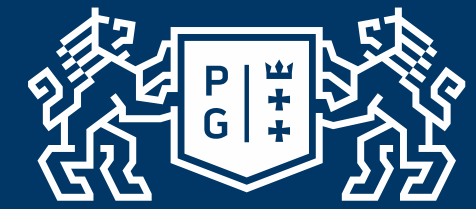


Architektura komputerów

sem. zimowy 2024/25

cz. 1

Tomasz Dziubich

POLITECHNIKA GDAŃSKA

Sprawy organizacyjne

POLITECHNIKA GDAŃSKA

Zasady zaliczenia przedmiotu (1)

- W trakcie semestru można uzyskać łącznie 100 pkt.:
  - ćwiczenia tablicowe 25 pkt.
  - ćwiczenia laboratoryjne 25 pkt.
  - egzamin – część teoretyczna 25 pkt.
  - egzamin – część zadaniowa 25 pkt.
- Próg zaliczenia wynosi 50 pkt. (ocena dostateczna).
- Warunkiem koniecznym przystąpienia do egzaminu jest uzyskanie co najmniej 8 punktów z ćwiczeń.
- Uzyskanie 18 lub więcej punktów z ćwiczeń tablicowych powoduje zwolnienie z części zadaniowej egzaminu (w tym przypadku liczba punktów z części teoretycznej jest podwajana).

POLITECHNIKA GDAŃSKA

Zasady zaliczenia przedmiotu (2)

Przedział punktowy	Ocena końcowa
<0; 50)	2
<50 ; 61)	3
<61 ; 71)	3,5
<71 ; 81)	4
<81; 90)	4,5
<91; 100>	5

POLITECHNIKA GDAŃSKA

Literatura (1)

- **Patterson D.A., Hennessy J.L. : Computer Organization and Design, 5th Edition, 2014, wyd. Morgan Kaufmann.**
- **Hennessy J.L., Patterson D.A.: Computer Architecture, 6th Edition, 2017, Morgan Kaufmann.**
- **Metzger P.: Anatomia PC. Wyd. Helion 2007.**
- Petzold C.: Kod. Ukryty język sprzętu komputerowego i oprogramowania. Wyd. Helion 2002.
- Chalk B.S.: Organizacja i architektura komputerów. Warszawa WNT 1998

Pojęcie – architektura komputera

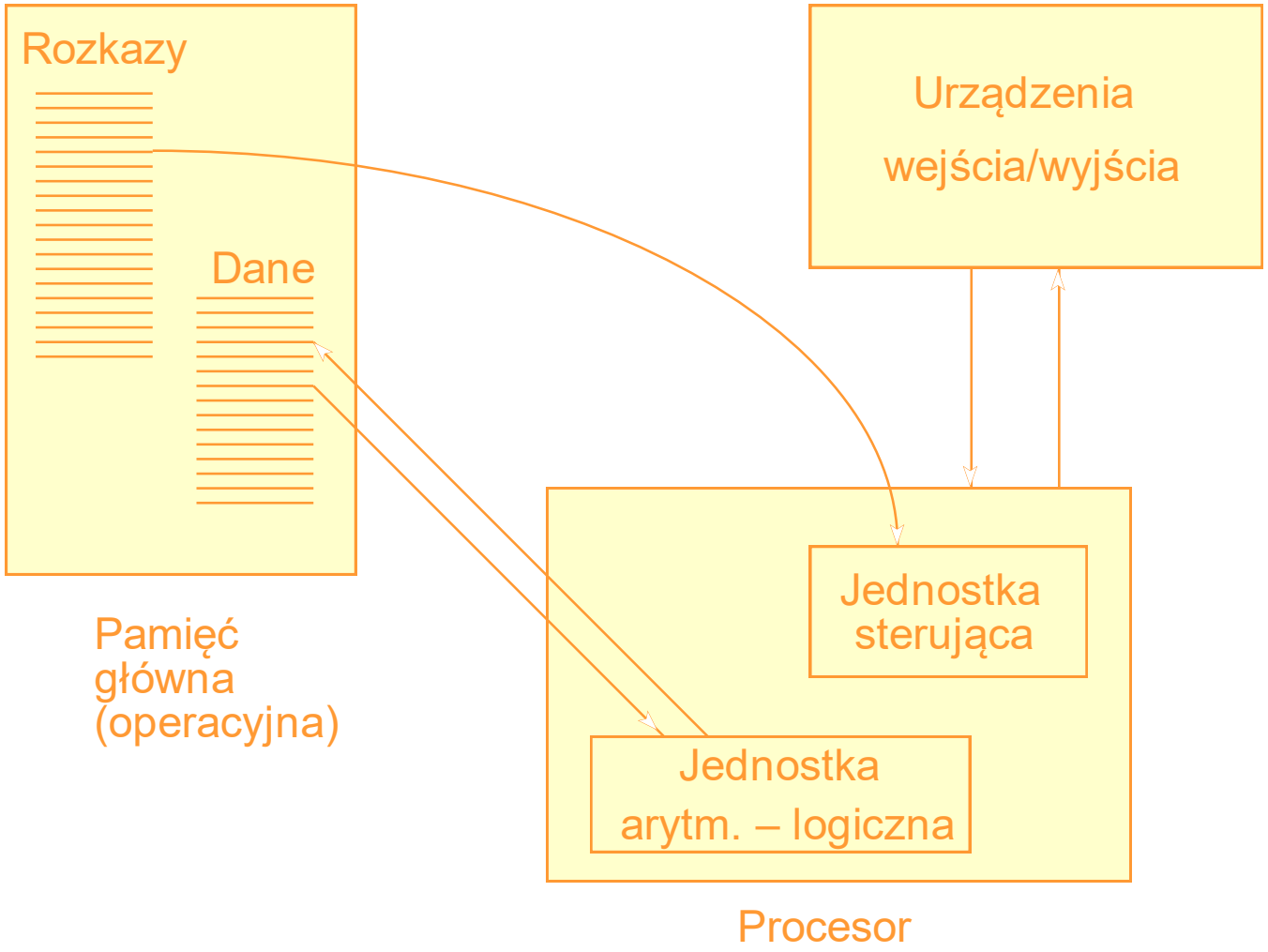
POLITECHNIKA GDAŃSKA

Rozwój konstrukcji komputerów i oprogramowania (1)

- Kalkulatory elektromechaniczne: Bell 1930, Zuse 1941 (arytmetyka zmiennoprzecinkowa)
- 1942 – 1946 pierwsze komputery elektroniczne
- 1945 koncepcje von Neumanna
- 1948 Opracowanie tranzystora
- 1949 Rozwój oprogramowania: biblioteki podprogramów, asembler
- 1951 Komputer EDVAC (von Neumann) — program przechowywany w pamięci

POLITECHNIKA GDAŃSKA

Model komputera wg von Neumanna (1)



POLITECHNIKA GDAŃSKA

Model komputera wg von Neumanna (2)

- Procesor steruje podstawowymi operacjami komputera, wykonuje operacje arytmetyczne i logiczne, przesyła i odbiera sygnały, adresy i dane z jednego podzespołu komputera do drugiego.
- Procesor pobiera kolejne instrukcje (rozkazy) programu i dane z pamięci głównej (operacyjnej) komputera, przetwarza je i ewentualnie odsyła wyniki do pamięci. W każdej chwili procesor wykonuje dokładnie jedną instrukcję (rozkaz).
- Sposób przechowywania danych i instrukcji jest identyczny — **zapisany kod nie pozwala odróżnić instrukcji od danych**.

POLITECHNIKA GDAŃSKA

Model komputera wg von Neumanna (3)

- Mimo upływu wielu lat prawie wszystkie współczesne komputery ogólnego przeznaczenia stanowią realizację tego modelu.
- Zasadniczą i centralną część każdego komputera stanowi procesor — jego własności decydują o pracy całego komputera.
- Komunikacja ze światem zewnętrznym realizowana jest za pomocą urządzeń wejścia/wyjścia.

POLITECHNIKA GDAŃSKA

Rozwój konstrukcji komputerów i oprogramowania (2)

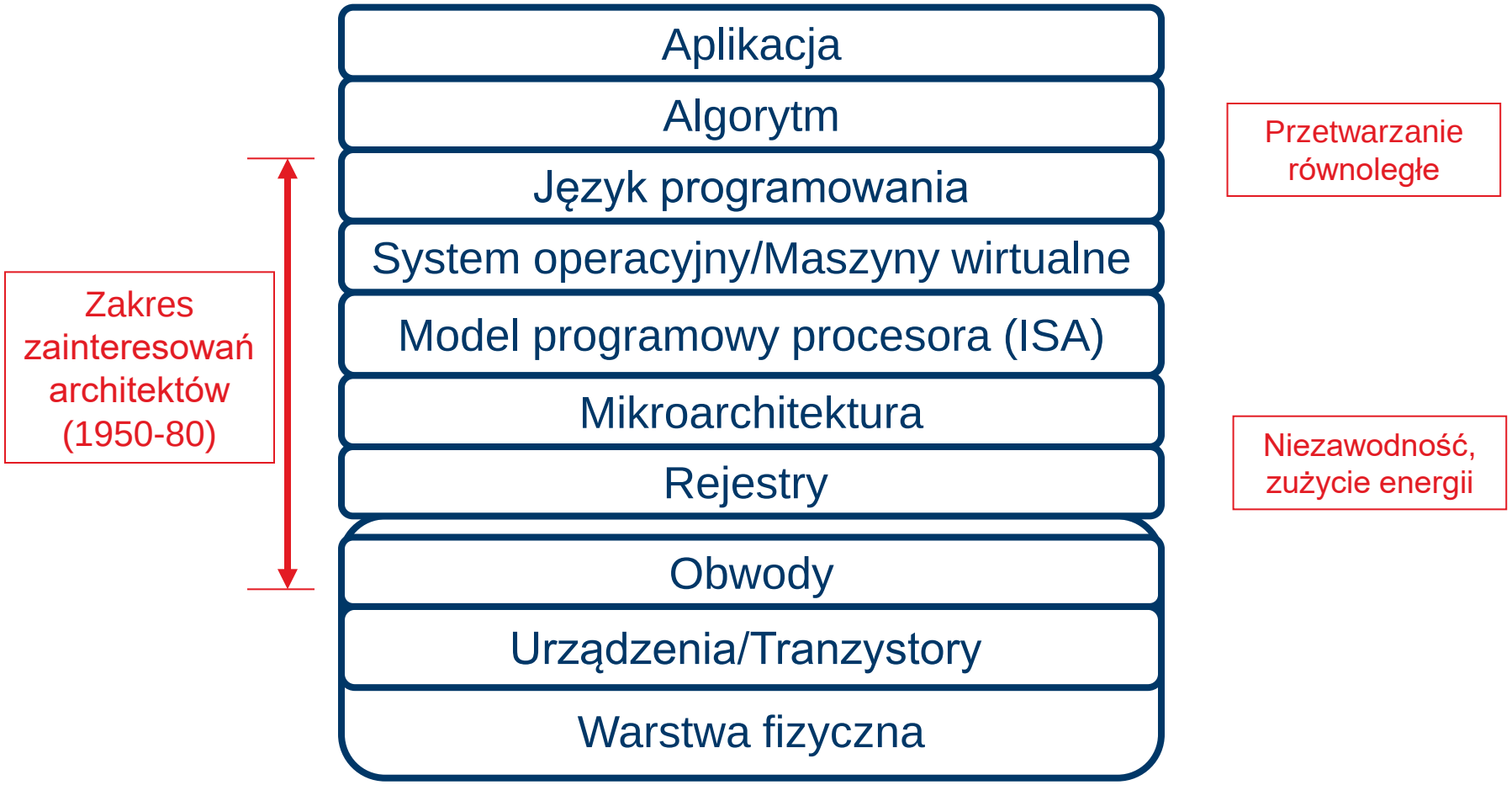
- 1954 język programowania FORTRAN
- 1955 pierwszy komputer tranzystorowy
- 1959 komputer PDP–1
- 1969 System Unix
- lata 70 minikomputery
- 1972 język C
- 1981 komputer IBM PC (16 KB RAM)
- 1990 system Windows 3.0
- 2004 procesory wielordzeniowe
- 2007-2024 procesory Ryzen, ThreadRipper (mikroarchitektura Zen, AMD), Core 14 gen.(mikroarchitektura Raptor, Intel)

Architektura komputerów – definicja

- Architektura komputerów jest projektem warstw abstrakcji, dzięki którym mamy możliwość implementowania aplikacji do efektywnego przetwarzania informacji, wykorzystując dostępne technologie. [Computer.Organization.and.Design.The.Hardware.Software.Interface, Patterson, Hennessy, 2005]

POLITECHNIKA GDAŃSKA

Warstwy abstrakcji w nowoczesnych systemach komputerowych



POLITECHNIKA GDAŃSKA

Architektura i jej atrybuty

- Architektura systemu opisywana jest przez zestaw ***atrybutów*** (interfejsów). Są one definiowane z punktu widzenia programisty i mają bezpośredni wpływ na logiczne wykonanie programu.
- Zbiór tych atrybutów określa się jako ***model programowy*** *procesora* – tzn. architekturę na poziomie zbioru instrukcji - ISA (Instruction Set Architecture).
- Procesory posiadające ten sam model programowy są ze sobą kompatybilne, co oznacza, że mogą wykonywać te same programy i generować te same rezultaty.
- Procesory mogą posiadać identyczne atrybuty (identyczny model programowy), ale mogą być zbudowane w odmienny sposób (mają inną ***organizację***)

POLITECHNIKA GDAŃSKA

Atrybuty architektury procesora

- Dostępne operacje (zbiory instrukcji)
- Dostępne rejestry
- Typy i rozmiar operandów
- Klasy odwołań do pamięci (register – mem, load – store)
- Tryby adresowania (reg, imm, disp)
- Wykorzystywane instrukcje sterowania przepływem
- Sposób kodowanie instrukcji (fixed, var)
- Sposób wyrównywania adresów (alignment)

POLITECHNIKA GDAŃSKA

Podstawowe architektury (modele) systemów

W zależności od sposobu przechowywania danych i rozkazów przez system komputerowy możemy wyróżnić:

- **architekturę von Neumanna** – zarówno dane, jak i programy są przechowywane w tym samym bloku pamięci;
- architekturę harwardzką – rozkazy i dane są przechowywane w oddzielnych pamięciach;
- architekturę mieszaną (hybrydową) – połączenie dwóch powyższych typów: rozdzielono pamięci rozkazów i danych, jednak wykorzystują one wspólne magistrale.

