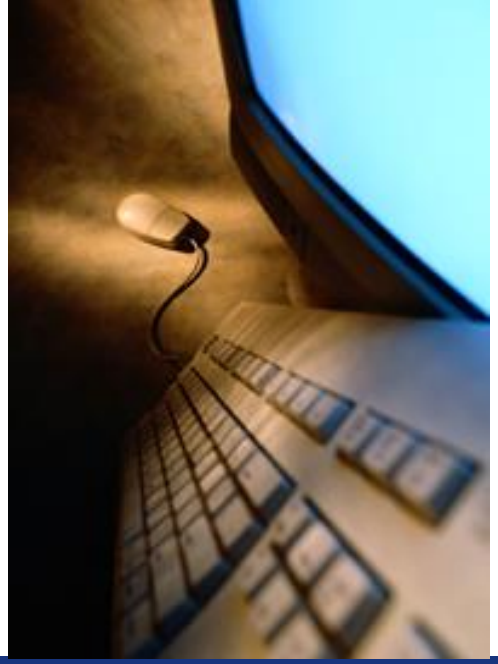
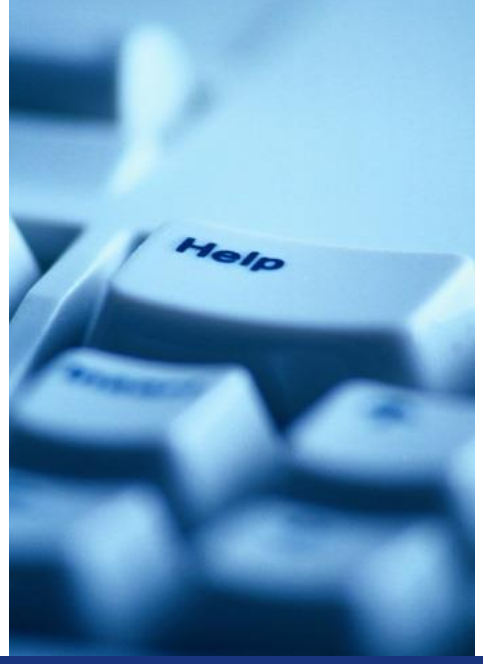
Instytut
Elektroniki
i Technik
Informacyjnych





Programowanie Niskopoziomowe

Wykład 1

Wstęp do języka asemblerowego

dr hab. inż. Piotr Kisała, prof. PL

Przysłowie chińskie

- „Co usłyszę zapomnę,
  - Co zobaczę zapamiętam,
    - Co zrobię – zrozumiem”

Wykłady i laboratoria

a)

90% ludzi to wzrokowcy

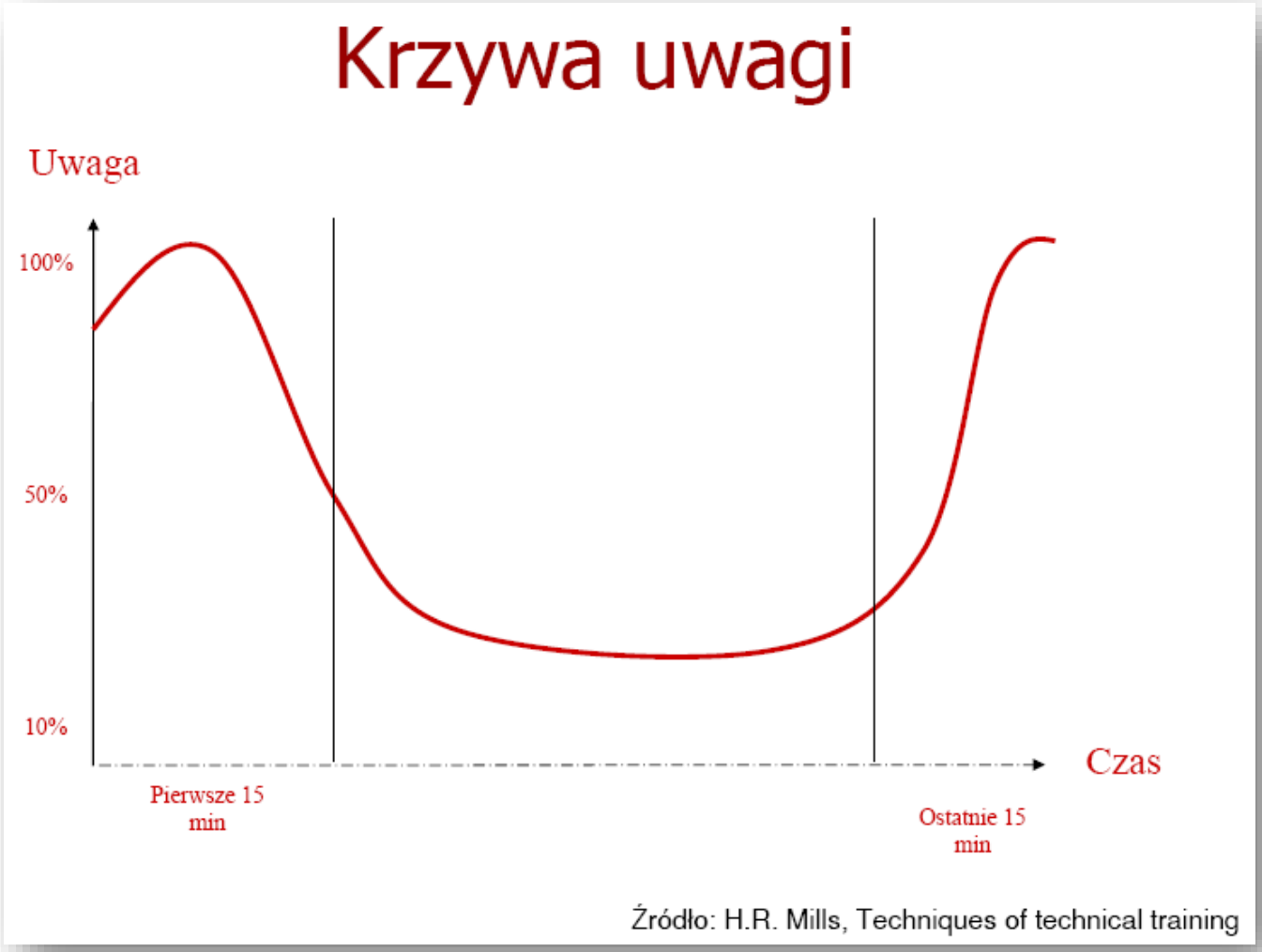
b)

Jeden obraz wart jest 1000 słów

c)

1000 obrazów nie jest warte jednego doświadczenia

Krzywa uwagi



Plan wykładu

1. x86
2. generacje procesorów x86
3. asembler x86
4. rejestry procesorów 80x86

Cele przedmiotu

C1	Zapoznanie z zasadami programowania niskopoziomowego
C2	Opanowanie języka asemblerowego
C3	Nabycie umiejętności posługiwania się asemblerem jako narzędziem kontroli komputera
C4	Poznanie architektury procesorów rodziny 80x86

Wymagania wstępne w zakresie wiedzy, umiejętności i innych kompetencji

1	Kompetencje uzyskiwane w zakresie przedmiotu Analiza matematyczna z algebrą
2	Znajomość podstawowej obsługi komputera

Efekty

Efekty kształcenia	
	W zakresie wiedzy:
EK 1	Znajomość architektury współczesnych komputerów
EK 2	Znajomość sposobów reprezentacji danych
EK 3	Znajomość trybów adresowania pamięci komputerów rodziny 80x86
EK 4	Znajomość niskopoziomowych struktur sterujących wykonaniem programu
	W zakresie umiejętności:
EK 5	Umiejętność posługiwania się operacjami arytmetycznymi
EK 6	Umiejętność deklaracji i odwoływania się do tablic wielowymiarowych
EK 7	Umiejętność tworzenia procedur i funkcji w języku asemblerowym
EK 8	Umiejętność posługiwania się operacjami zmiennoprzecinkowymi
EK 9	Umiejętność wykonywania operacji na plikach
EK 10	Umiejętność wykorzystywania podstawowych funkcji API
	W zakresie kompetencji społecznych
EK 11	Student zna potrzebę ciągłego pogłębiania i zdobywania wiedzy, jak też dzielenia się nią z innymi osobami

Treści programowe przedmiotu

	Treści programowe	Liczba godzin
W1	Wprowadzenie do rodziny procesorów 80x86	2
W2	Reprezentacja i organizacja danych	3
W3	Operacje arytmetyczne	3
W4	Dostęp do pamięci i jej organizacja	3
W5	Tryby adresowania procesorów 80x86	4
W6	Tablice wielowymiarowe	3
W7	Procedury i moduły	3
W8	Arytmetyka zmiennoprzecinkowa	3
W9	Niskopoziomowe struktury sterujące wykonaniem programu	3
W10	Rodzaje plików i operacja na plikach	3
W11	Wykorzystanie podstawowych funkcji API	
	Suma godzin:	30

Sposoby oceny

	Ocena formująca
F1	Krótkie pytania sprawdzające poziom zrozumienia podstawowych zagadnień podczas wykładu
F2	Wspólne omówienie zagadnień realizowanych na laboratoriach
	Ocena podsumowująca
P1	Kartkówki sprawdzające znajomość zagadnień wykonywanych na laboratoriach
P2	Ocena wykonania przeprowadzonych zajęć laboratoryjnych
P3	Dwa pisemne sprawdziany w połowie oraz na końcu prowadzonych wykładów

Formy oceny

	Na ocenę 2 (ndst)	Na ocenę 3 (dst)	Na ocenę 4 (db)	Na ocenę 5 (bdb)
EK 1	Nie zna generacji procesorów rodziny 80x86 i architektury komputera	Zna generacje procesorów rodziny 80x86	Zna i ogólnie charakteryzuje generacje procesorów rodziny 80x86 oraz zna architekturę komputera	**Zna, wymienia i wyczerpująco charakteryzuje generacje procesorów rodziny 80x86 oraz dokładnie charakteryzuje architekturę komputera**
EK 2	Nie zna rodzajów danych i sposobów ich prezentacji w asemblerze	Zna podstawowe rodzaje danych i sposoby ich prezentacji w asemblerze	Zna i ogólnie charakteryzuje podstawowe rodzaje danych i sposoby ich prezentacji w asemblerze	**Zna i wyczerpująco charakteryzuje wszystkie rodzaje danych i sposoby ich prezentacji w asemblerze**
EK 3	Nie zna trybów adresowania pamięci komputerów rodziny 80x86	Zna podstawowe tryby adresowania pamięci komputerów rodziny 80x86	Zna i charakteryzuje tryby adresowania pamięci komputerów rodziny 80x86	**Zna i wyczerpująco charakteryzuje wszystkie tryby adresowania pamięci komputerów rodziny 80x86**
EK 4	Nie zna niskopoziomowych elementów sterujących wykonaniem programu	Zna podstawowe niskopoziomowe elementy sterujące wykonaniem programu	Zna i charakteryzuje niskopoziomowe elementy sterujące wykonaniem programu	**Zna i wyczerpująco charakteryzuje wszystkie niskopoziomowe elementy sterujące wykonaniem programu**
EK 5	Nie potrafi wymienić ani zastosować w praktyce żadnej operacji arytmetycznej	Potrafi wymienić podstawowe operacje arytmetycznej	Potrafi wymienić i zastosować w praktyce podstawowe operacje arytmetyczne	**Potrafi szczegółowo wymienić i zastosować w praktyce wszystkie operacje arytmetyczne**
EK 6	**Nie potrafi deklarować i odwoływać się do tablic wielowymiarowych**	**Potrafi deklarować tablice wielowymiarowe**	**Potrafi deklarować i odwoływać się do tablic wielowymiarowych**	**Potrafi deklarować i odwoływać się do tablic wielowymiarowych, zna wszystkie ich rodzaje oraz układy**

