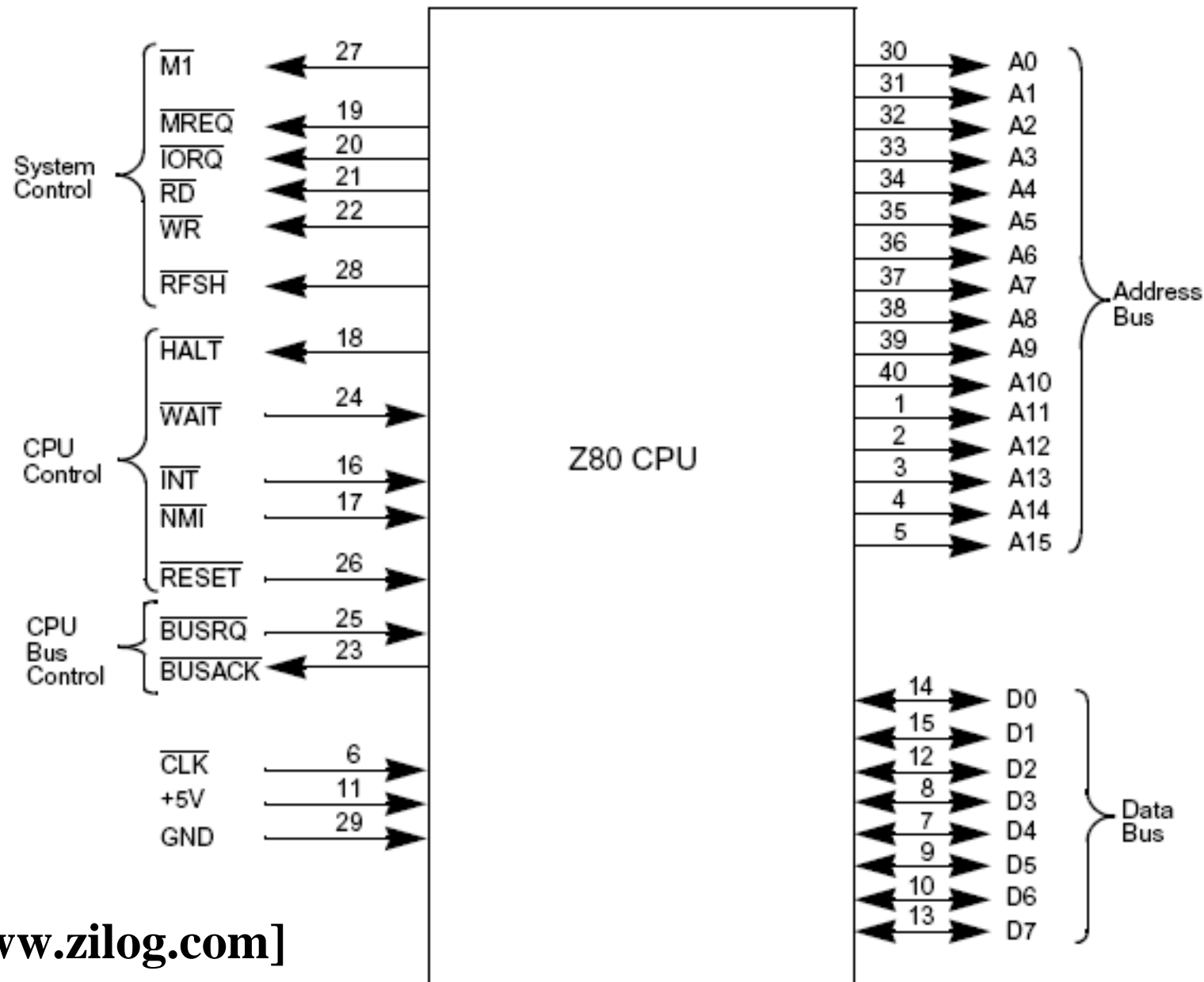
**Rejestr PC-** 16 bitowy licznik rozkazów, po RESET ustawiany na 0000. Wskazuje adres następnego rozkazu do odczytu.

**Rejestr R-** rejestr odświeżania zawiera cyklicznie zwiększany adres (osiem młodszych bitów magistrali adresowej) do odświeżania pamięci dynamicznych.

**Rejestr I-** rejestr adresu tablicy przerwań



# Wprowadzenia mikroprocesora Z80



# Wyprowadzenia mikroprocesora Z80

**A0-A15 (wy)**- 16-to bitowa magistrala adresowa umożliwiająca zaadresowanie 64-kB pamięci oraz 256B urządzeń we-wy.

**D0-D7 (we-wy)**- 8-mio bitowa magistrala danych.

**RD (wy)**- strob odczytu pamięci lub urządzenia wejścia-wyjścia (aktywny stan niski)

**WR (wy)**- strob zapisu do pamięci lub urządzenia wejścia-wyjścia (aktywny stan niski)

**MREQ (wy)**- informacja iż bieżący cykl jest związany z dostępem do pamięci (aktywny stan niski)

**IORQ (wy)**- informacja iż bieżący cykl jest związany z dostępem do urządzenia we-wy (aktywny stan niski)

**M1 (wy)**- informacja, że bieżący cykl jest pierwszym cyklem maszynowym cyklu rozkazowego (pobranie pierwszego bajtu KO z pamięci) (aktywny stan niski)

# Wyprowadzenia mikroprocesora Z80

**IORQ + M1 (wy)**- cykl akceptacji przerwania (informuje o przyjęciu przerwania maskowalnego)

**WAIT (we)**- sygnał niegotowości, wykorzystywany do wydłużania cyklu magistrali przy dostępie do pamięci lub urządzenia we-wy (aktywny stanem niskim). Sygnał jest testowany opadającym zboczem w takcie T2 zegara w czasie każdego cyklu maszynowego.

**HALT (wy)**- sygnał informujący iż mikroprocesor jest w trakcie wykonywania instrukcji HALT (aktywny stan niskim)

**RFSH (wy)**- sygnał informujący iż wystawiony adres dotyczy odświeżania pamięci dynamicznej (aktywny stanem niskim)

**RESET (we)**- wejście zerowania mikroprocesora, powoduje wyzerowanie licznika rozkazów i przygotowanie do pracy (aktywny stanem niskim).

# Wyprowadzenia mikroprocesora Z80

**CLK (we)**- wejście sygnału zegarowego

**BUSRQ (we)**- wejście żądania zwolnienia magistrali (aktywne stanem niskim). Testowane w każdym cyklu maszynowym. Jeżeli pod koniec bieżącego cyklu maszynowego mikroprocesor wykryje stan niski na tym wejściu, kończy cykl (nie kończy rozkazu) na liniach danych, adresowych i sterujących pojawia się stan wysokiej impedancji. Wysterowuje linię BUSACK. Po wycofaniu sygnału procesor dokończy bieżący cykl rozkazowy.

**BUSACK (wy)**- informuje iż procesor oddał magistralę.