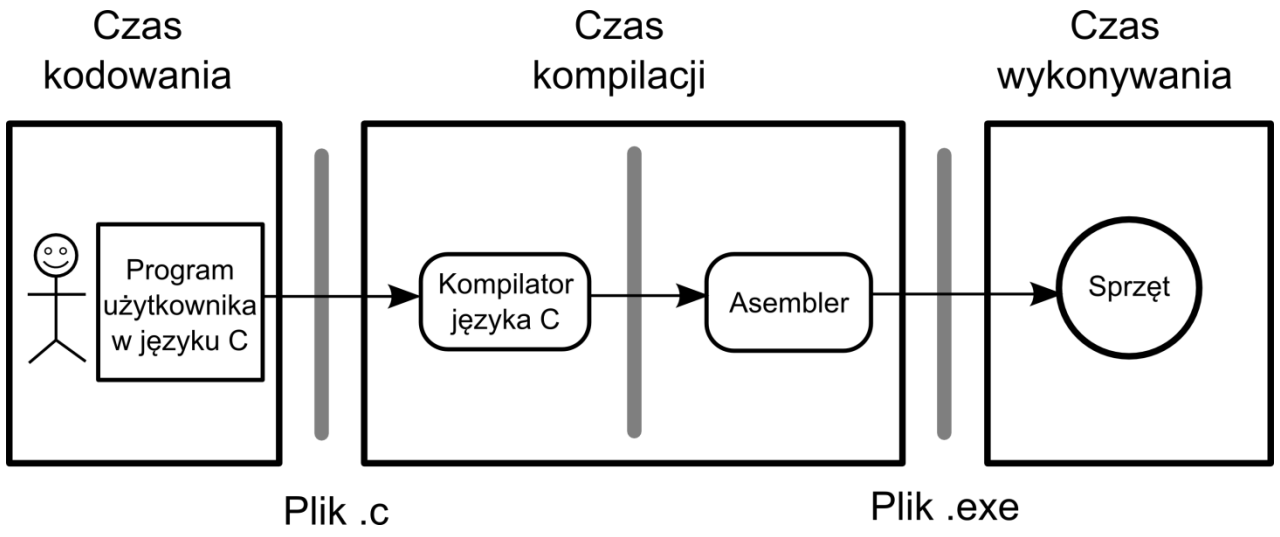
POLITECHNIKA GDAŃSKA

Architektura a organizacja

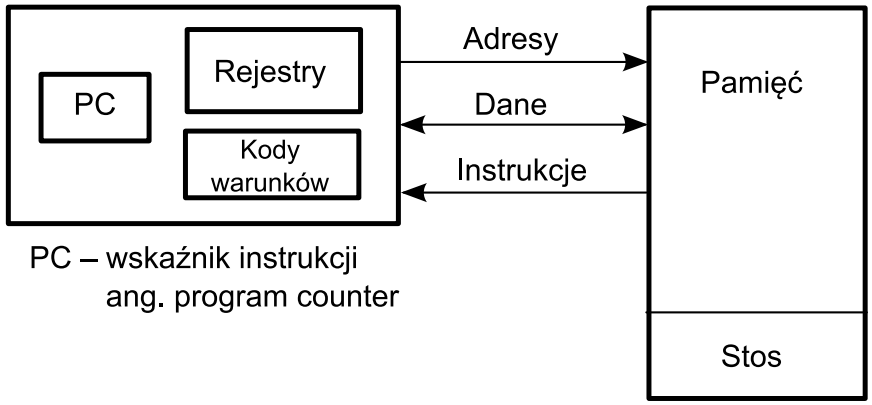
- Organizacja komputerów odnosi się do jednostek przetwarzających (tzw. operacyjnych) i ich wzajemnych połączeń, które realizują przyjęte dla architektury specyfikacje.
- Organizacja komputerów to implementacja architektury.
- Atrybuty związane z organizacją komputera to szczegóły sprzętowe, przezroczyste dla programisty, takie jak sygnały sterujące, interfejsy między komputerem a peryferiami, technologie wykorzystane w konstrukcji pamięci operacyjnej.
- Współczesne procesory są bardzo złożonymi układami cyfrowymi. Zachowują również kompatybilność wsteczną z poprzedzającymi architekturami. W środku mogą zawierać inne mikroprocesory o prostszej architekturze. Szczegóły te w odniesieniu do samego procesora obecnie nazywa się mikroarchitekturą.
- Zatem pełną architekturę procesora opisuje model programowy ISA oraz mikroarchitektura.

POLITECHNIKA GDAŃSKA

Cel przedmiotu: model programowy procesora



Architektura na poziomie instrukcji



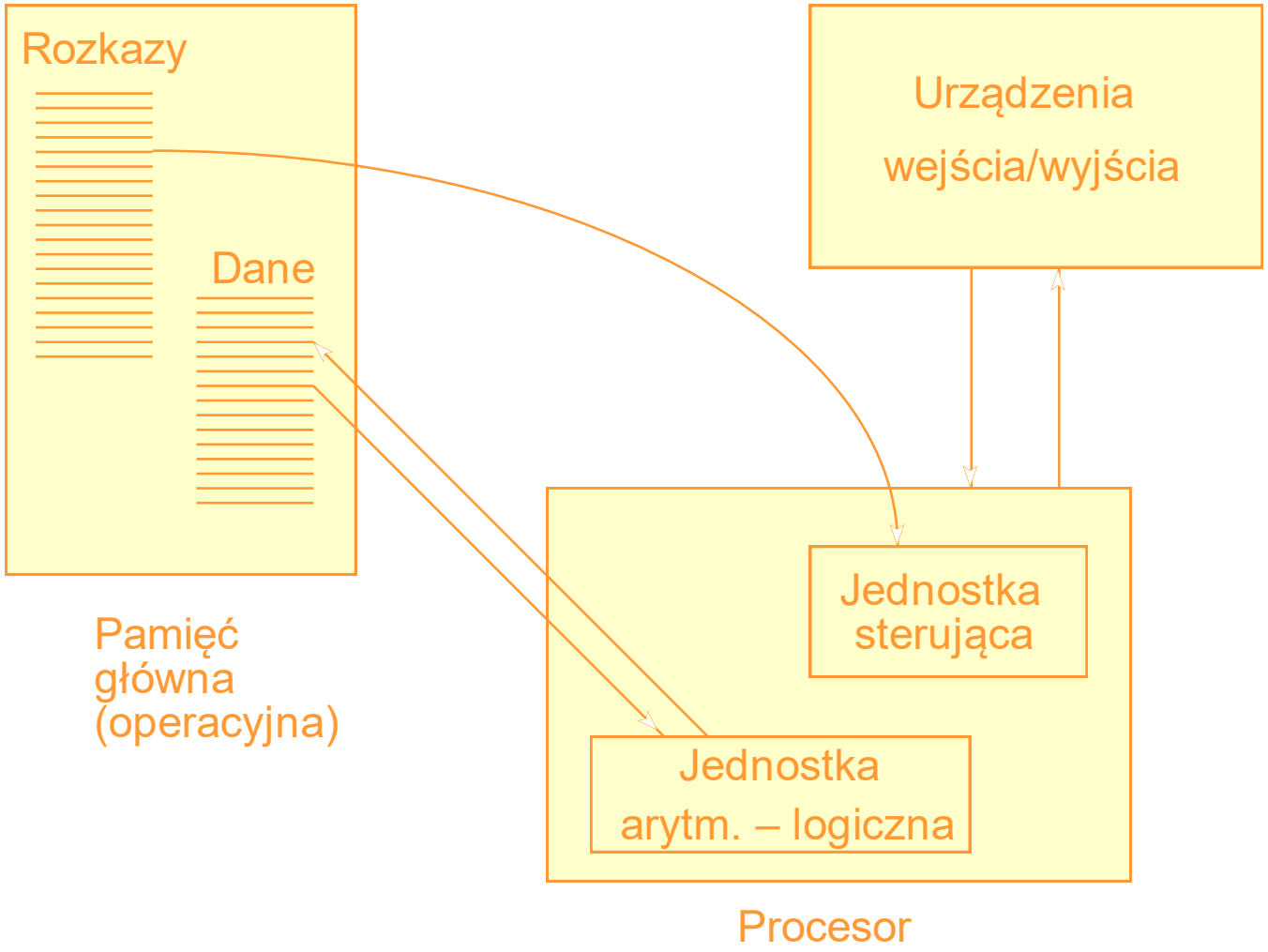
PC – wskaźnik instrukcji
ang. program counter

POLITECHNIKA GDAŃSKA

Podstawowe elementy modelu systemu komputerowego

POLITECHNIKA GDAŃSKA

Model komputera wg von Neumanna (1)

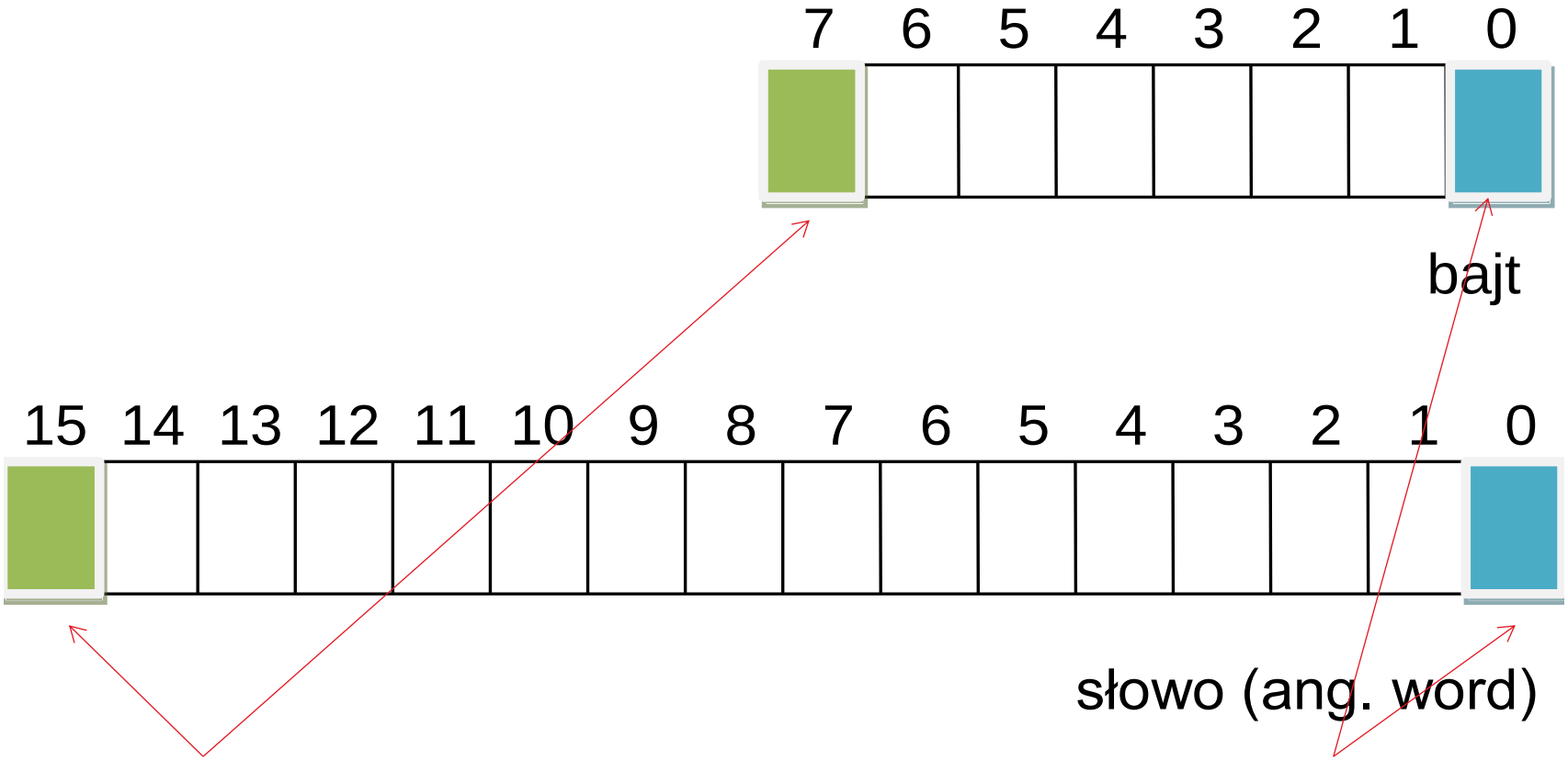


POLITECHNIKA GDAŃSKA

Pamięć główna (operacyjna) (1)

- Pamięć główna (operacyjna) komputera składa z dużej liczby komórek (np. kilku miliardów), a każda komórka utworzona jest z pewnej liczby bitów.
- Gdy komórkę pamięci tworzy 8 bitów, to mówimy, że *pamięć ma organizację bajtową* — taka organizacja jest typowa dla większości współczesnych komputerów.
- Poszczególne komórki mogą zawierać dane, na których wykonywane są obliczenia, jak również mogą zawierać rozkazy (instrukcje) dla procesora.

Bity, bajty, słowa, ... (1)



Bity, bajty, słowa, ... (2)

- Tworzone są zespoły bajtów:
  16-bitowe *słowa*,
  32-bitowe *podwójne słowa*,
  64-bitowe *poczwórne słowa*,
  w miarę potrzeby tworzy się także większe zespoły bajtów;
- Producenci procesorów ustalają konwencję numeracji bitów w bajtach i słowach — numeracja przyjęta m. in. w procesorach firmy Intel pokazana jest na rysunku na poprzednim slajdzie.

Pamięć główna (operacyjna) (2)

- Poszczególne bajty (komórki) pamięci są ponumerowane od 0 — numer komórki pamięci nazywany jest jej *adresem (fizycznym)*.
- Współczesne procesory są wielozadaniowe i posiadają zaawansowane mechanizmy tzw. translacji adresu (np. logicznego na fizyczny).
- Adres fizyczny przekazywany jest przez procesor (lub inne urządzenie) do podzespołów pamięci w celu wskazania położenia bajtu, który ma zostać odczytany lub zapisany.
- Zbiór wszystkich adresów fizycznych nazywa się *fizyczną przestrzenią adresową*.