Formy oceny

EK 7	**Nie potrafi zaprezentować struktury i nie potrafi tworzyć procedury i funkcji w języku asemblerowym**	**Potrafi przedstawić strukturę procedury i funkcji w języku asemblerowym**	**Potrafi przedstawić podstawową strukturę oraz tworzyć procedury i funkcje w języku asemblerowym**	**Potrafi przedstawić strukturę oraz tworzyć zaawansowane procedury i funkcje w języku asemblerowym**
EK 8	Nie potrafi wymienić ani zastosować w praktyce żadnej operacji zmiennoprzecinkowej	Potrafi wymienić podstawowe operacje zmiennoprzecinkowe	Potrafi wymienić i zastosować w praktyce podstawowe operacje zmiennoprzecinkowe	**Potrafi szczegółowo wymienić i zastosować w praktyce wszystkie operacje zmiennoprzecinkowe**
EK 9	Nie potrafi wykonywać operacji na plikach	Potrafi wymienić rodzaje plików i potrafi je scharakteryzować	Potrafi wymienić wszystkie rodzaje plików oraz potrafi wykonać podstawowe operacje na plikach	**Potrafi wymienić wszystkie rodzaje plików oraz potrafi szczegółowo wymienić i wykonać w praktyce podstawowe operacje na plikach**
EK 10	Nie potrafi wymienić ani zastosować w praktyce żadnej funkcji API	Potrafi wymienić podstawowe funkcje API	Potrafi wymienić i zastosować w praktyce podstawowe funkcje API	**Potrafi szczegółowo wymienić i zastosować w praktyce podstawowe funkcje API**
EK 11	**Nie przygotowuje się do zajęć**	**Przygotowuje się do zajęć w stopniu pozwalającym na wykonanie laboratoriów**	**Jest dobrze przygotowany do zajęć**	**Przygotowuje się do zajęć wykorzystując do tego również materiały dodatkowe, wyszukane samodzielnie**

Plan wykładów:

- Wykład 1
  - Wstęp do języka asemblerowego.
- Wykład 2
  - Reprezentacja danych.
- Wykład 3
  - Dostęp do pamięci i jej organizacja
- Wykład 4
  - Stałe, zmienne i typy danych
- Wykład 5
  - Procedury i moduły
- Wykład 6
  - Arytmetyka
- Wykład 7
  - Niskopoziomowe struktury sterujące wykonaniem programu

Plan wykładów

- Wykład 8
  - Pliki
- Wykład 9
  - Zaawansowane obliczenia w języku asemblerowym
- Wykład 10
  - Makrodefinicje i język czasu kompilacji
- Wykład 11
  - Manipulowanie bitami
- Wykład 12
  - Instrukcje MMX
- Wykład 13
  - Podstawy programowania Windows
- Wykład 14
  - Win32ASM – graficzny interfejs użytkownika GUI
- Wykład 15
  - Instrukcje MMX

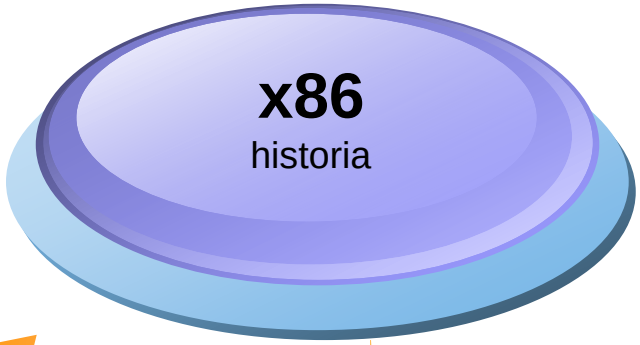
zaczynamy

x86

- x86 to rodzina architektur (modelów programowych) procesorów firmy Intel, należących do kategorii CISC, stosowana w komputerach PC, zapoczątkowana przez i wstecznie zgodna z 16-bitowym procesorem 8086, który z kolei wywodził się z 8-bitowego układu 8085. Nazwa architektury wywodzi się od nazw pierwszych modeli z tej rodziny, których numery kończyły się liczbą 86.

- Nazwa x86 w odniesieniu do modelu programowego procesorów dotyczy pierwszych procesorów tej rodziny - od 8086 do 286, które były układami o architekturze 16-bitowej.

Diagram

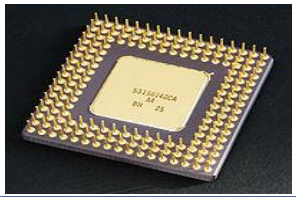


Procesor Intel 8086 z 1978

The first INTEL 16 bits CPU, the 8086

Procesor Intel 80486 DX z 1989 od spodu

Intel 80486DX CPU. Bottom view with gold plated pins.



Procesor Intel Core 2 z 2006

Intel Core 2 Duo E6600 "Conroe" micro processor .

x86-32 (IA-32)

- Drugie stadium rozwoju rodziny zapoczątkował w 1985 procesor 80386, w którym dokonano rozszerzenia słowa do 32 bitów, unikając jednak konieczności natychmiastowej wymiany wszystkich komputerów, poprzez zachowanie trybów zgodności z poprzednimi rozwiązaniami. Tak zmodyfikowaną architekturę (model programowy) x86 oznacza się zazwyczaj symbolem IA-32 (od Intel Architecture 32 bit) lub x86-32.

- Model ten z czasem został rozszerzony o nowe technologie, głównie wspierające zastosowania multimedialne, takie jak MMX czy SSE. Procesory oparte o ten model do dnia dzisiejszego stanowią większość procesorów używanych w komputerach na świecie.

x86-64 (AMD64)

- Trzecim stadium rozwoju procesorów wywodzących się z architektury x86 są procesory 64-bitowe. Architekturę (model programowy) takich procesorów, ze względu na wciąż zachowywaną wsteczną kompatybilność z pierwowzorami o architekturze x86, oznacza się symbolem x86-64. Rozwiązanie to zostało wprowadzone jednak przez firmę AMD, a dopiero później zaadoptowane przez Intela jako EM64T.

- Procesory o architekturze IA-64 nie należą do rodziny x86

Najważniejsze zmiany w kolejnych generacjach procesorów

Intel 8086

- układ 16-bitowy; posiadał jedynie tryb rzeczywisty; współpracował z koprocesorem 8087; posiadał 16-bitową szynę danych (lub ośmiobitową w tańszej wersji "SX" czyli 8088); nie posiadał MMU; składał się z dwóch jednostek - współpracy z pamięcią czyli kolejki oraz wykonawczą; zegar od 4,77 MHz do około 20 MHz (obecnie spotyka się wersje do 300MHz)

- Liczba tranzystorów: 0,029 mln.
- Maks. częstotliwość taktowania (w momencie wprowadzenia): 8 MHz
- Rok: 1978

Najważniejsze zmiany w kolejnych generacjach procesorów

Intel 80186

- poprawiona wersja 8086, o podobnych cechach jak poprzednik; posiadał nieco większą wydajność, kilkanaście nowych rozkazów i szybszy zegar.

- Maks. częstotliwość taktowania (w momencie wprowadzenia): 12 MHz
- Rok: 1982

Najważniejsze zmiany w kolejnych generacjach procesorów

Intel 80286

- druga generacja układów; procesor wciąż 16-bitowy, o zwiększonej do 24-bitów szynie adresowej; istotnie usprawniona wydajność; nowe rozkazy, nowy tryb pracy - chroniony (wspierający wielozadaniowość); adresowanie 16 MB RAM i 1 GB pamięci wirtualnej; zegar do 25 MHz

- Liczba tranzystorów: 0,134 mln.
- Maks. częstotliwość taktowania (w momencie wprowadzenia): 12,5 MHz
- Rok: 1982

Najważniejsze zmiany w kolejnych generacjach procesorów

Intel 80386

- trzecia generacja układów i równocześnie największa rewolucja w zakresie architektury x86, procesor 32-bitowy, o poszerzonych rejestrach wewnętrznych, szynie danych i adresowej (do 32 bitów); istotnie usprawniona wydajność; wiele nowych rozkazów; wbudowany MMU; zmieniony tryb chroniony; wprowadzony tryb wirtualny; pozwalający obsłużyć do 4 GB pamięci RAM.

- Liczba tranzystorów: 0,275 mln.
- Maks. częstotliwość taktowania (w momencie wprowadzenia): 20 MHz
- Rok: 1985

Procesor 80386 w stosunku do poprzednich przedstawicieli rodziny x86 posiada rozszerzone do 32-bitów rejestry ogólnego przeznaczenia (w stosunku do wersji 16-bitowych dodano do nazwy przedrostek "E": EAX, EBX, ECX, EDX, EBP, ESI, EDI, ESP), rejestr EIP (IP - ang. Instruction Pointer - wskaźnik bieżącej instrukcji) oraz rejestr flagowy EFLAGS. W procesorze dodano także rejestry kontrolne CRx

Najważniejsze zmiany w kolejnych generacjach procesorów

Intel i486

- dodano kilka nowych instrukcji; zwiększono wydajność jednostki stałoprzecinkowej oraz wbudowano i przeprojektowano koprocesor (FPU); dodatkowo wbudowano kontroler i pamięć cache poziomu L1.