Pamięć główna (operacyjna) (3)

- W wielu procesorach adresy fizyczne są 32-bitowe, co określa od razu maksymalny rozmiar zainstalowanej pamięci:
  2^{32} = 4 294 967 296 bajtów (4 GB)

Pamięć główna (operacyjna) (4)

- Do adresowania pamięci niezbędna jest określona liczba linii adresowych skojarzonych z bitami rejestru adresowego.
- Aktualnie wytwarzane procesory 64-bitowe pozwalają na dostęp do 1 TB pamięci (faktycznie 192 GB).
- Wersje 64-bitowe systemu MS Windows adresują 16 TB.

Rozmiar pamięci	Liczba bitów potrzebnych do adresowania
4 GB	32
64 GB	36
1 TB (1 terabajt)	40
16 TB	44
256 TB	48
4 PT (petabajty)	52

POLITECHNIKA GDAŃSKA

Procesor (CPU) - rejestry ogólnego przeznaczenia (1)

- W trakcie wykonywania obliczeń często wyniki pewnych operacji stają się danymi dla kolejnych operacji — w takim przypadku nie warto odsyłać wyników do pamięci operacyjnej, a lepiej przechować te wyniki w komórkach pamięci wewnątrz procesora — komórki te mają zazwyczaj postać rejestrów, które oznaczane są symbolami literowymi.
- W komputerach występuje także odrębny rodzaj pamięci określany jako pamięć podręczna (ang. cache memory) — temat ten będzie omawiany później.

POLITECHNIKA GDAŃSKA

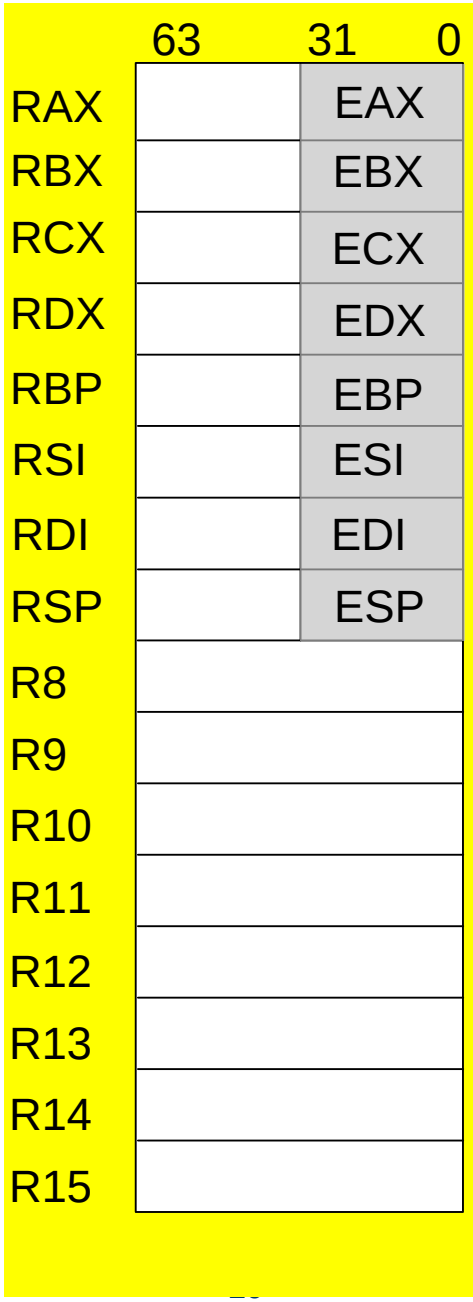
Rejestry ogólnego przeznaczenia (2)

- We wczesnych wersjach procesorów rodziny 8086 firmy Intel używano rejestrów 8-bitowych (np. AL, AH, BL, ...) i 16-bitowych (np. AX, BX, CX, ...).
- Później rozszerzono te rejestry do 32 bitów nadając im nazwy EAX, EBX,..
- Z kolei, w ostatnich latach rejestry 32-bitowe rozszerzono do 64 bitów nadając im nazwy RAX, RBX, ...

POLITECHNIKA GDAŃSKA

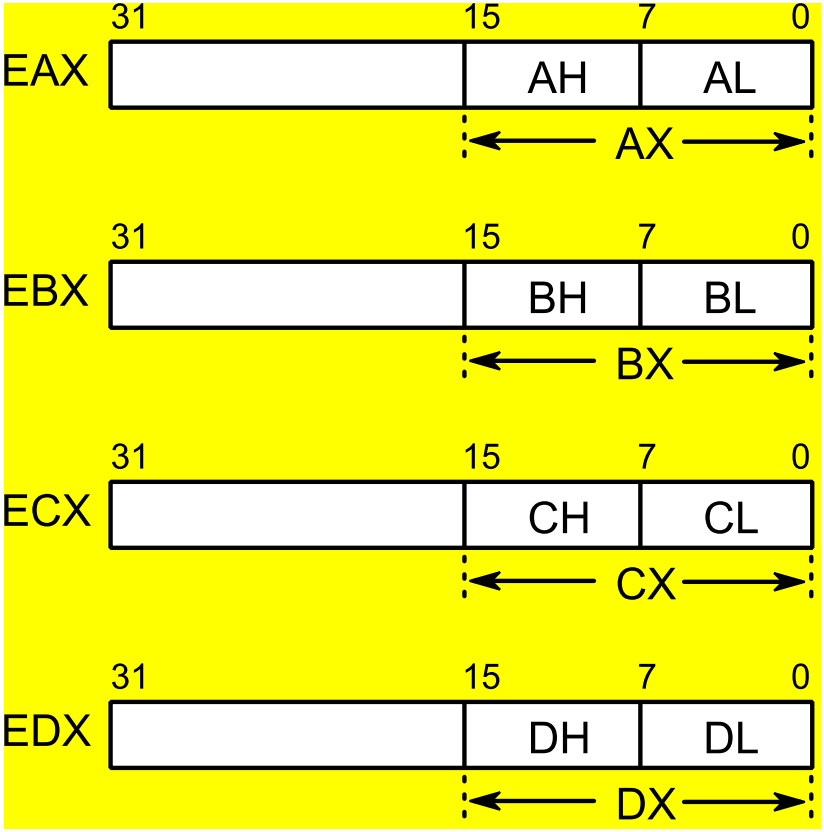
Rejestry ogólnego przeznaczenia (3)

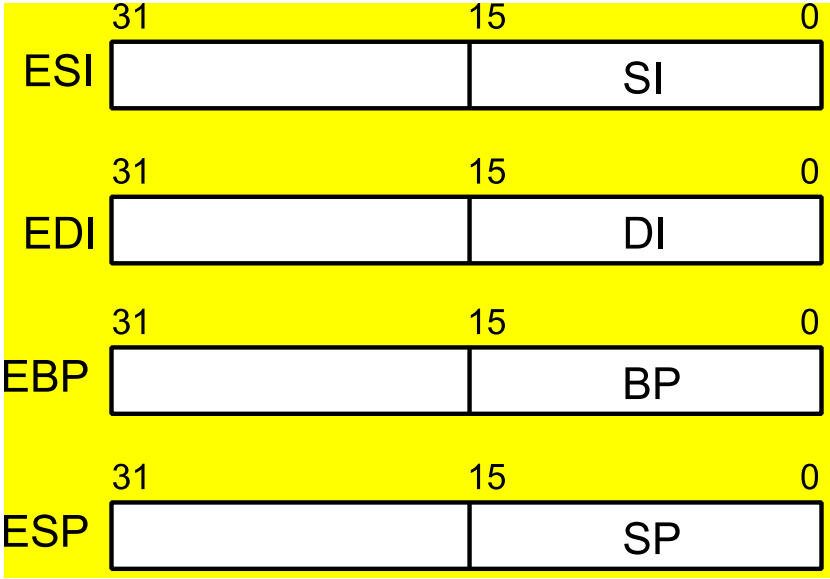
- Rejestry R8, R9, …, R15 dostępne są tylko w trybie 64-bitowym.
- Na kolejnych rysunkach przedstawiono współzależności rejestrów 8-, 16-, 32- i 64-bitowych.



POLITECHNIKA GDAŃSKA

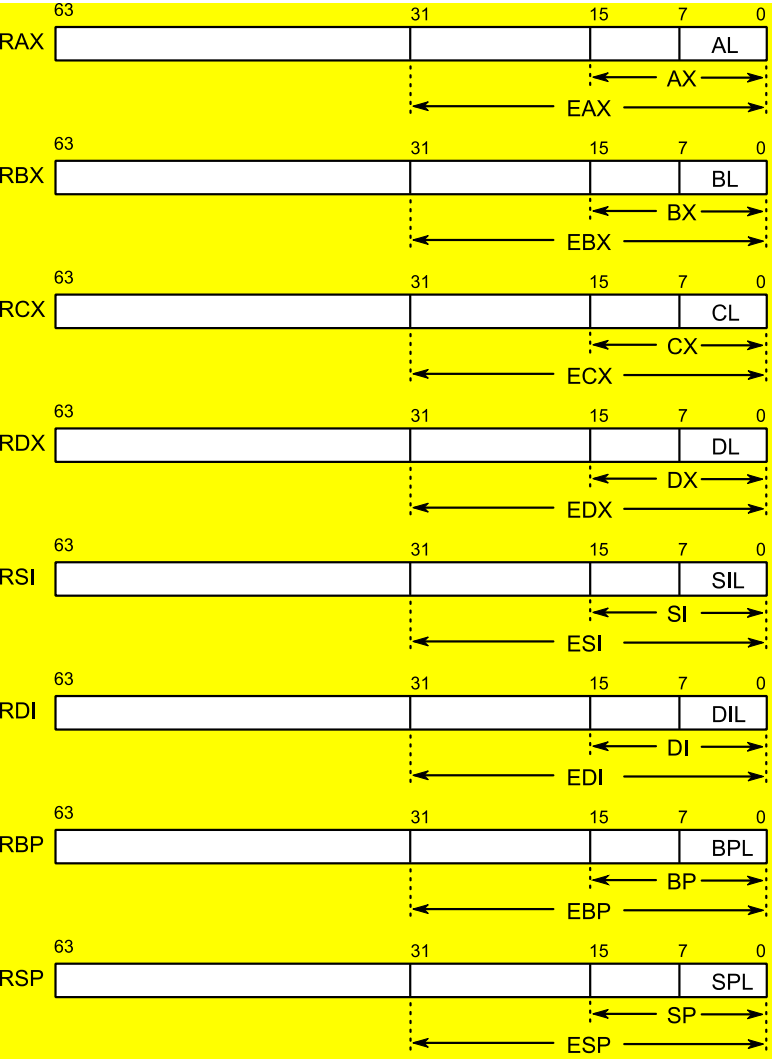
Rejestry ogólnego przeznaczenia (32-bitowe) (4)

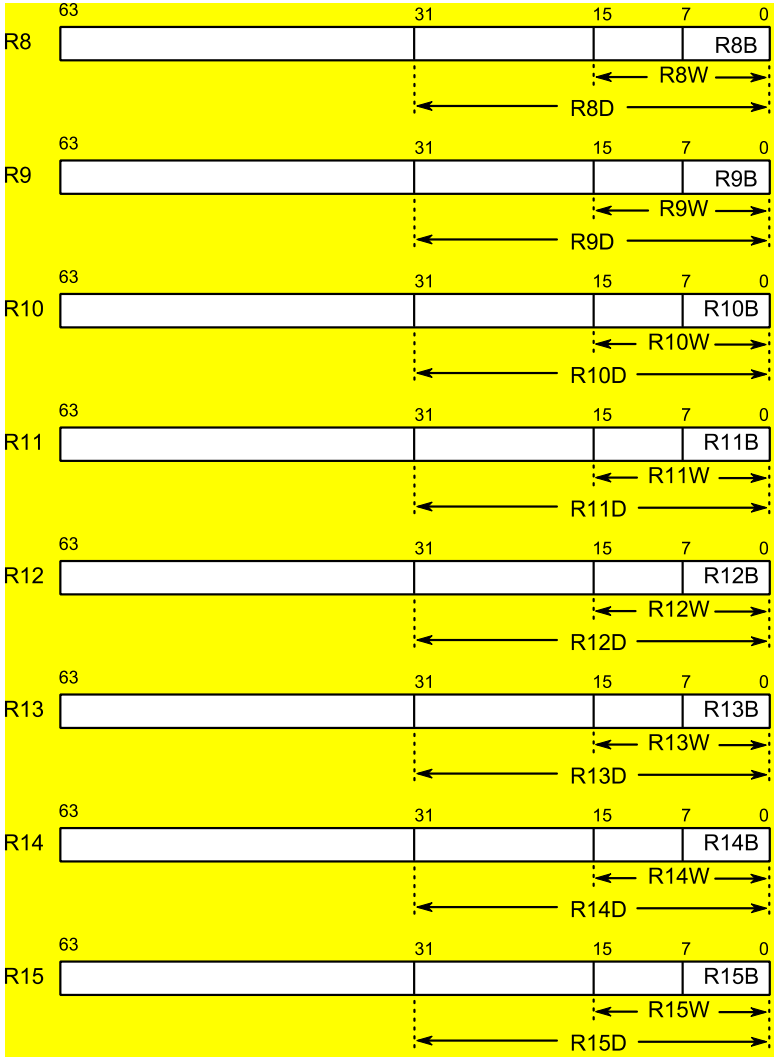




POLITECHNIKA GDAŃSKA

Rejestry ogólnego przeznaczenia (64-bitowe) (5)





POLITECHNIKA GDAŃSKA

Rejestry ogólnego przeznaczenia (6)

- Zauważmy, że rejestr AL stanowi 8 najmłodszych bitów rejestru RAX, a rejestr AX stanowi 16 najmłodszych bitów rejestru RAX.
- Analogiczny układ oznaczeń przyjęto dla innych rejestrów, przy czym nieco inny układ oznaczeń przyjęto dla rejestrów R8 – R15 (zob. rysunek na poprzednim slajdzie).
- Oprócz rejestrów ogólnego przeznaczenia, w procesorze mogą występować rejestry specjalne i sterujące. Omówione zostaną w dalszej części wykładu.

POLITECHNIKA GDAŃSKA

Urządzenia zewnętrzne

- Urządzenie zewnętrzne (np. klawiatura) stanowią oddzielną grupę komponentów sprzętowych.
- Zostaną omówione w dalszej części wykładu

POLITECHNIKA
GDAŃSKA

Procesor i cykl rozkazowy

POLITECHNIKA GDAŃSKA

Procesory powszechnego użytku (1)

- We współczesnych komputerach osobistych stosuje się różne typy procesorów, z których większość wywodzi się z procesora 8086 firmy Intel (r. 1978).
- Procesor 8086 (lub jego odmiana 8088) został zastosowany w komputerze osobistym IBM (rok 1981); później, mniej więcej co 4 lata, pojawiały się jego coraz bardziej rozbudowane wersje: 80286, 80386, 486, różne wersje Pentium, …, Intel Core i7, Intel Core 9.
- Obok firmy Intel, procesory zgodne programowo z ww. produkuje także firma AMD.

Procesory powszechnego użytku (2)

- Dla wygody opisu omawiane procesory (firmy Intel i AMD) oznaczane są w literaturze jako procesory zgodne z architekturą x86, przy czym do oznaczania wersji 64-bitowych używa się symbolu x86-64. W odniesieniu do procesorów firmy Intel używa się też oznaczenia *Intel 32* dla procesorów 32-bitowych i *Intel 64* dla procesorów 64-bitowych.
- Charakterystyczną cechą tych procesorów jest kompatybilność wsteczna, co oznacza że każdy nowy model procesora realizuje funkcje swoich poprzedników.

POLITECHNIKA GDAŃSKA

Procesory powszechnego użytku (3)

- M.in. programy dla komputera IBM PC opracowane na początku lat osiemdziesiątych mogą być wykonywane także w komputerze wyposażonym w procesor Intel Core i5, jednak systemy operacyjne nie przewidują takiej możliwości — pozostaje wykonywanie programu za pomocą maszyny wirtualnej, np. DOSBox.

Lista rozkazów procesora (1)

- Procesor składa się z wielu różnych podzespołów wykonawczych, które wykonują określone działania (np. sumowanie liczb) — podzespoły te reprezentowane są przez jednostkę arytmetyczno–logiczną (ang. arithmetic logic unit).
- Podzespoły wykonawcze podejmują działania wskutek sygnałów otrzymywanych z jednostki sterującej.

Lista rozkazów procesora (2)

- Dla każdego typu procesora konstruktorzy ustalają pewien podstawowy zbiór operacji — każdej operacji przypisuje się ustalony kod w postaci ciągu zero-jedynkowego.

- Do zbioru operacji podstawowych należą zazwyczaj cztery działania arytmetyczne, operacje logiczne na bitach (negacja, suma logiczna, iloczyn logiczny), operacje przesyłania, operacje porównywania i wiele innych.

POLITECHNIKA GDAŃSKA

Lista rozkazów procesora (3)

- Operacje zdefiniowane w zbiorze podstawowym nazywane są *rozkazami* lub *instrukcjami* procesora; każdy rozkaz ma przypisany ustalony kod zero-jedynkowy, który ma postać jednego lub kilku bajtów o ustalonej zawartości.
- Podstawowy zbiór operacji procesora jest zwykle nazywany *listą rozkazów procesora*.
- Ze względu na to, że posługiwanie się w procesie kodowania programu wartościami zero-jedynkowymi byłoby bardzo kłopotliwe, wprowadzono skróty literowe (tzw. mnemoniki) dla poszczególnych rozkazów (instrukcji) procesora.

POLITECHNIKA GDAŃSKA

Lista rozkazów procesora (4)

- Przekazanie procesorowi x86 rozkazu (instrukcji) w formie bajtu 01000010 spowoduje zwiększenie o 1 liczby umieszczonej w rejestrze EDX, natomiast przekazanie bajtu 01001010 spowoduje zmniejszenie tej liczby o 1.
- Podane tu operacje zapisuje się zazwyczaj w postaci symbolicznej, używając skrótów literowych opisujących funkcje rozkazu (mnemoników) i nazw rejestrów:

  kod 01000010 zastępuje się przez

  INC EDX (ang. increment zwiększenie)

  kod 01001010 zastępuje się przez

  DEC EDX (ang. decrement zmniejszenie)

Lista rozkazów procesora (5)

- Często polecenia przekazywane procesorowi składają się z kilku bajtów, np. bajty 10000000 11000111 00100101 traktowane są przez procesor jako polecenie dodania liczby 37 do liczby znajdującej się w rejestrze BH.
- Powyższa operacja zapisana w postaci symbolicznej ma postać:

  ADD BH, 37 (ang. addition dodawanie)

- Mnemoniki, nazwy rejestrów czy liczby dziesiętne są niezrozumiałe dla procesora i przed wprowadzeniem programu do pamięci muszą być zamienione na odpowiadające im kody zero-jedynkowe — programy dokonujące takiej konwersji nazywane są *asemblerami*

POLITECHNIKA GDAŃSKA

Wykonywanie programu w języku maszynowym przez procesor (1)

- W podanym ujęciu algorytm obliczeń przedstawiany jest za pomocą operacji ze zbioru podstawowego.
- Algorytm zakodowany jest w postaci sekwencji ciągów zero-jedynkowych zdefiniowanych w podstawowym zbiorze operacji procesora — tak zakodowany algorytm nazywać będziemy *programem w języku maszynowym.*

POLITECHNIKA GDAŃSKA

Wykonywanie programu w języku maszynowym przez procesor (2)

- Program w języku maszynowym przechowywany jest w pamięci głównej (operacyjnej) komputera.
- Wykonywanie programu polega na przesyłaniu kolejnych ciągów zero-jedynkowych z pamięci głównej do układu sterowania procesora.
- Zadaniem układu sterowania, po odczytaniu takiego ciągu, jest wygenerowanie odpowiedniej sekwencji sygnałów kierowanych do poszczególnych podzespołów wykonawczych , tak by w rezultacie wykonać wymaganą operację (np. dodawanie).

Język maszynowy a asembler

- Język maszynowy jest kłopotliwy w użyciu nawet dla specjalistów; znacznie wygodniejszy jest spokrewniony z nim język asemblera, który będzie omawiany dalej.
- W asemblerze kody bajtowe rozkazów zapisuje się w postaci skrótów literowych (mnemoników), np. `ADD`, `DIV`, `MOV`, `LOOP`, `JMP`, itd.; dostępnych jest wiele innych udogodnień, jak na przykład możliwość zapisu liczb w postaci dziesiętnej lub szesnastkowej.

POLITECHNIKA GDAŃSKA

Cykl rozkazowy (1)

- Wstępnie założymy, że sekwencja instrukcji tworzących program (w postaci ciągów zerojedynkowych) ułożona jest pamięci komputera w porządku naturalnym, tj. następna instrukcja umieszczona jest w pamięci bezpośrednio za poprzednią.
- W dalszej części wykładu pokażemy, że założenie to nie dotyczy struktur decyzyjnych (czyli instrukcji typu `if .. then .. else` i różnych typów pętli).

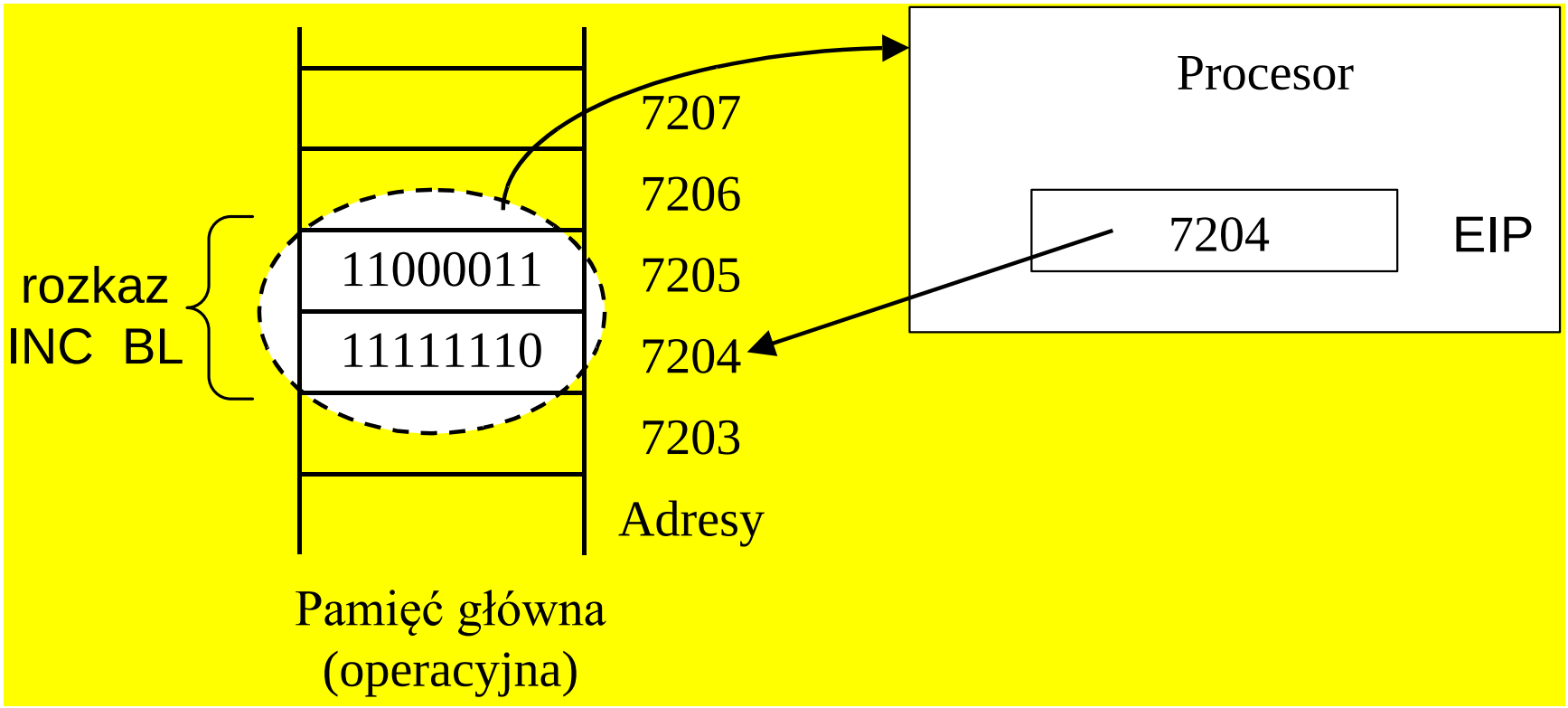
Cykl rozkazowy (2)

- Tak więc proces pobierania kolejnych instrukcji z pamięci operacyjnej i ich wykonywania musi być precyzyjnie zorganizowany, tak by natychmiast po wykonaniu kolejnej instrukcji procesor pobierał z pamięci następną.
- Aby pobrać tę instrukcję, procesor musi oczywiście znać jej położenie w pamięci głównej (operacyjnej) — informacje o położeniu kolejnej instrukcji są umieszczone w specjalnym rejestrze, nazywanym *wskaźnikiem instrukcji.*

Cykl rozkazowy (3)

- W architekturze x86 używany jest 32-bitowy wskaźnik instrukcji oznaczony jest symbolem EIP (ang. extended instruction pointer).
- W architekturze x86-64 używany jest 64-bitowy rejestr RIP; w innych architekturach omawiany rejestr nazywany jest także *licznikiem rozkazów* lub *licznikiem programu* (PC – ang. program counter).

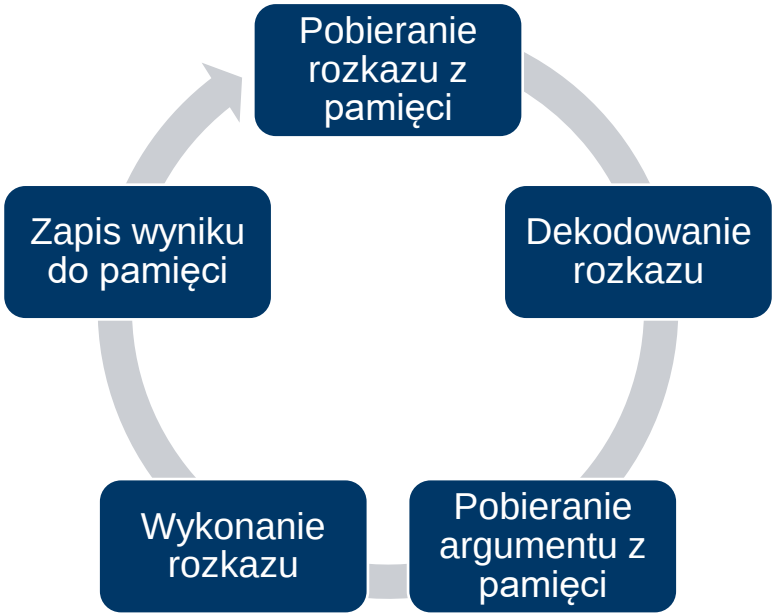
Cykl rozkazowy (4)



POLITECHNIKA GDAŃSKA

Uproszczony cykl rozkazowy (5)

- Czynności wykonywane przez procesor w trakcie pobierania i wykonywania poszczególnych rozkazów powtarzane są cyklicznie, a cały proces nosi nazwę *cyklu rozkazowego.*
- Cykl rozkazowy jest podstawą działania wszystkich komputerów.



POLITECHNIKA GDAŃSKA

Rozkazy niesterujące (1)

- Podstawowe operacje przetwarzania: przesyłanie, operacje arytmetyczne i logiczne, przesunięcia, itd., realizują rozkazy (instrukcje) określane jako niesterujące .
- Charakterystyczną cechą tej klasy rozkazów jest to, że nie zmieniają one naturalnego porządku wykonywania rozkazów, co oznacza, że po wykonaniu rozkazu z tej klasy procesor rozpoczyna wykonywać rozkaz bezpośrednio przylegający w pamięci do rozkazu właśnie wykonanego.