- Liczba tranzystorów: 1,2 mln.
- Maks. częstotliwość taktowania (w momencie wprowadzenia): 25 MHz
- Rok: 1989

Najważniejsze zmiany w kolejnych generacjach procesorów

Pentium

- kolejna wersja procesora w architekturze x86, w procesorze tym powiększono cache L1, zmodernizowano FPU, dodano jednostkę przewidywania skoków, dodano kilka nowych instrukcji, zwiększono zewnętrzną magistralę danych do 64 bitów (procesor pozostał 32-bitowy) oraz szynę adresową do 36 bitów; procesor składa się z dwóch jednostek wykonawczych dość podobnych do 486 - większość czasów wykonania instrukcji pozostała bez zmian, procesor w określonych sytuacjach może jednak wykonywać dwa rozkazy równolegle.

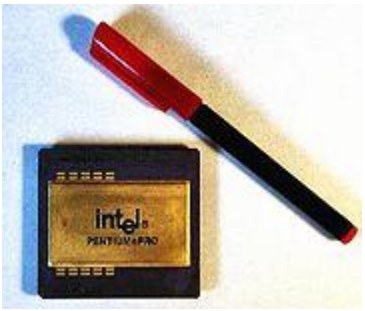
- Liczba tranzystorów: 3,1 mln.
- Maks. częstotliwość taktowania (w momencie wprowadzenia):
  66 MHz
- Rok: 1993

Najważniejsze zmiany w kolejnych generacjach procesorów

Pentium Pro

- była to **nowa generacja** architektury, nieformalnie oznaczana jako i686; procesor ten był dedykowany w szczególności do serwerów i wydajnych stacji roboczych, z punktu widzenia wewnętrznej struktury układu największa zmiana w koncepcji układu, pod względem mikroarchitektury układ ma wiele cech procesora RISC (choć zewnętrznie pozostaje zgodny z architekturą CISC); posiada 6 potoków; architektura Pentium Pro jest podstawą procesorów Pentium II i Pentium III, kompletnie zmieniona realizacja wewnętrzna - procesor wewnętrznie działa zupełnie inaczej niż wszystkie poprzednie, stanowiąc pod tym względem **pierwszą największą zmianę od czasu 8086**; L2 cache wbudowany w procesor jako osobny płatek krzemu zamknięty wraz z procesorem w jednej obudowie; 36 bitów szyny adresowej; 64 bity szyny danych; istotnie zmieniony sposób pracy szyny danych.

- Liczba tranzystorów: 5,5 mln.
- Maks. częstotliwość taktowania (w momencie wprowadzenia): 200 MHz
- Rok: 1995

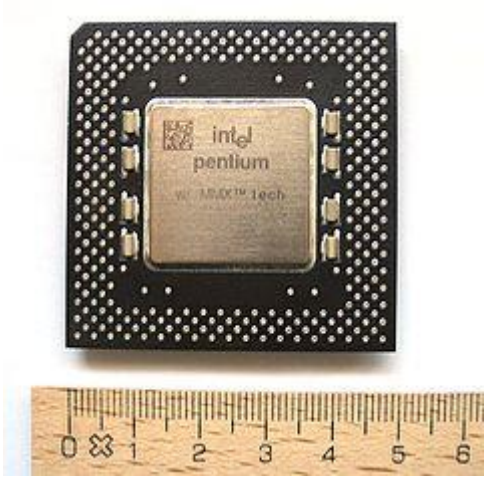


Najważniejsze zmiany w kolejnych generacjach procesorów

Pentium MMX

- nieco ulepszony Pentium, dodane rozkazy MMX.

- Liczba tranzystorów: 4,5 mln.
- Maks. częstotliwość taktowania (w momencie wprowadzenia): 233 MHz
- Rok: 1995



Najważniejsze zmiany w kolejnych generacjach procesorów

Pentium II

- połączenie Pentium Pro z Pentium MMX. W odróżnieniu od poprzednich wersji Pentium, Pentium II nie miał obudowy typu "socket" (gniazdo) ale "slot" (łącze krawędziowe). Takie rozwiązanie było wymagane z dwóch powodów: po pierwsze ułatwiało pozbycie się dużych ilości ciepła generowanych przez Pentium II, a po drugie umożliwiło odseparowanie cache L2 od procesora ale nadal pozwalało na bliskie położenie tych dwóch komponentów.

- Liczba tranzystorów: 7 mln.
- Maks. częstotliwość taktowania (w momencie wprowadzenia):
  266 MHz
- Rok: 1997

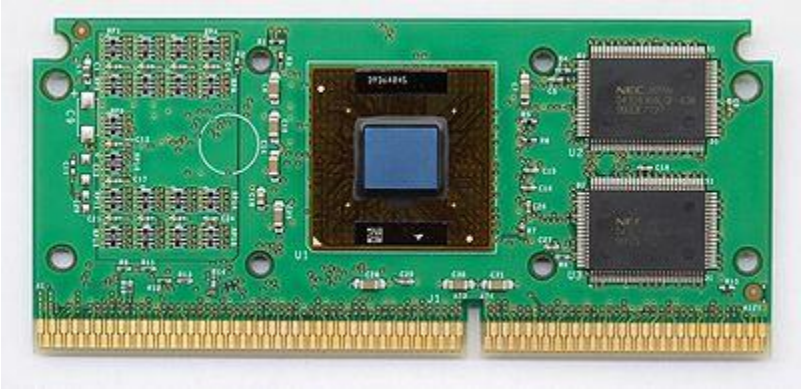
Najważniejsze zmiany w kolejnych generacjach procesorów

Pentium III

- wprowadzone rozkazy SSE*. Pentium III jest pod wieloma względami bardzo podobny do swego poprzednika czyli modelu Pentium II. Podwyższenie częstotliwość taktowania stanowi naturalny krok na drodze ewolucji w tej dziedzinie.

- Liczba tranzystorów: 8,2 mln.
- Maks. częstotliwość taktowania (w momencie wprowadzenia): 500 MHz
- Rok: 1999

*SSE (ang. "Streaming SIMD Extensions") jest nazwą zestawu instrukcji wprowadzonego w 1999 roku po raz pierwszy w procesorach Pentium III firmy Intel. SSE daje przede wszystkim możliwość wykonywania działań zmiennoprzecinkowych na 4-elementowych wektorach liczb pojedynczej precyzji (48 rozkazów). Ponadto wprowadzono jedenaście nowych rozkazów stałoprzecinkowych w zestawie MMX, a także dano możliwość wskazywania, które dane powinny znaleźć się w pamięci podręcznej.



Najważniejsze zmiany w kolejnych generacjach procesorów

Pentium 4

- kompletnie przeprojektowany procesor maksymalizujący technikę potokowości, dzięki czemu możliwe stało się osiąganie bardzo dużych częstotliwości zegara.

- Liczba tranzystorów: **42** mln.
- Maks. częstotliwość taktowania (w momencie wprowadzenia): 1500 MHz
- Rok: 2000

Najważniejsze zmiany w kolejnych generacjach procesorów

EM64T

- zmiana architektury x86 o znaczeniu prawie tak wielkim jak pojawienie się 80386; poszerzenie rejestrów do 64 bitów, nowe rejestry, nowe rozkazy, poszerzenie przestrzeni adresowej, nowe tryby pracy; pierwsza istotna zmiana architektury nie będąca autorstwa Intela.

- Maks. częstotliwość taktowania (w momencie wprowadzenia): 2200 MHz
- Rok: 2003

- Architektura jest obecna w nowszych modelach procesorów Pentium 4, Pentium D, Celeron D, Xeon, Pentium Dual Core, a także we wszystkich modelach procesorów Core 2, Core i5 oraz Core i7.

Najważniejsze zmiany w kolejnych generacjach procesorów

Pentium D

- dwurdzeniowy procesor całkowicie oparty o Pentium 4 po którym odziedziczył wiele wad. W odróżnieniu od innych procesorów wielordzeniowych w których rdzenie umieszczone są na jednej płytce krzemowej, pojedynczy układ Pentium D zawiera dwa osobno wyprodukowane i połączone ze sobą rdzenie Pentium 4 Prescott.

- Liczba tranzystorów: 230 mln.
- Maks. częstotliwość taktowania (w momencie wprowadzenia): 3200 MHz
- Rok: 2004

Najważniejsze zmiany w kolejnych generacjach procesorów

Intel Core 2

- Układ dwurdzeniowy. Procesory z serii Core 2 charakteryzują się zdecydowanie mniejszym zużyciem prądu niż procesory Pentium, a co za tym idzie wydzielają mniej ciepła. Procesory Core 2 odznaczają się stosunkowo wysokim współczynnikiem IPC (Instructions Per Cycle) - około 3,5. Oznacza to, że potrafią one w jednym cyklu rozkazowym wykonać średnio 3,5 rozkazu. Sporym ulepszeniem w stosunku do dwurdzeniowych procesorów Pentium 4 jest zastosowanie wspólnej pamięci cache dla obu rdzeni procesora. Dzięki temu uniknięto konieczności "mozolnego" uzgadniania zgodności pamięci podręcznych L2 (cache) w obu rdzeniach

- Liczba tranzystorów: 321 mln.
- Maks. częstotliwość taktowania (w momencie wprowadzenia): 3000 MHz
- Rok: 2006

Najważniejsze zmiany w kolejnych generacjach procesorów

Intel Core 2 Duo

- wydajniejsze układy dwurdzeniowe od Core 2. To ósma generacja mikroprocesorów firmy Intel w architekturze x86. Wykorzystana jest w niej nowa mikroarchitektura Intel Core, która zastąpiła architekturę NetBurst, na której oparte były wszystkie procesory tej firmy powstałe po 2000 roku

- Maks. częstotliwość taktowania (w momencie wprowadzenia): 3600 MHz
- Rok: 2006

Najważniejsze zmiany w kolejnych generacjach procesorów

Intel Core 2 Quad

- układy czterordzeniowe
- Maks. częstotliwość taktowania (w momencie wprowadzenia): 3600 MHz
- Rok: 2006

Hyper-Threading Technology

- Hyper-threading służy zwiększeniu wydajności obliczeń prowadzonych równolegle (czyli wykonywaniu wielu zadań jednocześnie) przez mikroprocesory. Dla każdego fizycznego rdzenia procesora system operacyjny przypisuje dwa procesory wirtualne (ang. virtual processors), a następnie dzieli obciążenie obliczeniami między nimi, jeżeli jest to możliwe. Hyper-threading wymaga nie tylko wsparcia ze strony systemu operacyjnego, ale również oprogramowania zoptymalizowanego specjalnie dla obsługi tej technologii. Intel zaleca wyłączanie jej, jeżeli używany jest system operacyjny bez takiej optymalizacji.
- Hyper-threading działa poprzez duplikowanie pewnych fragmentów procesora – tych, które przechowują stany procesów (architectural state) – ale nie jest to duplikowanie głównych zasobów wykonawczych. To pozwala procesorowi wykorzystującemu Hyper-threading być widocznym dla systemu operacyjnego jako dwa „logiczne” procesory, pozwalając mu na zaplanowanie wykonania dwóch wątków lub procesów jednocześnie

Najważniejsze zmiany w kolejnych generacjach procesorów

Intel Core i7

- bardzo wydajne układy czterordzeniowe. Generacja procesorów firmy Intel oparta na architekturze x86-64. Wykorzystuje ona mikroarchitekturę procesora o nazwie Nehalem. Jest to następca układów Intel Core 2 Duo i Intel Core 2 Quad z rdzeniem Penryn

- Maks. częstotliwość taktowania (w momencie wprowadzenia): 4200 MHz
- Rok: 2008

Modele czterordzeniowe

- **Intel**
  - niektóre Core i5 oraz Core i7
- **AMD**
  - Athlon II X4 oraz Phenom II X4 AMD

Modele sześciordzeniowe

- AMD
  - Phenom II X6
- Intel
  - Core i7 serii 9x0

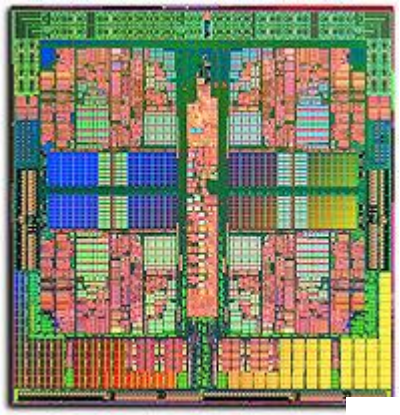
Przeznaczone do komputerów klasy desktop

Procesory ośmiordzeniowe

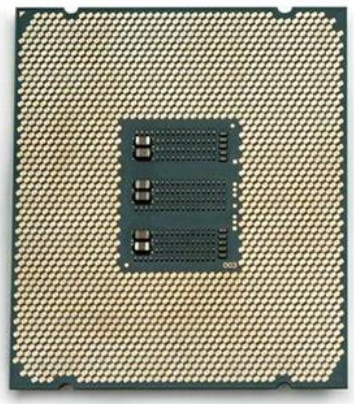
- AMD jako pierwsze wprowadziło na rynek procesory ośmiordzeniowe
- Intel również posiada taką konstrukcję w ofercie
  - i7 5960X o taktowaniu wynoszącym 3 GHz
  - jednostka wykonana w 22 nm procesie technologicznym, na bazie architektury Haswell
  - wydana w trzecim kwartale 2014 r

16 i 24-rdzeniowe

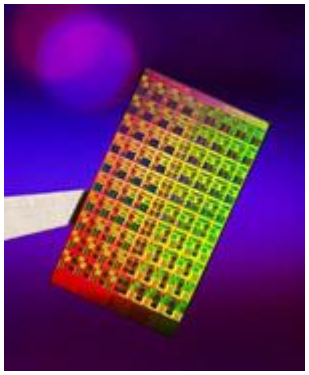
- Procesory do zastosowań serwerowych mają
  - 16 rdzenie
  - (AMD Opteron)



  - 24 rdzenie
  - (Intel Xeon Processor E7 v4
  - w 2016 roku)



80-rdzeniowy procesor Intela



programowalny procesor o wydajności dorównującej superkomputerom, mieszczący się w jednym 80-rdzeniowym układzie i zużywający mniej energii niż większość współczesnych urządzeń AGD.

Procesor powstał w wyniku inicjatywy "Tera-scale computing", która ma zapewnić wydajność rzędu teraflopów - bilionów operacji na sekundę - przyszłym komputerom osobistym i serwerom.

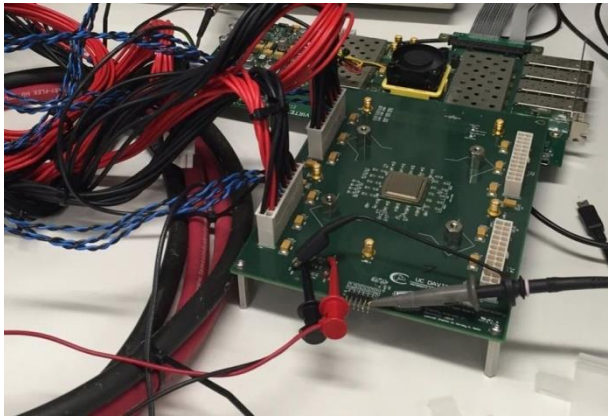
Wydajność rzędu teraflopów oraz możliwość przenoszenia terabajtów danych odegra kluczową rolę w przyszłych komputerach, które przy powszechnym dostępie do Internetu pozwolą na opracowanie nowych aplikacji do nauki i pracy zespołowej, a także na dostęp do cyfrowej rozrywki w komputerach PC, serwerach i urządzeniach przenośnych.

Na przykład sztuczna inteligencja, komunikacja wideo, fotorealistyczne gry, przeszukiwanie danych multimedialnych oraz rozpoznawanie mowy w czasie rzeczywistym kiedyś uważane za fantastykę naukową staną się codziennością.

Procesor 1000-rdzeniowy

- Procesor KiloCore
- zaprojektowany na Uniwersytecie Kalifornijskim - wyróżnia się on nie tylko ilością rdzeni, ale też efektywnością energetyczną

Procesor składa się z 621 mln tranzystorów i dysponuje 1000 niezależnych rdzeni – każdy może pracować z innym taktowaniem, które średnio wynosi 1,78 GHz (co pozwala uzyskać moc obliczeniową na poziomie **1,78 TFLOPS**)



Oprócz rekordowej liczby rdzeni, procesor może pochwalić się też świetną energooszczędnością, bo przy wydajności obniżonej do 115 GFLOPS, cały układ pobiera zaledwie **0,7 W i może być zasilany z pojedynczej baterii typu AA** – daje to ponad 100-krotnie lepszy wynik pod względem efektywności energetycznej

Asembler x86

- to język programowania z rodziny asemblerów do komputerów klasy PC, które posiadają architekturę głównego procesora zgodną z x86.

Zalety języka asemblerowego

Zalety języka asemblerowego

1

Mały rozmiar kodu

2

Duża szybkość działania:
Znając sztuczki optymalizacyjne, wiedząc które instrukcje są szybsze, eliminując zbędne instrukcje z pętli otrzymujemy kod wiele razy szybszy od tych napisanych w językach wysokiego poziomu.

3

Wiedza:

Zdobywasz wprost bezcenną wiedzę o tym, jak naprawdę działa komputer.

Mały rozmiar kodu

1a

Jako programista języka asembler wiesz co i gdzie w danej chwili się znajduje. Nie musisz ciągle przeładowywać zmiennych, co zabiera miejsce i czas. Możesz eliminować instrukcje, które są po prostu zbędne.

1b

Do kodu nie są dołączane żadne biblioteki z dziesiątkami procedur, podczas gdy ty używasz tylko jednej z nich. (marnotrawstwo)

1c

Jako znawca zestawu instrukcji, programista wie, które z bibliotek są krótsze.

Zalety języka asemblerowego

4

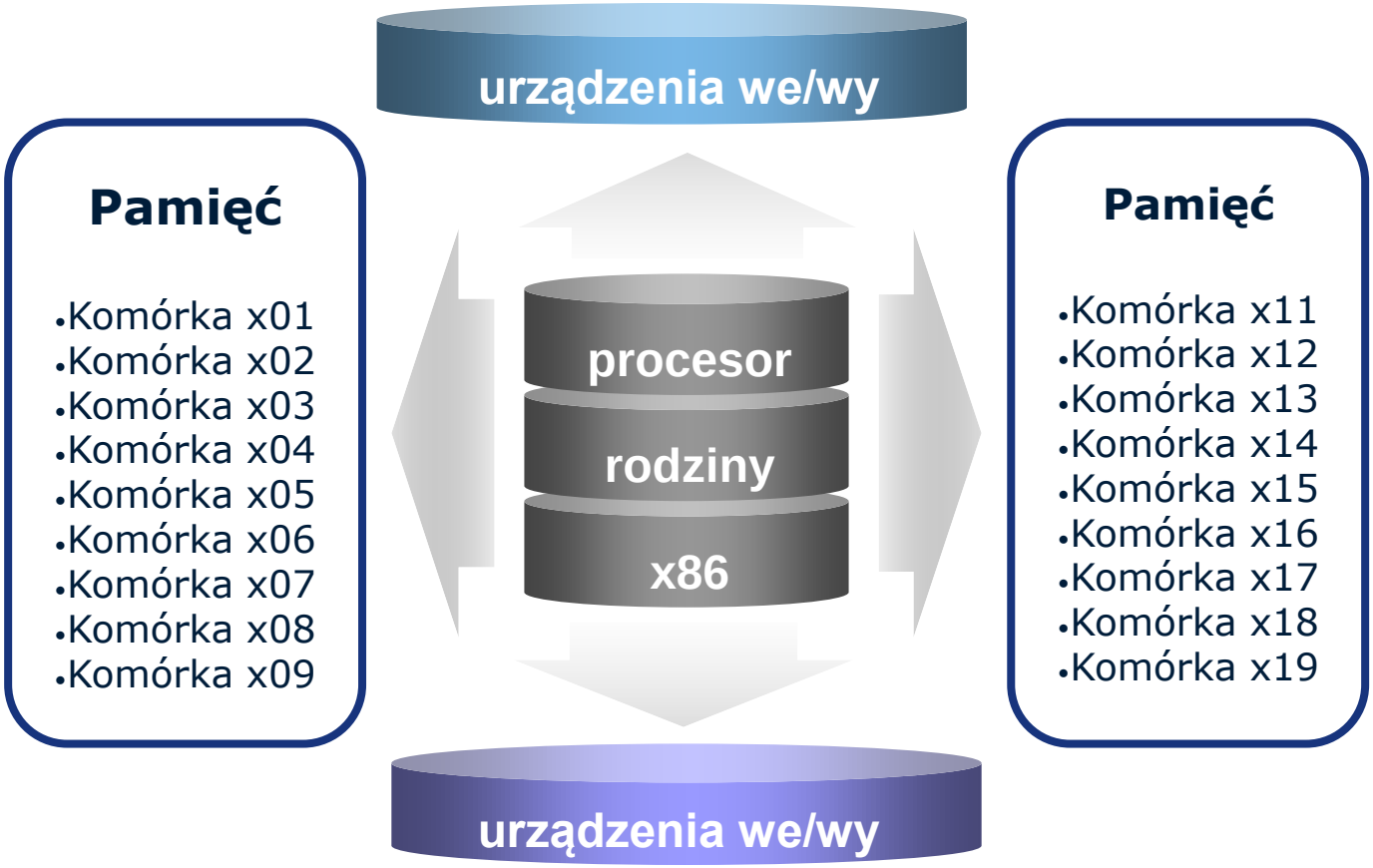
Możliwość tworzenia zmiennych o dużych rozmiarach, a nie ograniczonych do 4 czy 8 bajtów. W asemblerze zmienne mogą mieć dowolną ilość bajtów.