Rozkazy niesterujące (2)

- W procesorach zgodnych z architekturą x86 kolejna zawartość rejestru EIP jest obliczana wg formuły:

EIP ← EIP + <*liczba bajtów aktualnie wykonywanego rozkazu*>

POLITECHNIKA GDAŃSKA

Zapis asemblerowy typowych rozkazów niesterujących (1)

- Operacja przesłania zawartości komórki pamięci do rejestru realizowana przez rozkaz oznaczony skrótem literowym (mnemonikiem) MOV.
- Rozkaz ten ma dwa argumenty: pierwszy argument określa cel, czyli "*dokąd przesłać*", drugi zaś określa źródło, czyli "*skąd przesłać*" lub "*co przesłać*":

MOV dokąd przesłać , skąd (lub co) przesłać

POLITECHNIKA GDAŃSKA

Zapis asemblerowy typowych rozkazów niesterujących (2)

- Argumenty rozkazów wykonujących operacje dodawania ADD i odejmowania SUB zapisuje się podobnie jak argumenty rozkazu MOV

ADD dodajna , dodajnik

SUB odjemna , odjemnik

wynik wpisywany jest do obiektu wskazanego przez pierwszy argument

POLITECHNIKA GDAŃSKA

Zapis asemblerowy typowych rozkazów niesterujących (3)

- Rozkaz wykonuje operację na dwóch wartościach wskazanych przez pierwszy i drugi operand, a wynik wpisywany jest do pierwszego operandu

- Ogólnie, rozkaz postaci

  „operacja” cel, źródło

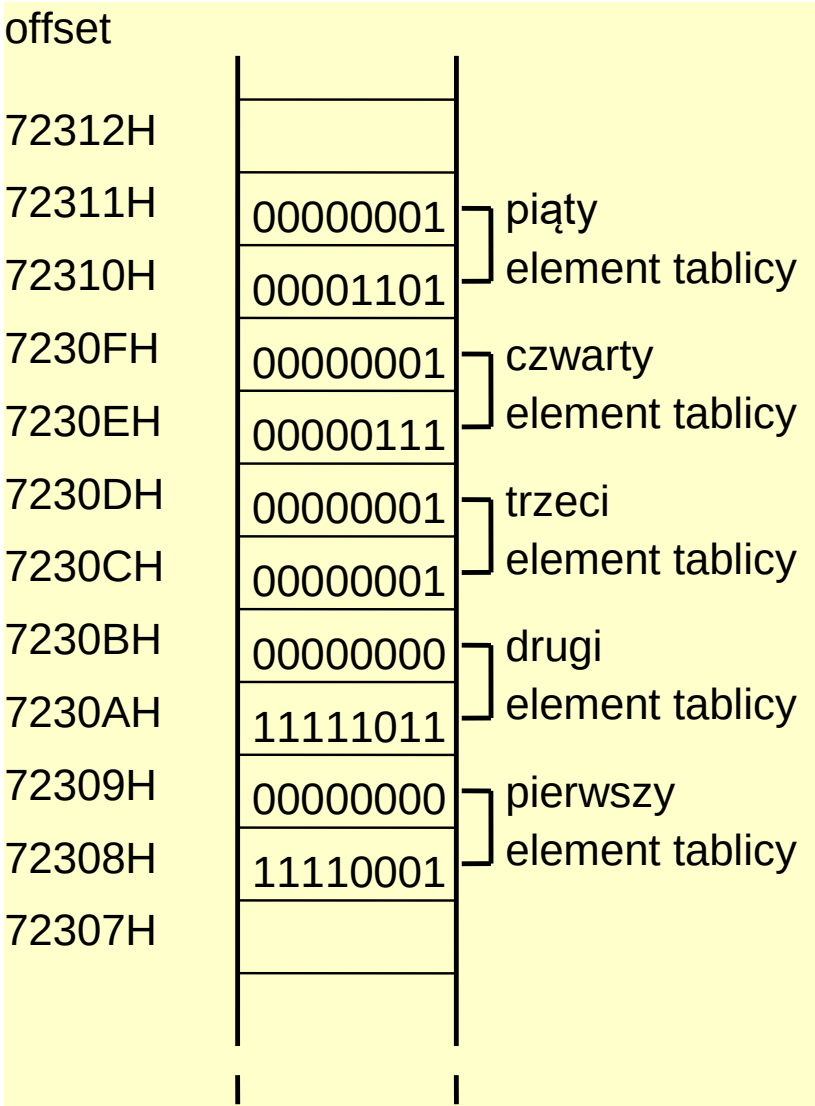
  wykonuje działanie

  cel ← cel „operacja” źródło

POLITECHNIKA GDAŃSKA

Przykład: sumowanie elementów tablicy (1)

- Przykładowa tablica zawiera pięć liczb binarnych 16-bitowych całkowitych bez znaku.
- Zakładamy, że w trakcie sumowania wszystkie wyniki pośrednie uzyskiwane w trakcie sumowania dadzą się przedstawić w postaci liczb 16-bitowych, tzn. nie wystąpi przepełnienie (nadmiar).



POLITECHNIKA GDAŃSKA

Przykład: sumowanie elementów tablicy (2)

- **Operacje sumowania w postaci symbolicznej**

  AX ← [72308H]

  AX ← AX + [7230AH]

  AX ← AX + [7230CH]

  AX ← AX + [7230EH]

  AX ← AX + [72310H]

- **AX oznacza tu zawartość 16-bitowego rejestru AX**

- Zapis [72308H] oznacza zawartość komórki pamięci znajdującej się w obszarze danych o adresie podanym w nawiasach kwadratowych.
- Litera H oznacza, że liczba podana jest w zapisie szesnastkowym.

POLITECHNIKA GDAŃSKA

Przykład: sumowanie elementów tablicy (3)

```
mov   ax, ds:[72308H]
add   ax, ds:[7230AH]
add   ax, ds:[7230CH]
add   ax, ds:[7230EH]
add   ax, ds:[72310H]
```

- symbol ds: oznacza, że pobierane dane znajdują się w obszarze danych programu

- Operacje sumowania w postaci sekwencji rozkazów procesora podanych w języku asemblera

POLITECHNIKA GDAŃSKA

Przykład: sumowanie elementów tablicy (4)

- Omawiane operacje w postaci zrozumiałej dla procesora wyglądają tak:

- 01100110 10100001 00001000 00100011 00000111 00000000

(mov ax, ds:[72308H])

- 01100110 00000011 00000101 00001010 00100011 00000111 00000000

(add ax, ds:[7230AH])

POLITECHNIKA GDAŃSKA

Przykład: sumowanie elementów tablicy (5)

- 01100110 00000011 00000101 00001100 00100011 00000111 00000000

  (add ax, ds:[7230CH])

- 01100110 00000011 00000101 00001110 00100011 00000111 00000000

  (add ax, ds:[7230EH])

- 01100110 00000011 00000101 00010000
  00100011 00000111 00000000

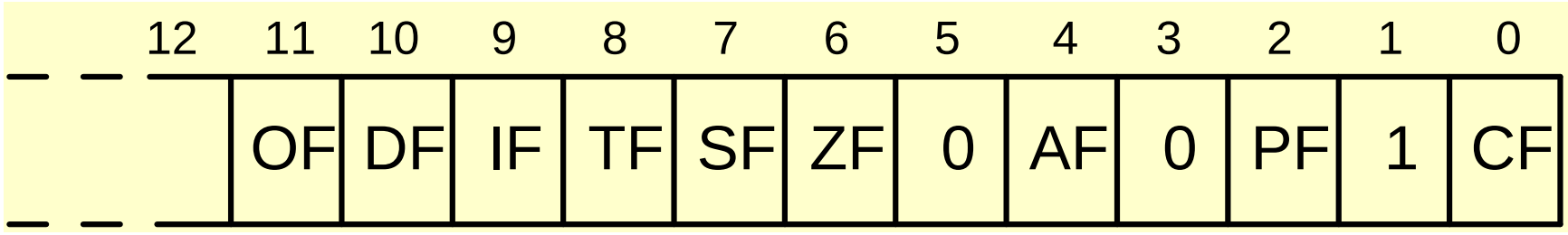
  (add ax, ds:[72310H])

POLITECHNIKA GDAŃSKA

Rejestr stanu procesora (rejestr znaczników) (1)

- Niektóre rozkazy niesterujące, prócz właściwych czynności (np. odejmowania dwóch liczb) przekazują dodatkowe informacje opisujące własności wyniku operacji: czy wynik jest równy zero, czy wynik jest liczbą ujemną, czy wystąpiło przeniesienie, itp. — omawiane informacje wpisywane są do 32-bitowego rejestru stanu procesora (nazywanego także *rejestrem znaczników*).
- Cały rejestr stanu procesora można traktować jako zespół rejestrów 1- lub 2-bitowych, które określane są jako znaczniki.

Rejestr stanu procesora (2)



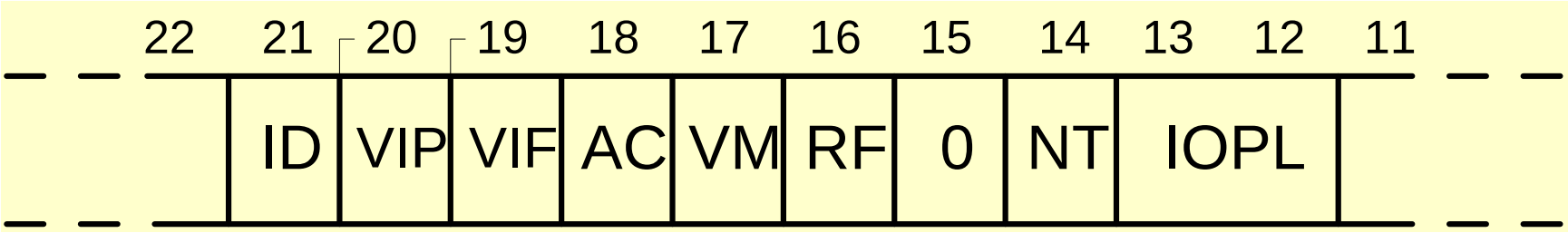
- Powyższy rysunek pokazuje fragment rejestru stanu procesora.
- Do znacznika zera ZF (ang. zero flag) wpisywana jest liczba 1, jeśli wynik operacji arytmetycznej lub logicznej wynosił 0 — w przeciwnym razie znacznik ZF jest zerowany.
- Znacznik CF ustawiany jest w stan 1, jeśli w trakcie dodawania wystąpiło przeniesienie, albo w trakcie odejmowania wystąpiła prośba o pożyczkę.

Rejestr stanu procesora (3)

- W miarę potrzeby stan znaczników może być testowany przez (omawiane dalej) *instrukcje sterujące*.
- Pozostałe bity rejestru stanu procesora opisują dalsze własności wyniku operacji (np. czy wynik jest liczbą ujemną — znacznik SF), jak również aktualnie włączony tryb adresowania dla operacji na blokach danych, zdolność procesora do przyjmowania sygnałów z urządzeń zewnętrznych i inne.

Rejestr stanu procesora (4)

- Bity rejestru stanu o numerach 12, 13, …, 21 zawierają informacje o stanie procesora, które są odczytywane i zapisywane przez system operacyjny.
- Bity o numerach 22 ÷ 31 nie są używane.



Rejestr stanu procesora (5)

- Zawartość rejestru stanu procesora można zapisać na wierzchołku stosu (zob. dalszy opis) za pomocą rozkazu `pushf`.
- Tak samo za pomocą rozkazu `popf` można przesłać zawartość wierzchołka stosu do rejestru stanu procesora. Jednak ewentualne zmiany znaczników wykorzystywanych przez system operacyjny zostaną zignorowane.

POLITECHNIKA GDAŃSKA

Reprezentacja podstawowych danych w pamięci

POLITECHNIKA GDAŃSKA

Reprezentacja danych w pamięci komputera (1)

- Współczesne komputery przetwarzają dane reprezentujące wartości liczbowe, teksty, dane opisujące dźwięki i obrazy i wiele innych.

- Dane o charakterze nieliczbowym (np. znaki, sygnały) muszą być zapisane (zakodowane) w postaci liczb — w rezultacie wszelkie dane przechowywane i przetwarzane w komputerze mają postać ciągów złożonych z zer i jedynek.

POLITECHNIKA GDAŃSKA

Reprezentacja danych w pamięci komputera (2)

- Dane liczbowe przedstawiane są w różnych formatach, ale z punktu widzenia działania procesora szczególne znaczenie mają liczby całkowite — opisy techniczne wielu rozkazów procesora odnoszą się bowiem do operacji wykonywanych na liczbach całkowitych kodowanych w naturalnym kodzie binarnym (liczby bez znaku) albo do operacji wykonywanych na liczbach całkowitych binarnych ze znakiem.

POLITECHNIKA GDAŃSKA

Reprezentacja danych w pamięci komputera (3)

- Nie oznacza to, że rozkazy procesora nie mogą wykonywać działań na liczbach ułamkowych — programista może odpowiednio dostosować algorytm obliczeń w taki sposób, ażeby działania na ułamkach zostały zastąpione przez działania na liczbach całkowitych.

POLITECHNIKA GDAŃSKA

Reprezentacja danych w pamięci komputera (4)

- Ponadto procesor, i stowarzyszony z nim koprocesor arytmetyczny, posiadają zdolność wykonywania działań na liczbach w formacie zmiennoprzecinkowym (zmiennopozycyjnym); liczby te kodowane są w specyficzny sposób — zagadnienia związane z liczbami zmiennoprzecinkowymi omawiane będą w dalszej części wykładu.

POLITECHNIKA GDAŃSKA

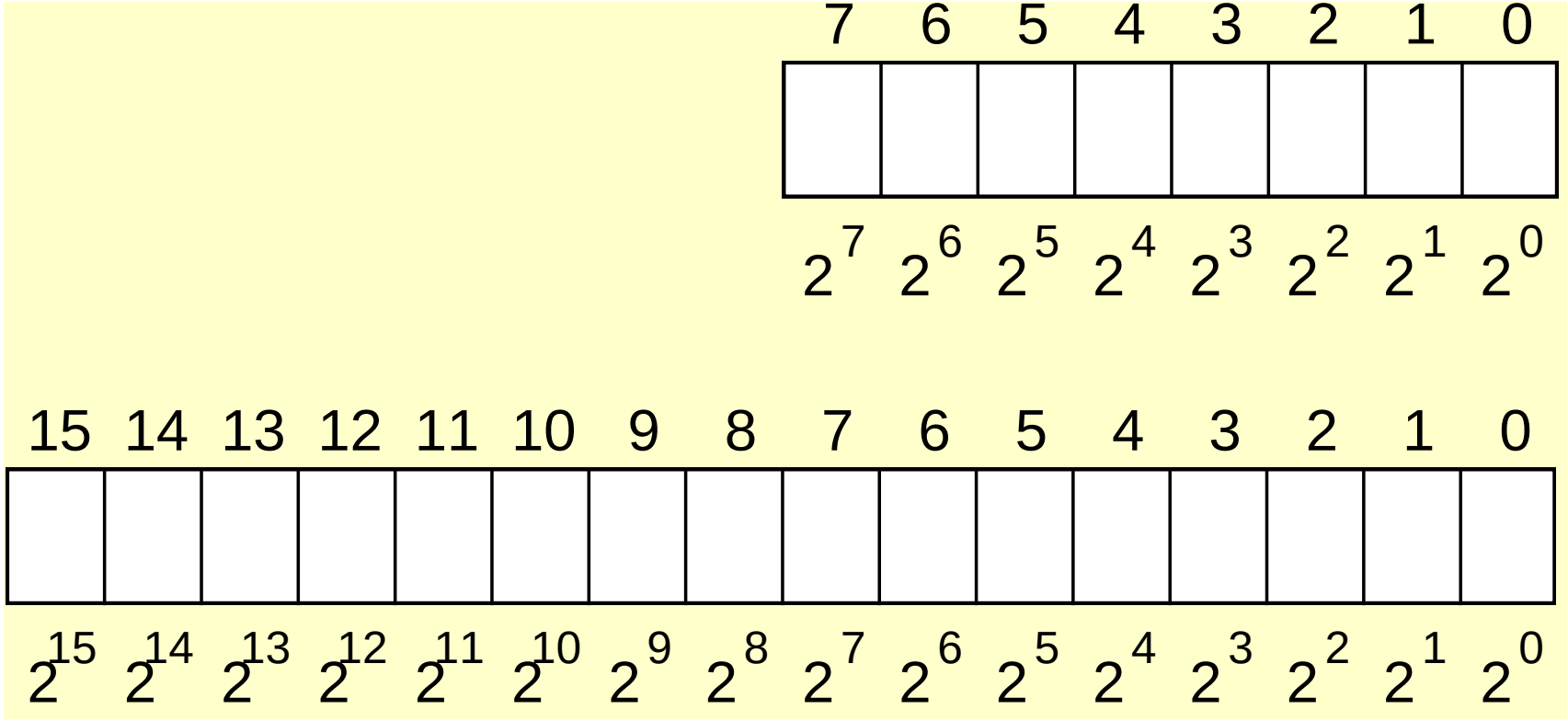
Kodowanie liczb całkowitych

- W wielu współczesnych procesorach wyróżnia się wiele formatów liczb — na razie rozpatrzymy tylko liczby całkowite bez znaku, kodowane w naturalnym kodzie binarnym (NKB).

- Numeracja bitów i przyporządkowanie wag dla liczb 8- i 16-bitowych stosowana w procesorach Intel i AMD pokazana jest na rysunku.

POLITECHNIKA GDAŃSKA

Kodowanie liczb całkowitych bez znaku (1)



Kodowanie liczb całkowitych bez znaku (2)

- Wartość liczby binarnej bez znaku (*m* oznacza liczbę bitów rejestru lub komórki pamięci) określa wyrażenie

$$w = \sum_{i=0}^{m-1} x_i \cdot 2^i$$

gdzie x_i oznacza wartość *i*-tego bitu liczby.

POLITECHNIKA GDAŃSKA

Przykłady kodowania liczb bez znaku

- liczba 253 w różnych formatach:

  8-bitowa: 1111 1101

  16-bitowa: 0000 0000 1111 1101

  32-bitowa: 0000 0000 0000 0000 0000 0000 1111 1101

- liczba 2 007 360 447 w postaci 32-bitowej liczby binarnej:

  0111 0111 1010 0101 1110 0011 1011 1111

POLITECHNIKA GDAŃSKA

Zakresy liczb całkowitych bez znaku

- liczby 8-bitowe: <0, 255>
- liczby 16-bitowe <0, 65535>
- liczby 32-bitowe <0, 4 294 967 295>
- liczby 64-bitowe <0, 18 446 744 073 709 551 615>
  (*osiemnaście trylionów*
  czterysta czterdzieści sześć biliardów
  siedemset czterdzieści cztery biliony
  siedemdziesiąt trzy miliardy
  siedemset dziewięć milionów
  pięćset pięćdziesiąt jeden tysięcy
  sześćset piętnaście)

- 1 trylion = 10^{18}

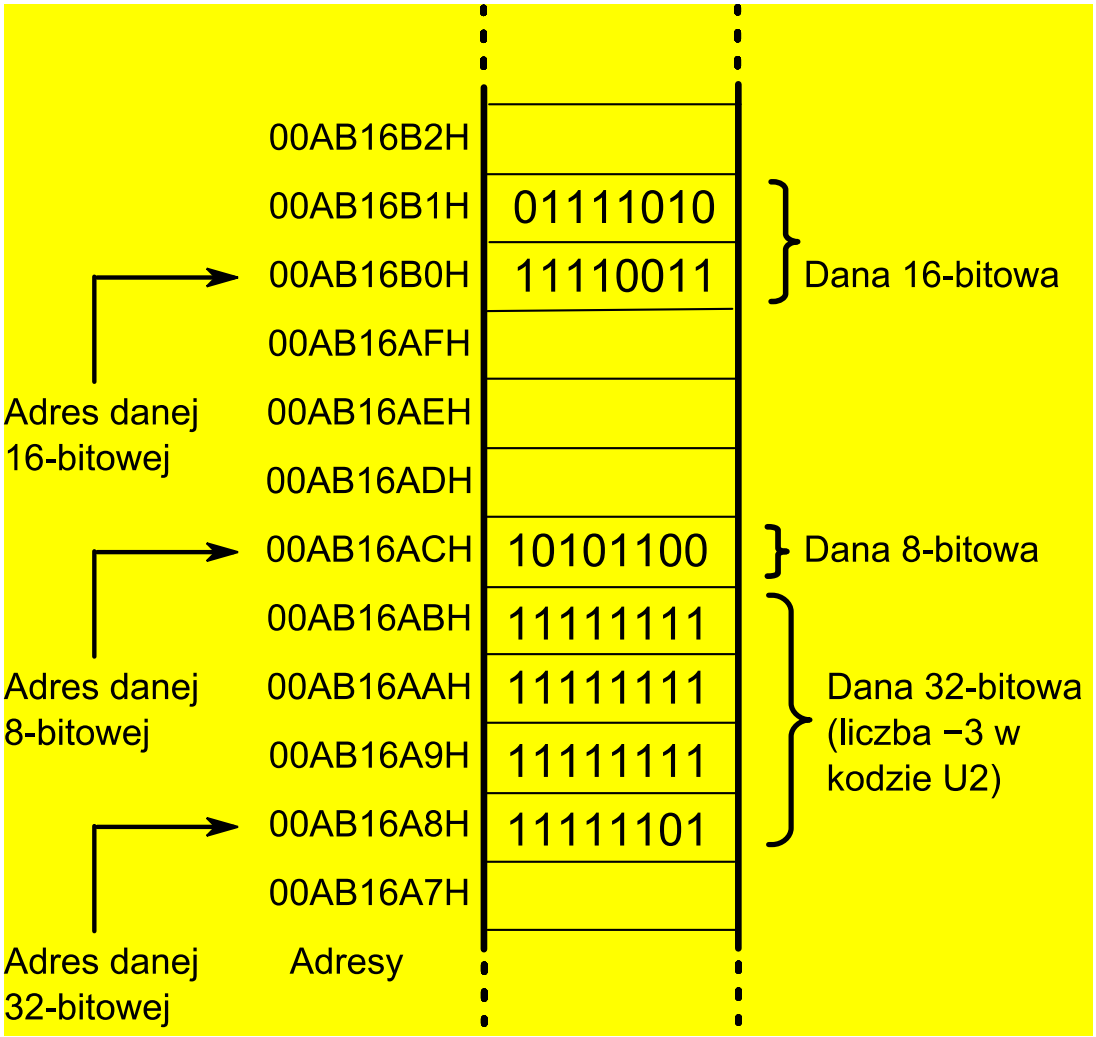
POLITECHNIKA GDAŃSKA

Przechowywanie danych w pamięci komputera (1)

- Liczby występujące w programach często przekraczają 255 i muszą być zapisywane na dwóch, czterech lub na większej liczbie bajtów.
- Także dane nie będące liczbami muszą być często zapisywane na większej liczbie bajtów (np. niektóre znaki w formacie UTF–8).
- Niezależnie od liczby bajtów zajmowanych przez daną przyjęto, że jej położenie w pamięci określa się poprzez podanie tylko jednego adresu, który wskazuje o bajt najmniejszym (najniższym) adresie – ilustruje to rysunek na następnym slajdzie.

POLITECHNIKA GDAŃSKA

Przechowywanie danych w pamięci komputera (2)



POLITECHNIKA GDAŃSKA

Porządek bajtów (1)

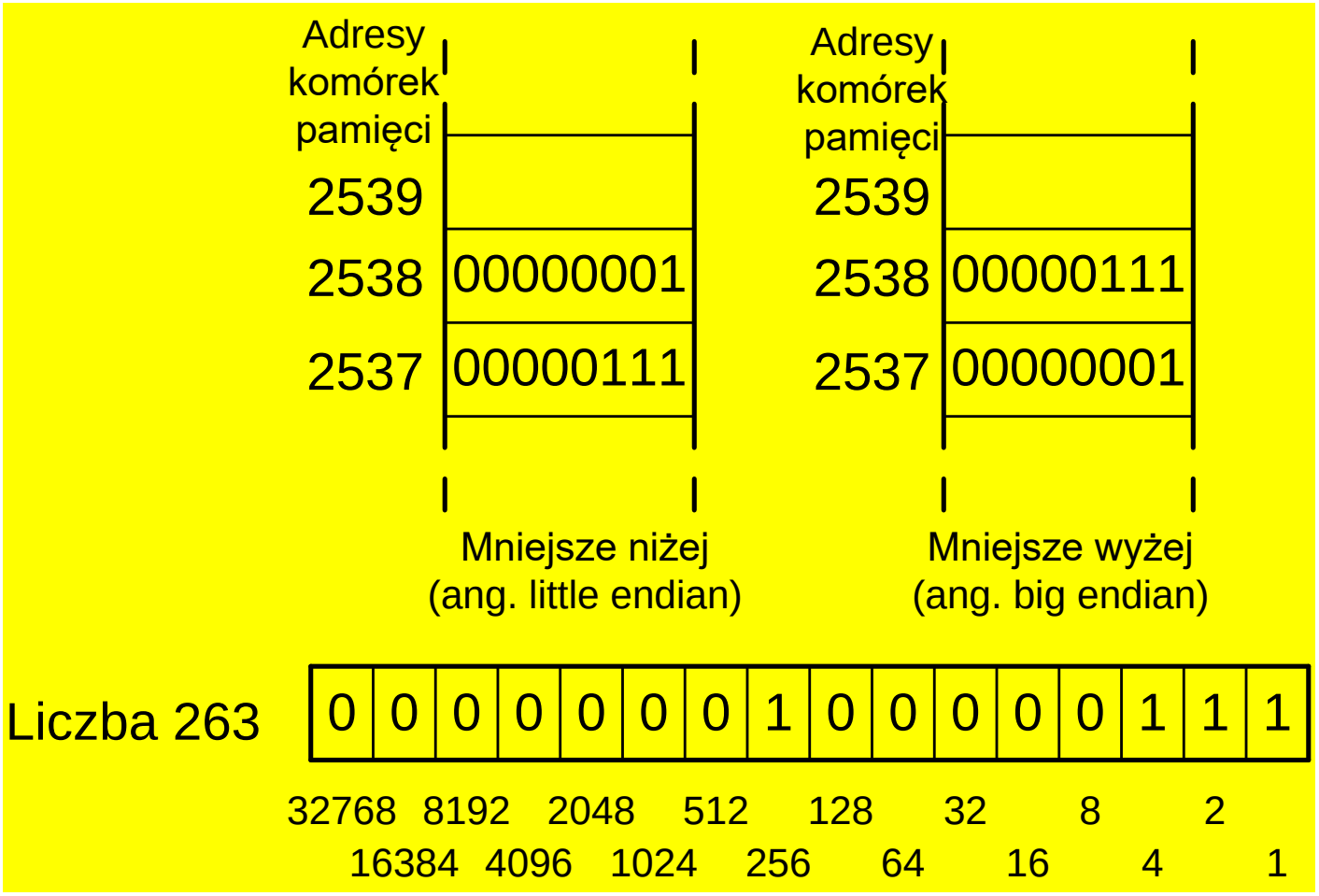
- Dane występujące w programach często muszą być zapisywane na dwóch, czterech lub na większej liczbie bajtów (zob. poprzedni slajd).
- W odniesieniu do liczb w komputerach przyjęto dwa podstawowe schematy rozmieszczenia bajtów w pamięci:

  mniejsze niżej (ang. little endian) – młodsza część liczby zajmuje bajty o niższych adresach;

  mniejsze wyżej (ang. big endian) – młodsza część liczby zajmuje bajty o wyższych adresach.
- Podany dalej przykład ilustruje dwa sposoby przechowywania w pamięci liczby 16-bitowej.