5

Wstawki asemblerowe:

W niektórych językach wysokiego poziomu istnieje możliwość wstawienia kodu napisanego w asemblerze wprost do programu.

Architektura x86

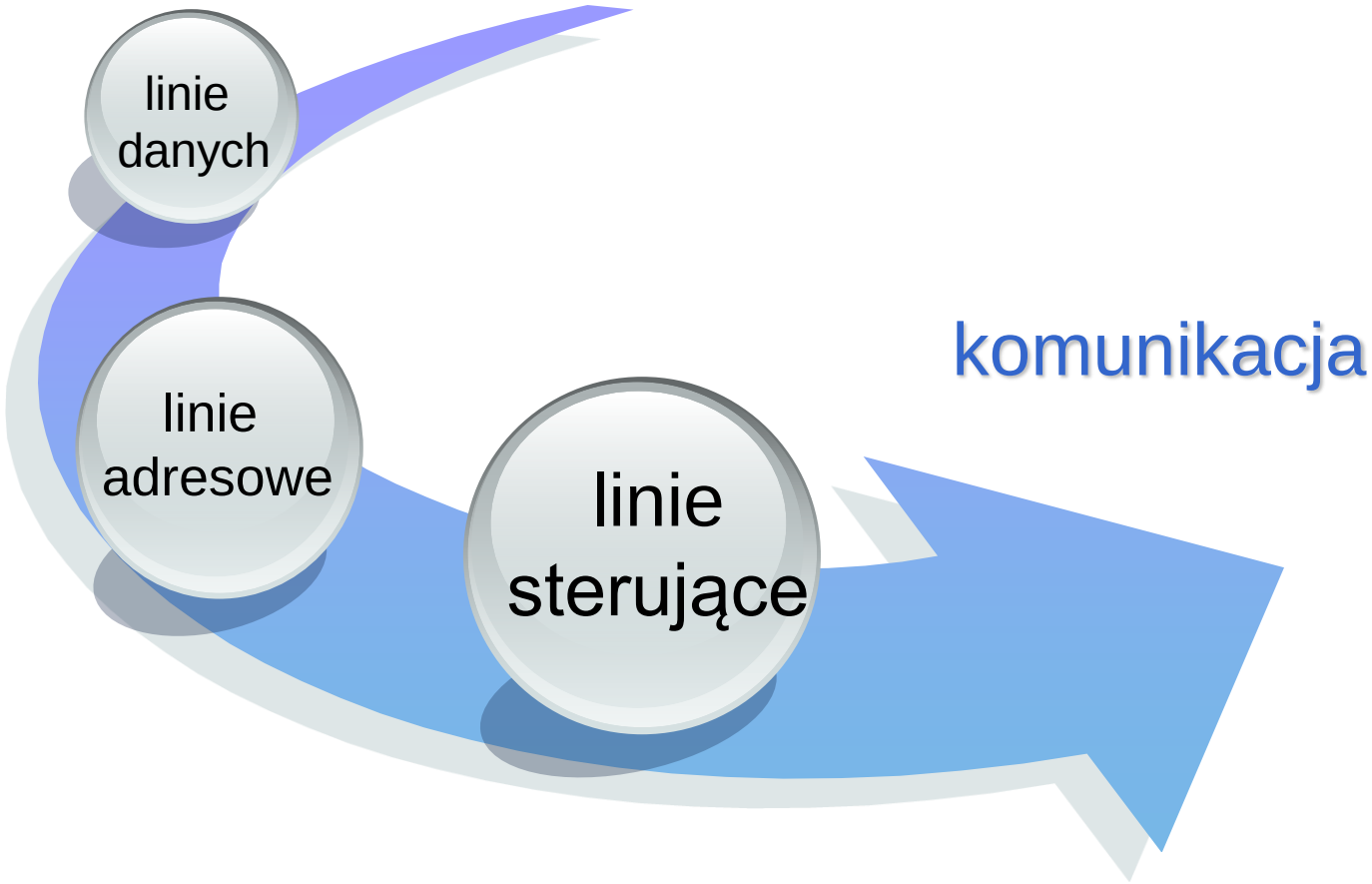


Architektura x86

Rodzina procesorów firmy Intel zaliczana jest do ogólnej kategorii **maszyn von Neumana**. Systemy komputerowe tej kategorii składają się z trzech głównych bloków funkcjonalnych: jednostki przetwarzającej, czyli **procesora** (CPU, *central processing unit*), **pamięci** oraz **urządzeń wejścia-wyjścia**.

Owe trzy komponenty są ze sobą połączone za pośrednictwem magistrali systemowej (ang. system bus) składającej się z linii danych, linii adresowych i linii sterujących.

Magistrala systemowa



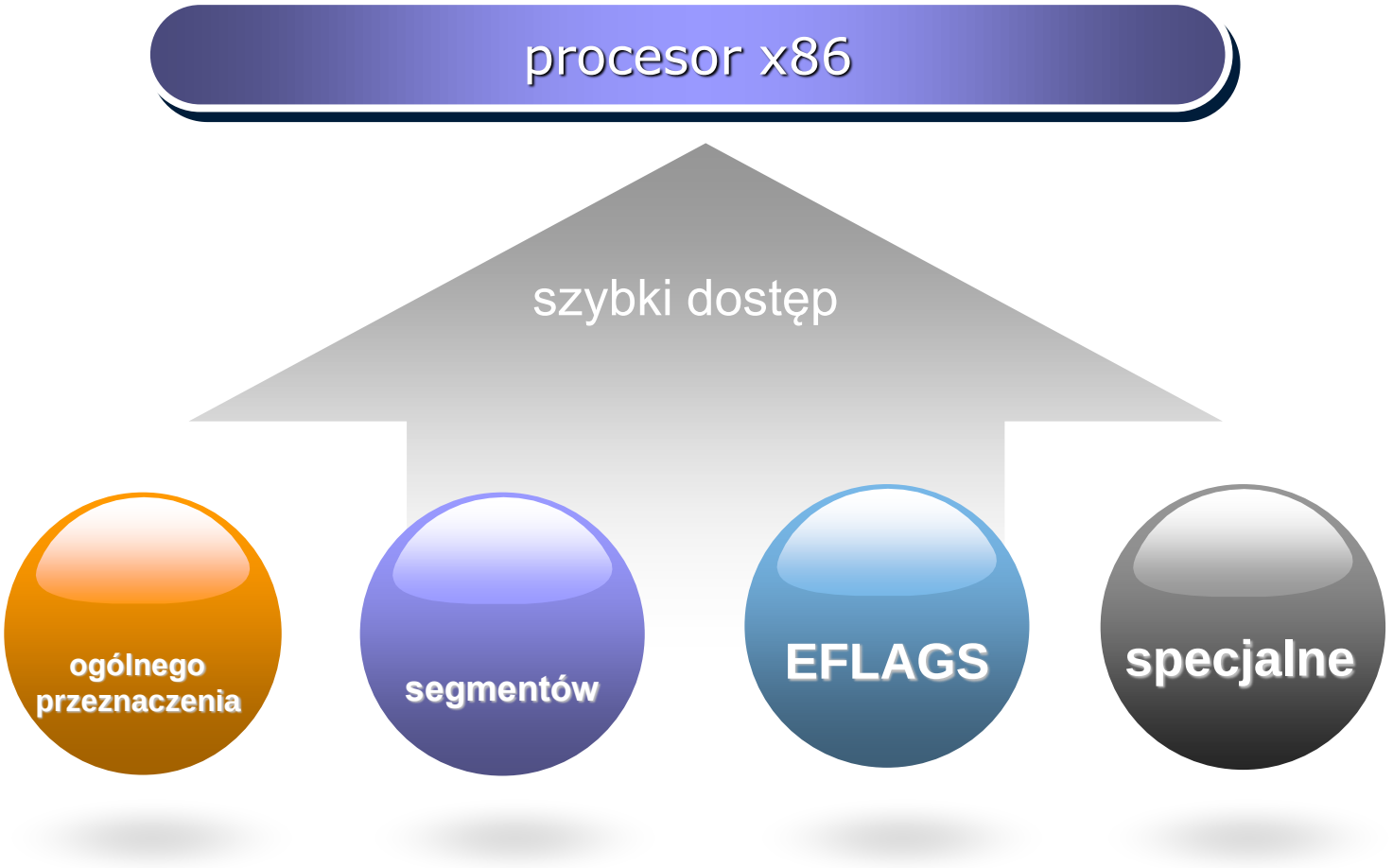
Magistrala systemowa – c.d.

linie danych
linie adresowe
linie sterujące

CPU komunikuj się z pamięcią i urządzeniami wejścia-wyjścia przez odpowiednie wysterowanie linii adresowych określające adres pamięci bądź numer urządzenia wejścia-wyjścia; każda z komórek pamięci i każde z urządzeń wejścia wyjścia dysponuje własnym, unikalnym adresem.

Następnie procesor wymienia dane z pamięcią lub urz. we-wy odpowiednio sterując stanami linii danych. Stan linii sterujących określa kierunek przesyłania danych (do bądź z pamięci, do bądź z urz. we-wy).

Rejestry x86



Rejestry procesorów 80x86

- rejestry ogólnego przeznaczenia
- rejestry specjalne trybu użytkownika
- rejestry segmentowe
- rejestry specjalne trybu nadzoru

Segmentowe – nie wykorzystywane w 32-bitowych syst. oper. (jak Windows, Linux).

Specjalne trybu nadzoru – wykorzystywane tylko przez twórców systemów operacyjnych, debugerów i innych specjalistycznych narzędzi systemowych.

Rejestry ogólnego przeznaczenia

- Sposób wykorzystania zależny wyłącznie od programisty.
- 8 rejestrów 32-bitowych:

  EAX, EBX, ECX, EDX, ESI, EDI, EBP, ESP.

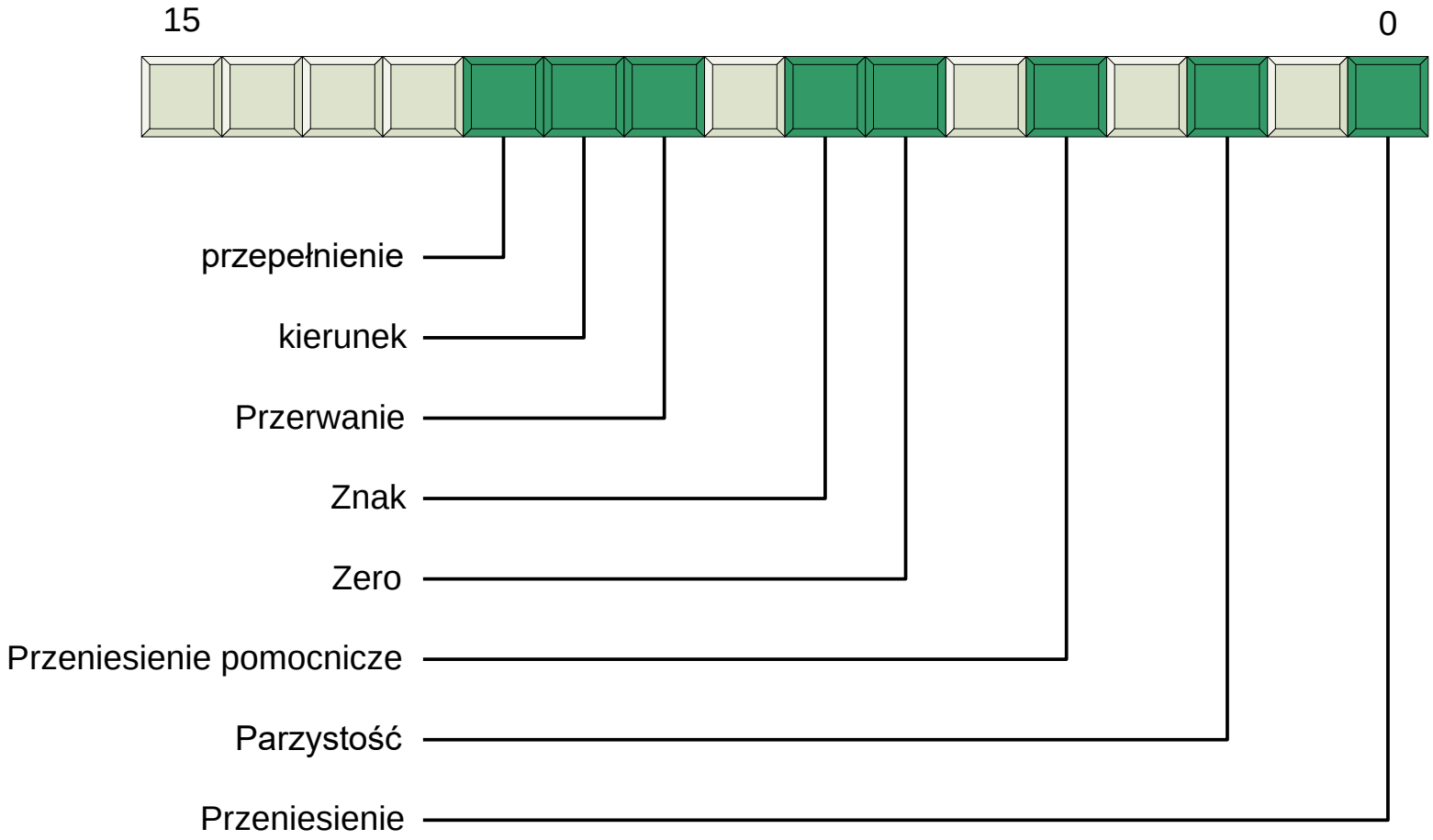
  Przedrostek E od extended, rozróżnia on rej. 32 od 16-bitowych
- 8 rejestrów 16-bitowych:

  AX, BX, CX, DX, SI, DI, BP, SP.
- 8 rejestrów 8-bitowych:

  AL, AH, BL, BH, CL, CH, DL, DH.

Rejestry ogólnego przeznaczenia

Rejestr EFLAGS



Rejestr EFLAGS

- **Większość bitów (znaczników) tego rejestru zarezerwowana jest dla trybu nadzoru (czyli dla kodu syst. oper.). Programistów interesuje jedynie 8 bitów tego rejestru.**

- **4 znaczniki są szczególnie ważne:**
  - przepełnienia
  - przeniesienia
  - znaku
  - zera

Funkcje rejestrów

Każda operacja angażuje rejestry. Aby dodać do siebie 2 wartości i umieścić ich sumę w trzeciej, należy załadować jeden ze składników do rejestru, dodać do niego **(w rejestrze)** drugi składnik sumy i dopiero potem wynik skopiować z rejestru do miejsca przechowywania sumy.

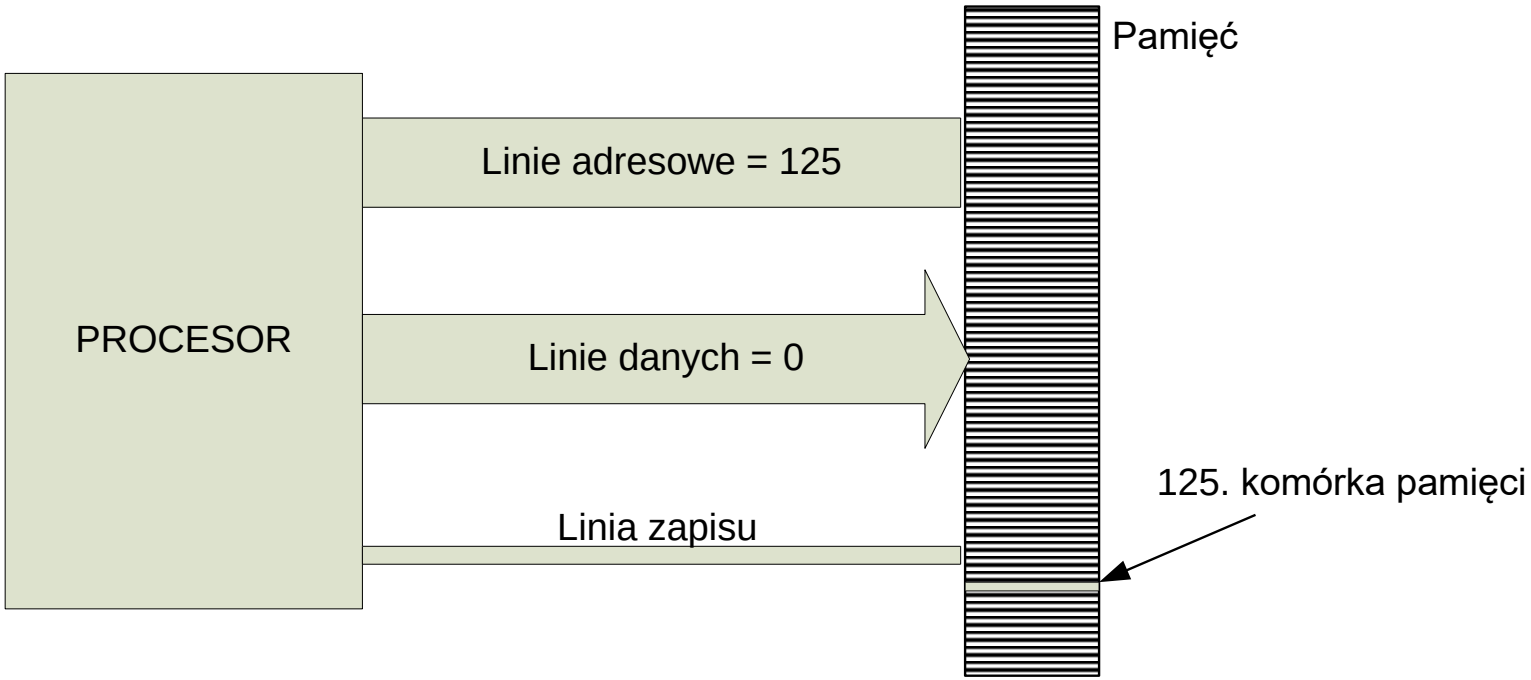
Rejestry stanowią bazę wszelkich obliczeń.

Obsługa pamięci

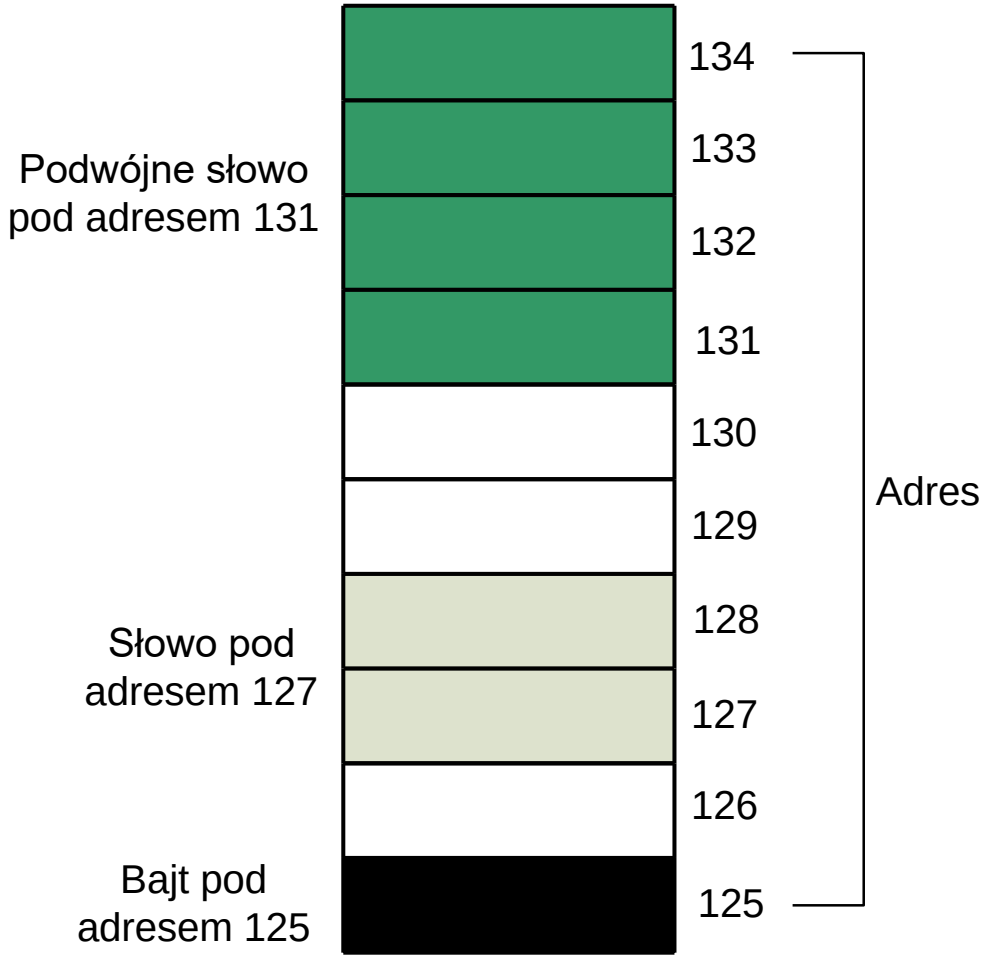
- Typowy procesor działający pod kontrolą 32-bitowego systemu operacyjnego może odwoływać się najwyżej do 2^{32} różnych adresów pamięci, czyli do nieco ponad czterech miliardów komórek pamięci – bajtów.
- Procesory 80x86 obsługują pamięć adresowaną bajtowo. Podstawową jednostką pamięci jest bajt, który wystarcza do zakodowania pojedynczego znaku bądź niewielkiej liczby całkowitej.
- Pamięć jest liniową tablicą bajtów. Adresem pierwszego bajta w tej tablicy jest 0, ostatni ma adres $2^{32} - 1$.