POLITECHNIKA GDAŃSKA

Porządek bajtów (2)

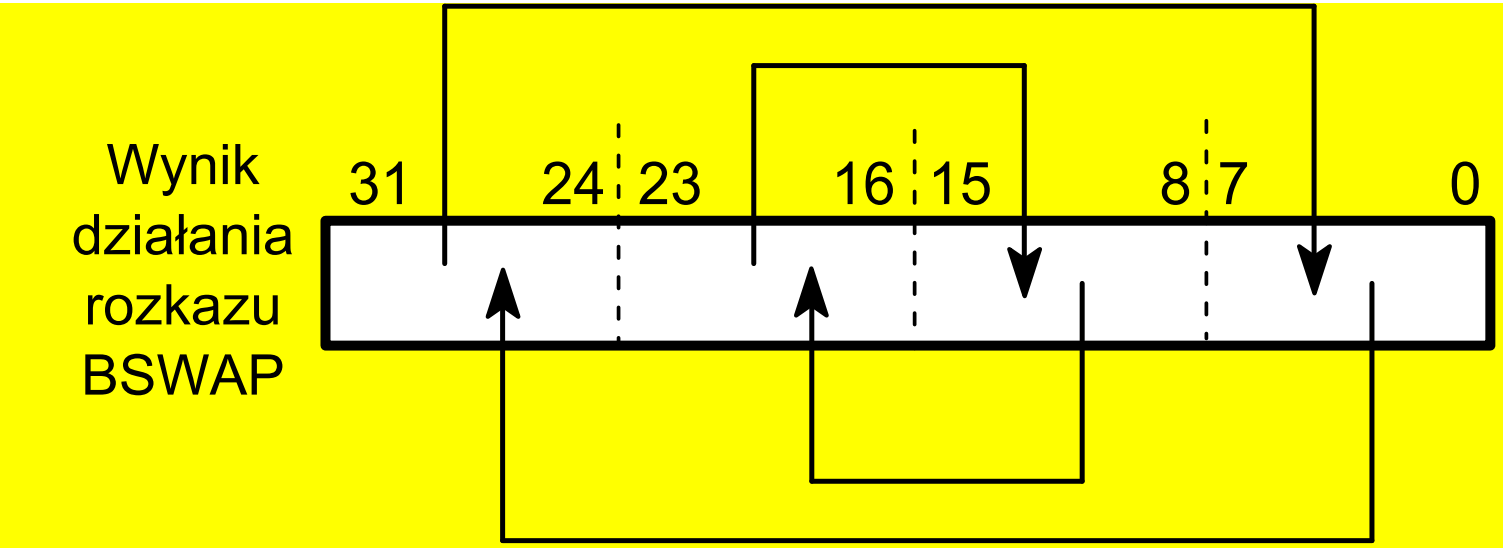


Porządek bajtów (3)

- Format *little endian* stosowany jest m.in. w procesorach rodziny x86/64 (AMD/Intel), VAX, Alpha.
- Format *big endian* stosowany jest m.in. w procesorach Motorola 680x0, SunSPARC i większości procesorów klasy RISC.
- Procesor PowerPC udostępnia oba tryby pracy — w rejestrze MSR (Machine Status Register) wprowadzono dwa bity, z których pierwszy określa stosowaną kolejność bajtów dla procesora działającego w trybie systemu operacyjnego (ang. kernel mode), drugi bit określa aktualną kolejność dla zwykłego programu.

Porządek bajtów (4)

- W procesorach rodziny x86 dostępny jest rozkaz BSWAP, który zamienia liczbę w 32-bitową w formacie mniejsze niżej (little endian) na format mniejsze wyżej (big endian) lub odwrotnie. Działanie rozkazu ilustruje poniższy rysunek.



Porządek bajtów (5)

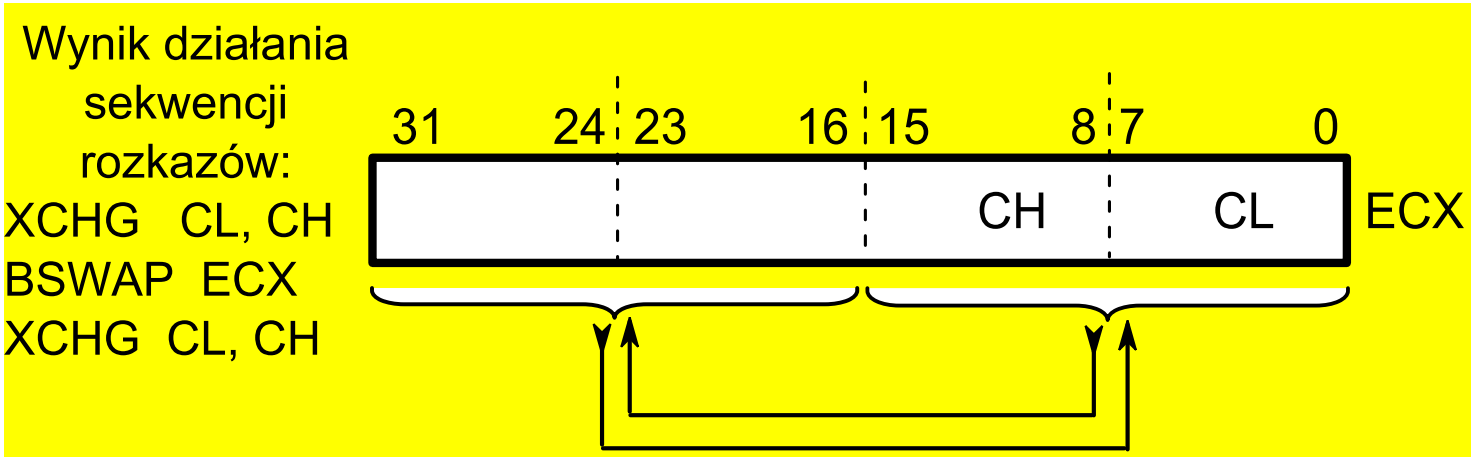
- Analogiczne działania rozkaz BSWAP wykonuje na rejestrze 64-bitowym.

- Zamianę formatu liczb 16-bitowych wykonuje się za pomocą rozkazu XCHG, który zamienia zawartości obu operandów, np.:

```
xchg  dl, dh
```

Porządek bajtów (6)

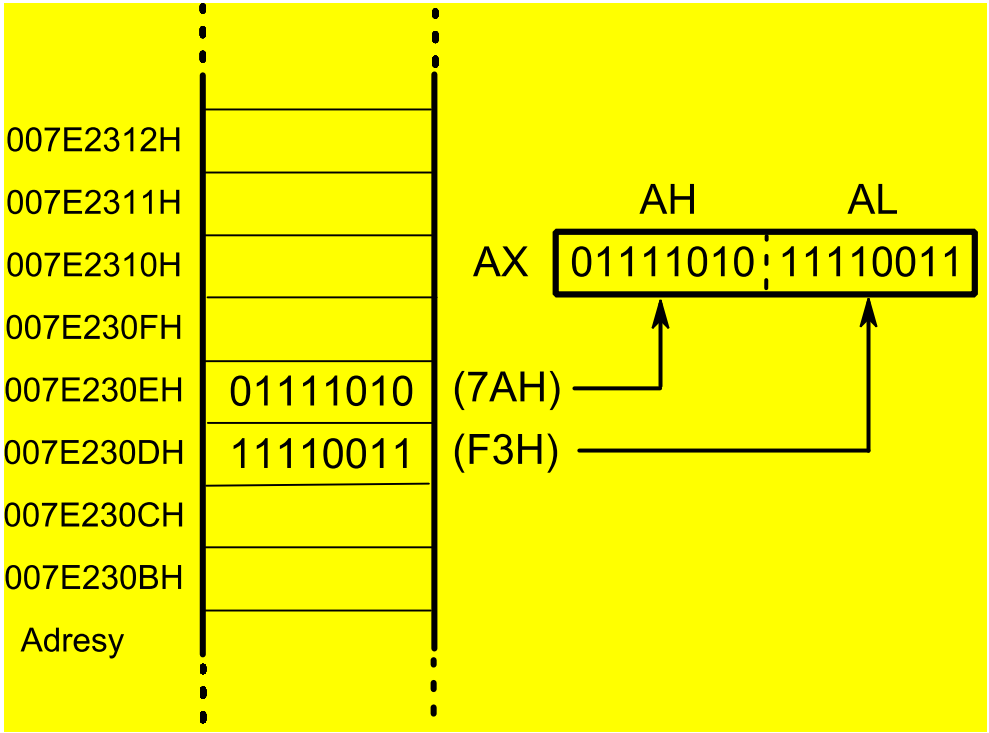
- Poniższy przykład ilustruje możliwość zamiany zawartości młodszej i starszej części rejestru 32-bitowego za pomocą rozkazów BSWAP i XCHG.



POLITECHNIKA GDAŃSKA

Przesyłanie liczb z pamięci do rejestru (i odwrotnie)

- Przykład przesłania 16-bitowej liczby 7AF3H (=31745) z pamięci głównej do rejestru AX (architektura x86)



POLITECHNIKA GDAŃSKA

Rozpoznawanie formatu mniejsze niżej/wyżej (1)

- Identyfikację schematu reprezentacji liczb stosowanego w danym komputerze można przeprowadzić za pomocą niżej podanego fragmentu programu w języku C.

- W podanym przykładzie założono, że wartości typu int są 32-bitowe.

POLITECHNIKA GDAŃSKA

Rozpoznawanie formatu mniejsze niżej/wyżej (2)

```
unsigned  int  liczba  =  0x12345678;

unsigned  char  *  wsk  =  (unsigned  char  *) & liczba;

if  (wsk[0] == 0x12)
  printf ("\nFormat mniejsze wyżej (big endian)");
else
  printf ("\nFormat mniejsze niżej (little endian)");
```

POLITECHNIKA GDAŃSKA

Wyrównywanie danych w pamięci głównej (operacyjnej)

- Dane liczbowe przechowywane w pamięci komputera mają długość 1, 2, 4, … bajtów. W przypadku danych o rozmiarze 2, 4, … bajtów pożądane jest by znajdowały się one w pamięci pod adresem podzielnym przez ich długość liczoną w bajtach — takie ulokowanie danych pozwala na uzyskanie najszybszego dostępu do nich.
- Jeśli adres danej jest podzielny przez jej długość (np. liczba 4-bajtowa została zapisana w pamięci pod adresem 456), to mówimy, że stosowane jest *wyrównanie naturalne*.

POLITECHNIKA GDAŃSKA

Kodowanie znaków (1)

- Współczesne komputery posiadają zdolność przechowywania i przetwarzania danych tekstowych — niezbędne jest więc ustalenie sposobów kodowania znaków używanych w tekstach w postaci odpowiednich ciągów zer i jedynek.
- Podobny problem kodowania pojawił się kilkadziesiąt lat wcześniej w komunikacji telegraficznej (dalekopisowej) — opracowano wówczas różne schematy kodowania znaków w postaci ciągów zer i jedynek.

Kodowanie znaków (2)

- W tej sytuacji w systemach komputerowych przyjęto kod opracowany w pierwszej połowie XX w. dla urządzeń dalekopisowych — około roku 1968 w USA ustalił się sposób kodowania znaków znany jako kod ASCII (ang. American Standard Code for Information Interchange).
- Kod ten obejmuje małe i wielkie litery alfabetu łacińskiego, cyfry, znaki przestankowe i sterujące (np. nowa linia)

POLITECHNIKA GDAŃSKA

Kod ASCII (1)

- Znaki w kodzie ASCII zapisywane są na 8 bitach, ale znaki *podstawowego kodu ASCII* (alfabet łaciński, znaki przestankowe i sterujące) wykorzystują tylko kody o wartościach 0 ÷ 127, spośród dopuszczalnego przedziału 0 ÷ 255.

A	0100 0001	41H
B	0100 0010	42H
C	0100 0011	43H
D	0100 0100	44H
E	0100 0101	45H
F	0100 0110	46H
...	...	...
Y	0101 1001	59H
Z	0101 1010	5AH

a	0110 0001	61H
b	0110 0010	62H
c	0110 0011	63H
d	0110 0100	64H
e	0110 0101	65H
f	0110 0110	66H
...	...	...
y	0111 1001	79H
z	0111 1010	7AH

POLITECHNIKA GDAŃSKA

Kod ASCII (2)

0	0011 0000	30H
1	0011 0001	31H
2	0011 0010	32H
3	0011 0011	33H
— — — — — —		
8	0011 1000	38H
9	0011 1001	39H

!	0010 0001	21H
"	0010 0010	22H
#	0010 0011	23H
$	0010 0100	24H
— — — — —		
{	0111 1011	7BH
|	0111 1100	7CH

POLITECHNIKA GDAŃSKA

Kod ASCII (3)

- Na bazie *podstawowego kodu ASCII* (kody o wartościach 0 ÷ 127) zaprojektowano wiele *kodów rozszerzonych*, w których wykorzystano także kody o wartościach z przedziału 128 ÷ 255
- W kodach rozszerzonych pierwsze 128 pozycji jest identyczne, jak w podstawowym kodzie ASCII, a następne 128 pozycji zawiera znaki alfabetów narodowych, symbole matematyczne, itp.

POLITECHNIKA GDAŃSKA

Kod ASCII (4)

- Istnieje wiele kodów rozszerzonych ASCII; w Polsce najbardziej znane są:
  - Windows 1250 (Microsoft CP 1250)
  - ISO 8859–2
  - Latin 2
  - Mazovia (wyszedł z użycia)

POLITECHNIKA GDAŃSKA

Kod ASCII (5)

Standard kodowania	Znak			
	a	ą	A	Ą
kody znaków podano w zapisie szesnastkowym				
Latin 2	61	A5	41	A4
Windows 1250	61	B9	41	A5
ISO 8859-2	61	B1	41	A1
Mazovia	61	86	41	8F
Unicode: format UTF – 16 (mniejsze niżej / little endian)	61 00	05 01	41 00	04 01
Unicode: format UTF – 16 (mniejsze wyżej / big endian)	00 61	01 05	00 41	01 04
Unicode: format UTF–8	61	C4 85	41	C4 84

POLITECHNIKA GDAŃSKA

Unicode (1)

- Kodowanie znaków na 8 bitach okazało się niewystarczające do przechowywania znaków narodowych krajów europejskich, i tym bardziej niewystarczające dla alfabetów krajów dalekiego wschodu — pojawiła się konieczność kodowania na większej liczbie bitów.
- Prace nad wprowadzeniem uniwersalnego zestawu podjęto na początku lat 90. ubiegłego stulecia — prace te podjęła zarówno Międzynarodowa Organizacja Normalizacyjna ISO, jak i konsorcjum zrzeszające producentów oprogramowania.

POLITECHNIKA GDAŃSKA

Unicode (2)

- Wynikiem prac obu instytucji było zdefiniowanie dwóch zestawów znaków:
  - UCS — Universal Character Set (ISO 10646).
  - Unicode (konsorcjum producentów).
- Znaki obu standardów są identyczne, ale obie instytucje wydają odrębne dokumenty, a także występują inne, drobne różnice.
- W dalszej części wykładu omawiany *uniwersalny zestaw znaków* określać będziemy terminem **Unikod** (lub Unicode).
- W tej sytuacji okazało się konieczne wprowadzenie kodowania 16-bitowego — wówczas wyłonił się standard znany jako Unicode
- W odniesieniu do znaków z podstawowego kodu ASCII, w Unikodzie rozszerzono ich kody binarne z 8 do 16 bitów poprzez „dopisanie” 8 zer z lewej strony

Unicode (3)

- W systemie Unicode każdemu znakowi przypisana jest wartość liczbowa określana jako *punkt kodowy* (ang. code point), przy czym dodatkowo każdemu znakowi przyporządkowana jest także nazwa, nie jest natomiast określony kształt drukowanego znaku.
- Przykładowo:
  - wielka litera A ma przypisany kod liczbowy, który zapisywany jest w postaci U+0041 (szesnastkowo), a oficjalna nazwa brzmi „LATIN CAPITAL LETTER A”.
  - litera ą ma przypisany kod U+0105, a oficjalna nazwa brzmi „LATIN SMALL LETTER A WITH OGONEK”.

Unicode (4)

- Wartości punktów kodowych dla znaków z podstawowego kodu ASCII (małe i wielkie litery alfabetu łacińskiego, cyfry, znaki przestankowe) są identyczne z wartościami odpowiednich kodów ASCII, aczkolwiek zazwyczaj zapisywane są w postaci liczb 16-bitowych, np. kod litery A w zapisie szesnastkowym ma postać:

  ASCII 61

  Unikod U+0061

Unicode (5)

- Punkty kodowe Unikodu zapisywane są w postaci liczb złożonych z 4, 5 lub 6 cyfr w zapisie szesnastkowym.
- Zazwyczaj stosowany jest zapis, w którym wartość liczbowa poprzedzona jest znakami U+, co należy traktować jako informację, że jest to wartość punktu kodowego podana w zapisie szesnastkowym.
- Kody kilku początkowych liter alfabetu polskiego zawiera tabela (wartości podane są w kodzie szesnastkowym).

Znak	Kod
a	0061
A	0041
ą	0105
Ą	0104
b	0062
B	0042
c	0063
C	0043
ć	0107
Ć	0106

POLITECHNIKA GDAŃSKA

Znaki grupy BMP (1)

- Przypuszcza się, że liczba różnych znaków, które używane są na świecie, wynosi ponad milion — wynika stąd konieczność przyjęcia sposobu kodowania tych znaków wykorzystujących co najmniej 21 bitów.
- Kierując się tym oszacowaniem, w standardzie Unikod udostępniono 1 114 112 punktów kodowych (w przedziale od 0 do 1 114 111). Punkty te tworzą *przestrzeń kodową* Unikodu.

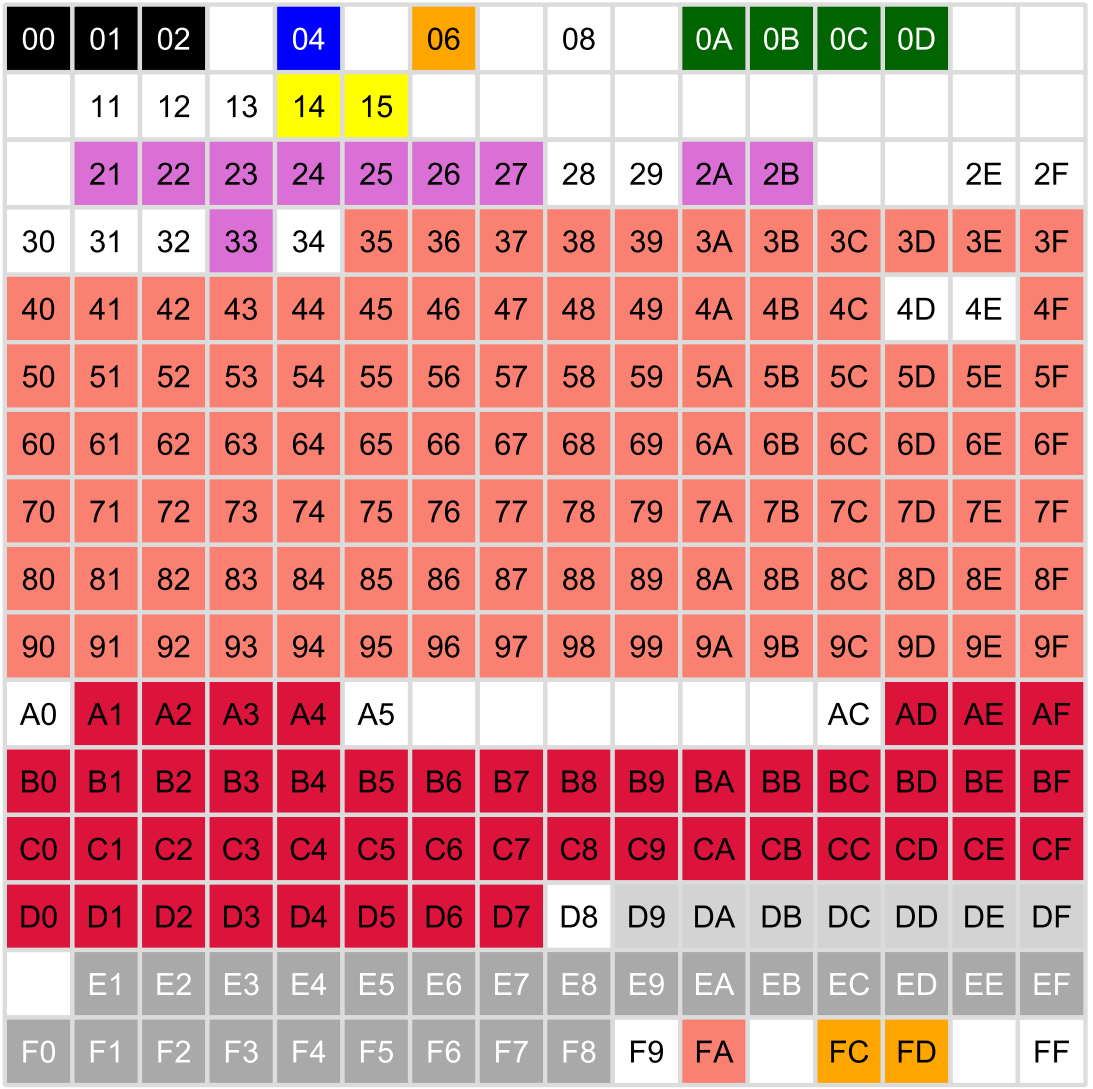
POLITECHNIKA GDAŃSKA

Znaki grupy BMP (2)

- W wersji 15.1 standardu opublikowanej w wrześniu 2023 r. zdefiniowano 149 186 znaków.
- Większość powszechnie używanych znaków jest przyporządkowana punktom kodowym o wartościach nie przekraczających 65 535 — zbiór ten, obejmujący kody od 0 do 65535, oznaczany jest skrótem BMP (ang. Basic Multilingual Plane). Zatem wartości punktów kodowych ze zbioru BMP dają się przedstawić w postaci 16-bitowych liczb binarnych.

POLITECHNIKA GDAŃSKA

Znaki grupy BMP (3)



- Latin script
- Non-Latin European scripts
- African scripts
- Middle Eastern and Southwest Asian scripts
- South and Central Asian scripts
- Southeast Asian scripts
- East Asian scripts
- CJK characters
- Indonesian and Oceanic scripts
- American scripts
- Notational systems
- Symbols
- Private use
- UTF-16 surrogates
- Unallocated code points

As of Unicode 9.0

POLITECHNIKA GDAŃSKA

Reprezentacja punktów kodowych Unikodu w pamięci komputera (1)

- Jeśli zamierzamy umieścić wartość punktu kodowego w pamięci komputera lub w pliku, to trzeba ustalić odpowiedni sposób kodowania dla używanego środowiska. Ponieważ niektóre wartości zajmują 21 bitów, wskazane byłoby zarezerwowanie na każdą wartość słowa 32-bitowego.
- Jeśli jednak uwzględnić podaną wyżej informację, że większość używanych znaków należy do zbioru BMP, gdzie punkty kodowe można zapisać na 16 bitach, to używanie słów 32-bitowych będzie powodować rozwlekłość kodowania.

POLITECHNIKA GDAŃSKA

Reprezentacja punktów kodowych Unikodu w pamięci komputera (2)

- Problem staje się jeszcze bardziej wyraźny, jeśli uwzględnić, że w zwykłych tekstach dominują litery alfabetu łacińskiego, które w kodzie ASCII zajmują 7 bitów.
- Z podanych powodów wprowadzono bardziej efektywne sposoby przechowywania znaków Unikodu w pamięci komputera — najczęściej używane są formaty UTF–8 i UTF–16 (Unicode Transformation Format).

UTF–8 (1)

- Przy zastosowaniu kodowania UTF–8 znaki podstawowe (znaki kodu ASCII z zakresu <1,127>) kodowane są jako 1-bajtowe, a inne znaki jako 2-bajtowe lub dłuższe.
- Przy kodowaniu UTF–8 obowiązują następujące reguły (wartości podane są w kodzie szesnastkowym):

  Znaki Unikodu o wartościach punktów kodowych 0000 do 007F są kodowane jako pojedyncze bajty o wartościach z przedziału 00 do 7F. Oznacza to, że pliki zawierające wyłącznie znaki z podstawowego zestawu ASCII mają taką samą postać zarówno w kodzie ASCII jak i w UTF–8.

UTF–8 (2)

- Wszystkie znaki Unikodu o wartościach punktów kodowych większych od 007F są kodowane jako sekwencja kilku bajtów, z których każdy ma ustawiony najstarszy bit na 1. Pierwszy bajt w sekwencji kilku bajtów jest zawsze liczbą z przedziału C0 do FD i określa ile bajtów następuje po nim. Wszystkie pozostałe bajty zawierają liczby z przedziału 80 do BF.
- Podana na następnym slajdzie tablica określa sposób kodowania UTF–8 dla różnych wartości punktów kodowych. Bity oznaczone *xxx..xx* zawierają reprezentację binarną kodu znaku.

UTF–8 (3)

Zakresy wartości punktów kodowych		Liczba kodo-wanych bitów	Reprezentacja w postaci UTF-8
od	do		
0000	007F	7	0xxxxxxx
0080	07FF	11	110xxxxx 10xxxxxx
0800	FFFF	16	1110xxxx 10xxxxxx 10xxxxxx
10000	1FFFFF	21	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

UTF–8 (4)

- Podana tablica zawiera kody UTF–8 kilku początkowych liter alfabetu języka polskiego (wartości podano w postaci liczb szenastkowych):

Znak	Kod UTF-8
a	61
A	41
ą	C4 85
Ą	C4 84
b	62
B	42
c	63
C	43
ć	C4 87
Ć	C4 86

POLITECHNIKA GDAŃSKA

Przykład kodowania UTF–8

- Litera ą w Unikodzie ma przypisaną wartość:

$$(0105)_{16} = (0000\ 0001\ 0000\ 0101)_2$$

- Ponieważ wartość ta należy do przedziału <0080,07FF>, więc reprezentacja UTF–8 zostanie wyznaczona wg schematu:

110*xxxxx* 10*xxxxxx*

- Zauważmy, że wszystkie liczby binarne z przedziału <0080,07FF> dają się zapisać na 11 bitach (w szczególności 07FF = 111 1111 1111) — zatem z podanej liczby $(0105)_{16}$ bierzemy tylko 11 najmłodszych bitów, które wstawiamy w miejsce symboli *xxx..xx* i otrzymujemy:

110 00100 10 000101 (= C4 85)

UTF–16 (1)

- Kodowanie UTF–16 przeznaczone jest do reprezentacji znaków Unikodu w środowiskach lub kontekstach ukierunkowanych na słowa 16-bitowe.

	kod UTF-16
a	0061
A	0041
ą	0105
Ą	0104
b	0062
B	0042

- W przypadku znaków z grupy BMP (kody od 0H do FFFFH), kod UTF–16 jest identyczny z wartością punktu kodowego. Obok podano przykładowe kody kilku liter alfabetu języka polskiego w formacie UTF-16 (zapis szesnastkowy).

UTF–16 (2)

- Dla znaków z przedziału od 10000H do 10FFFFH (nie należących do BMP) stosuje się dwa słowa 16-bitowe.
- Wartość z przedziału od 10000H do 10FFFFH zostaje najpierw pomniejszona o 10000H.
- W rezultacie pojawia się wartość 20-bitowa, z której 10 najstarszych bitów wpisywana jest do pierwszego słowa (pole *xxxxxxxxxx*), a pozostałe 10 bitów wpisywanych jest do drugiego słowa (pole *yyyyyyyyyy*), tak jak pokazano poniżej.

110110 xxxxxxxxxx	110111 yyyyyyyyyy

UTF–16 (3)

- Przykładowo, symbolowi CAT (kot) przyporządkowano wartość punktu kodowego 1F408 (zazwyczaj podawaną w postaci U+1F408).

- W celu uzyskania kodu UTF-16 odejmujemy najpierw liczbę 10000 (liczby w zapisie szesnastkowym i dwójkowym)

$$(1F408)_{16} – (10000)_{16} = (0F408)_{16} =$$
$$= (0000111101\ 0000001000)_2$$

UTF–16 (4)

- Starsze 10 bitów poprzedzamy ciągiem $(110110)_2$ i otrzymujemy ciąg 16-bitowy:

$$1101\ 1000\ 0011\ 1101 = (D83D)_{16}$$

- Z kolei młodsze 10 bitów poprzedzamy ciągiem $(110111)_2$ i otrzymujemy ciąg 16-bitowy:

$$(1101\ 1100\ 0000\ 1000)_2 = (DC08)_{16}$$

POLITECHNIKA GDAŃSKA

UTF–16 (5)

- Omawiany znak o kodzie U+1F408 można wyświetlić na ekranie za pomocą funkcji MessageBoxW (począwszy od Windows 8)

- W zapisie asemblerowym wymaga to także umieszczenia w sekcji danych programu dwóch słów 16-bitowych w postaci:

```
znaki          dw   0D83DH, 0DC08H
```

- W tabeli po prawej stronie pokazano reprezentację tablicy znaki w pamięci komputera (w postaci szesnastkowej).

k+3	DC
k+2	08
k+1	D8
k+0	3D

POLITECHNIKA GDAŃSKA

UTF–16 (6)

- Warto dodać, że w Unikodzie nie przyporządkowano jakiemukolwiek 16-bitowemu znakowi kodu zaczynającego od ciągu bitów 11011 — kody te zostały zarezerwowane do tworzenia omawianych par 16-bitowych (ang. surrogate pair). Innymi słowy, wartościom z przedziału <D800H, DFFFH> nie są przypisane żadne znaki.

POLITECHNIKA
GDAŃSKA

HISTORIA MĄDROŚCIĄ
PRZYSZŁOŚĆ WYZWANIEM