Operacja zapisu do pamięci

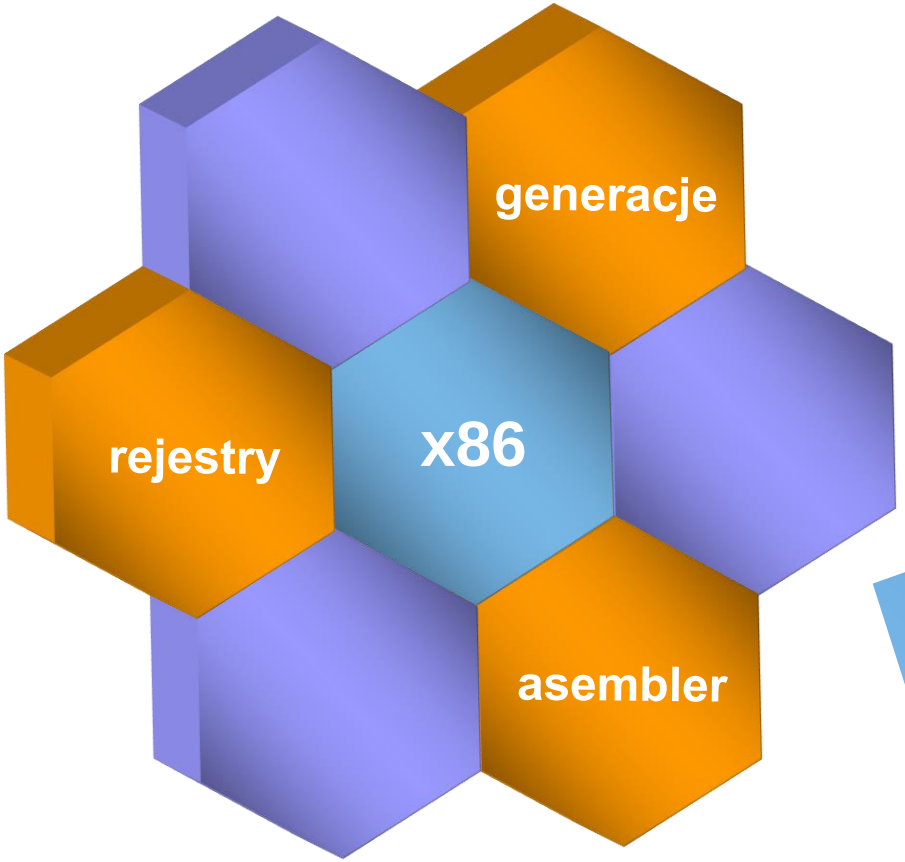
Aby zapisać do komórki pamięci o adresie 125 wartości 0 procesor umieszcza wartość zero na magistrali danych, wartość 125 na magistrali adresowej oraz ustawia linię zapisu (zapis to zwykle wyzerowanie).



Zapis/odczyt słowa i słowa podwójnego



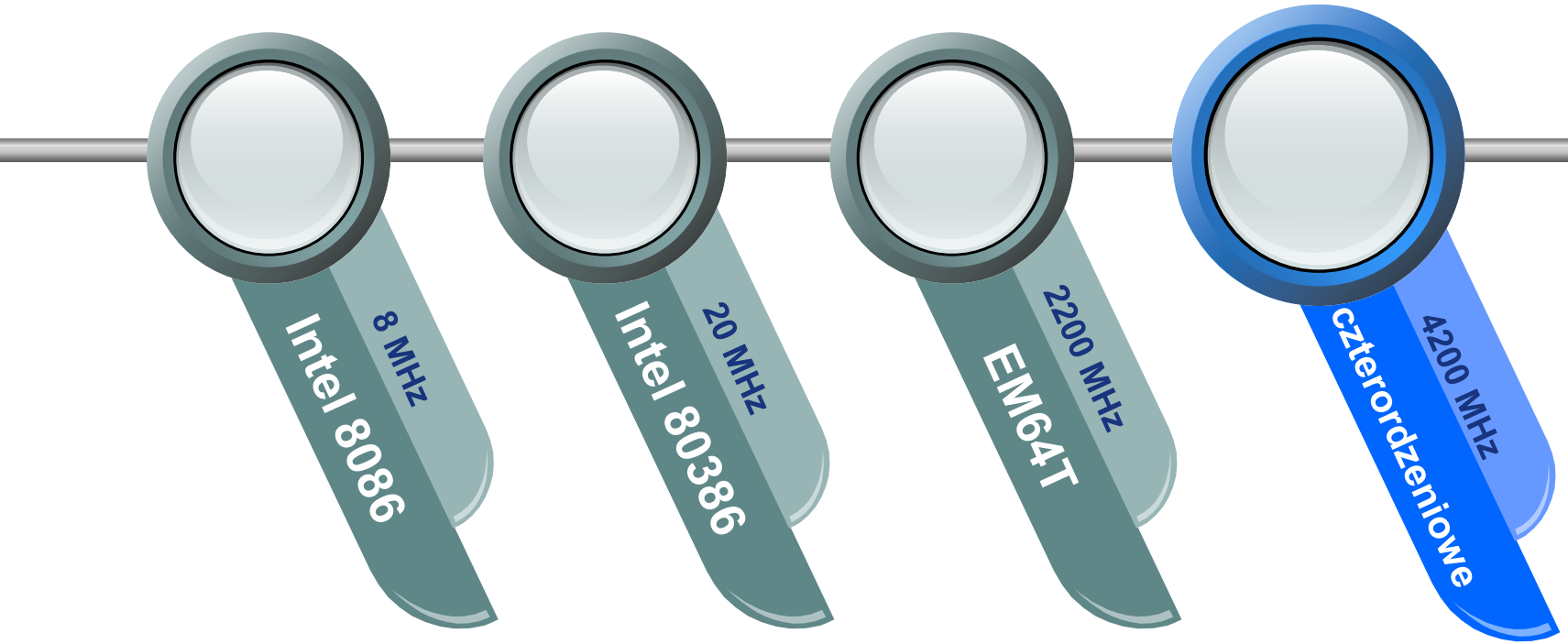
Podsumowanie



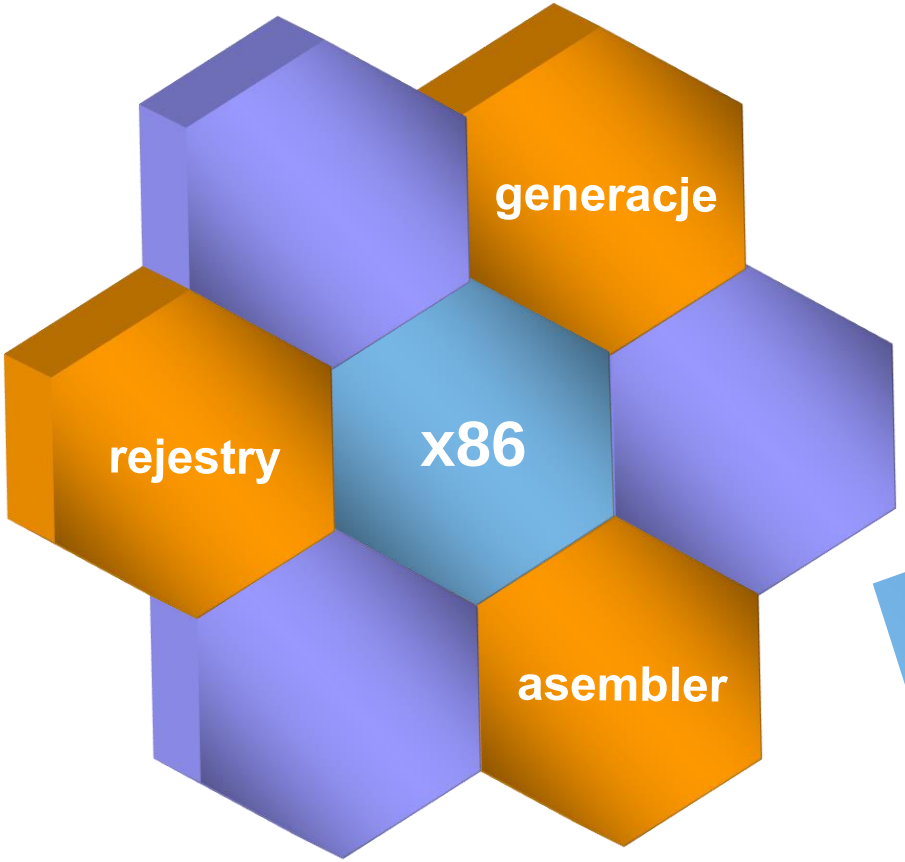
szybka powtórka

generacje x86 - kroki milowe

1978 → 1985 → 2003 → **2008**



Podsumowanie



szybka powtórka

Zalety języka asemblerowego

1

Mały rozmiar kodu

2

Duża szybkość działania:
Znając sztuczki optymalizacyjne, wiedząc które instrukcje są szybsze, eliminując zbędne instrukcje z pętli otrzymujemy kod wiele razy szybszy od tych napisanych w językach wysokiego poziomu.

3

Wiedza:

Zdobywasz wprost bezcenną wiedzę o tym, jak naprawdę działa komputer i możesz

Mały rozmiar kodu

1a

Jako programista języka asembler wiesz co i gdzie w danej chwili się znajduje. Nie musisz ciągle przeładowywać zmiennych, co zabiera miejsce i czas. Możesz eliminować instrukcje, które są po prostu zbędne.

1b

Do kodu nie są dołączane żadne biblioteki z dziesiątkami procedur, podczas gdy ty używasz tylko jednej z nich.
Co za marnotrawstwo.

1c

Jako znawca zestawu instrukcji, programista wie, które z bibliotek są krótsze.

Zalety języka asemblerowego

4

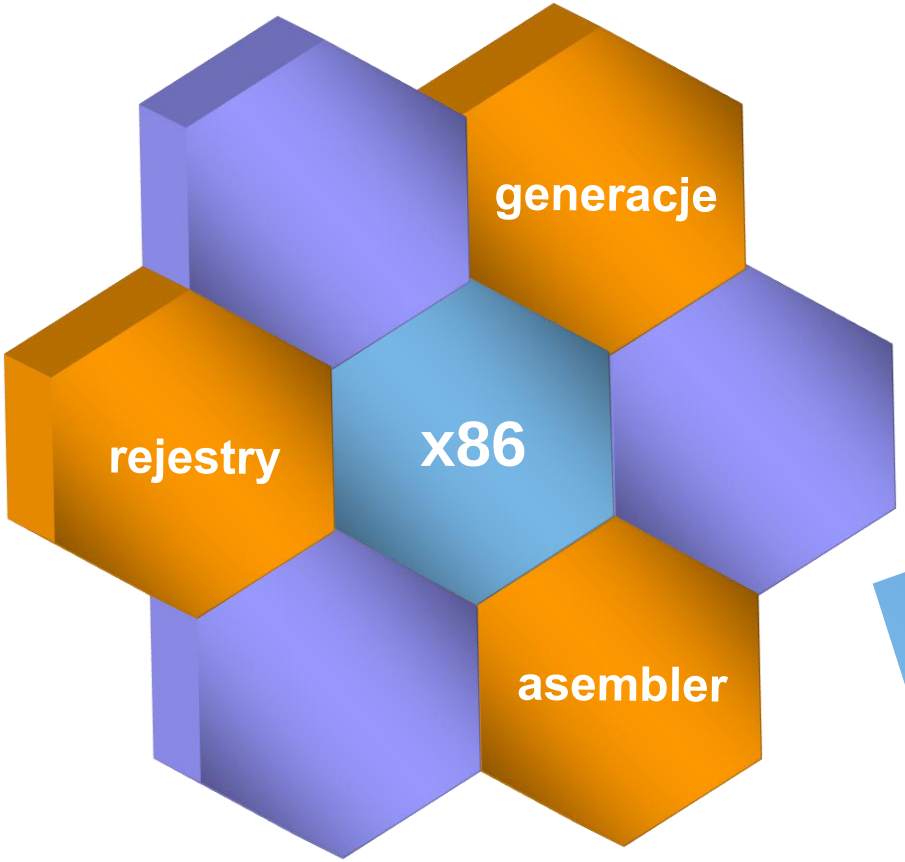
Możliwość tworzenia zmiennych o dużych rozmiarach, a nie ograniczonych do 4 czy 8 bajtów. W asemblerze zmienne mogą mieć dowolną ilość bajtów.

5

Wstawki asemblerowe:

W niektórych językach wysokiego poziomu istnieje możliwość wstawienia kodu napisanego w asemblerze wprost do programu.

Podsumowanie



szybka powtórka

Rejestry ogólnego przeznaczenia

- Sposób wykorzystania zależny wyłącznie od programisty.
- 8 rejestrów 32-bitowych:

  EAX, EBX, ECX, EDX, ESI, EDI, EBP, ESP.

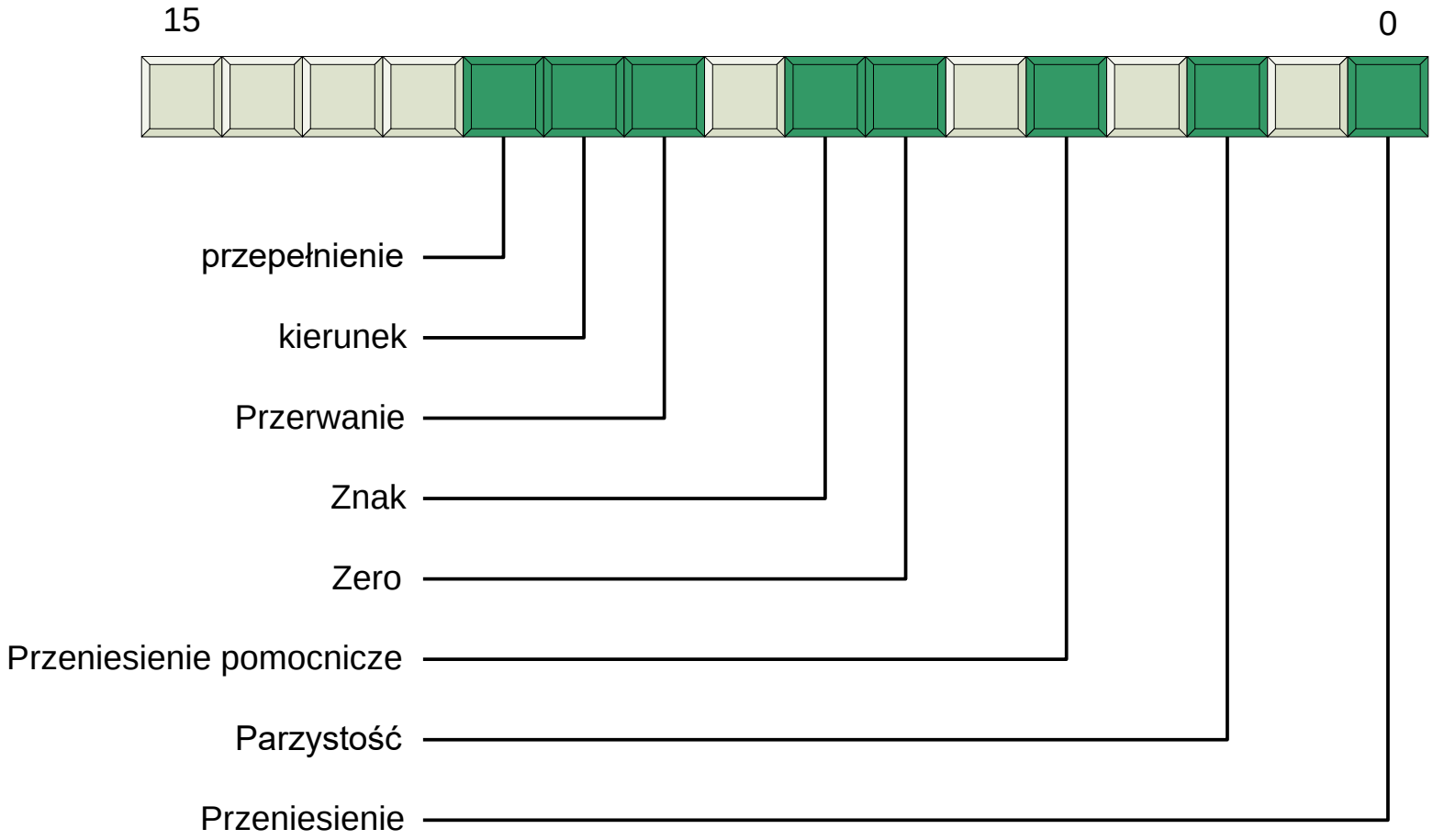
  Przedrostek E od extended, rozróżnia on rej. 32 od 16-bitowych
- 8 rejestrów 16-bitowych:

  AX, CX, DX, SI, DI, BP, SP.
- 8 rejestrów 8-bitowych:

  AL, AH, BL, BH, CL, CH, DL, DH.

Rejestry ogólnego przeznaczenia

Rejestr EFLAGS



Rejestr EFLAGS

- **Większość bitów (znaczników) tego rejestru zarezerwowana jest dla trybu nadzoru (czyli dla kodu syst. oper.). Programistów interesuje jedynie 8 bitów tego rejestru.**

- **4 znaczniki są szczególnie ważne:**
  - przepełnienia
  - przeniesienia
  - znaku
  - zera

Funkcje rejestrów

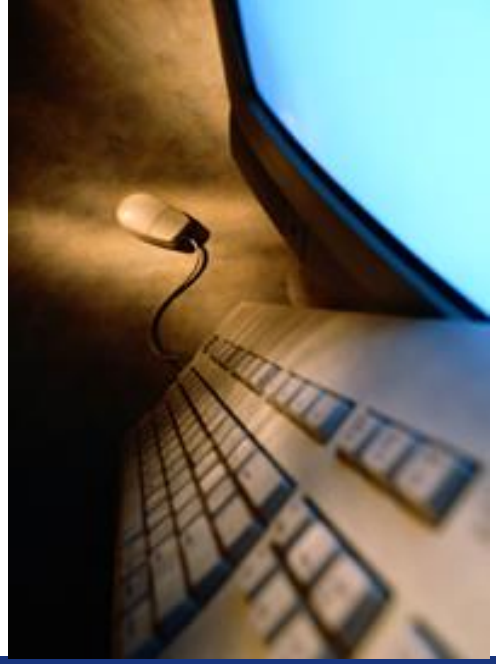
Każda operacja angażuje rejestry. Aby dodać do siebie 2 wartości i umieścić ich sumę w trzeciej, należy załadować jeden ze składników do rejestru, dodać do niego **(w rejestrze)** drugi składnik sumy i dopiero potem wynik skopiować z rejestru do miejsca przechowywania sumy.

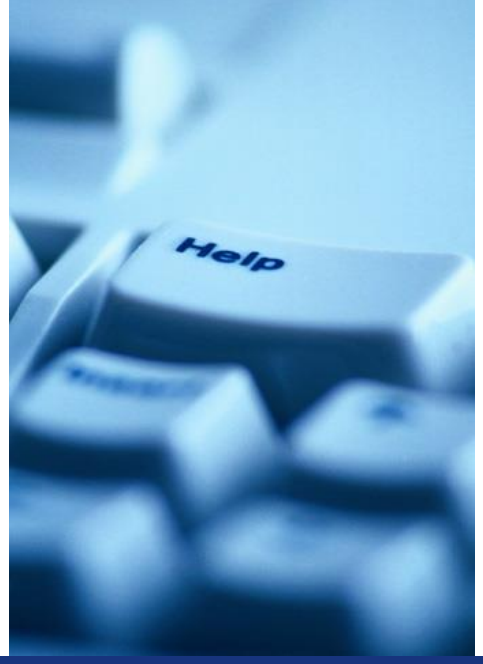
Rejestry stanowią bazę wszelkich obliczeń.

Źródła

Literatura do wykładu

- S. Kruk, „KURS PROGRAMOWANIA W JĘZYKU ASEMBLER” , wydawnictwo MIKOM 2001.
- B. Drozdowski, „Język asembler dla każdego”, 2008 r.
- R. Hyde, The Art. of Assembly Language”, 2004 r.
- S. Kruk, „ASEMBLER podręcznik użytkownika, 1999 r.
- G. Michałek, „Asembler MINIPRZEWODNIK”, wydawnictwo INFOLAND 2001.
- Materiały firmowe - dokumenty techniczne dostępne w sieci www - MIPS, Intel, AMD.
- Nawrocki J. R.: Programowanie komputerów.





Instytut Elektroniki i Technik Informacyjnych

Dziękuję za uwagę !

Piotr Kisała