

Zbigniew S. Szewczak

Podstawy Systemów Operacyjnych

Wykład 2

Ewolucja systemów komputerowych

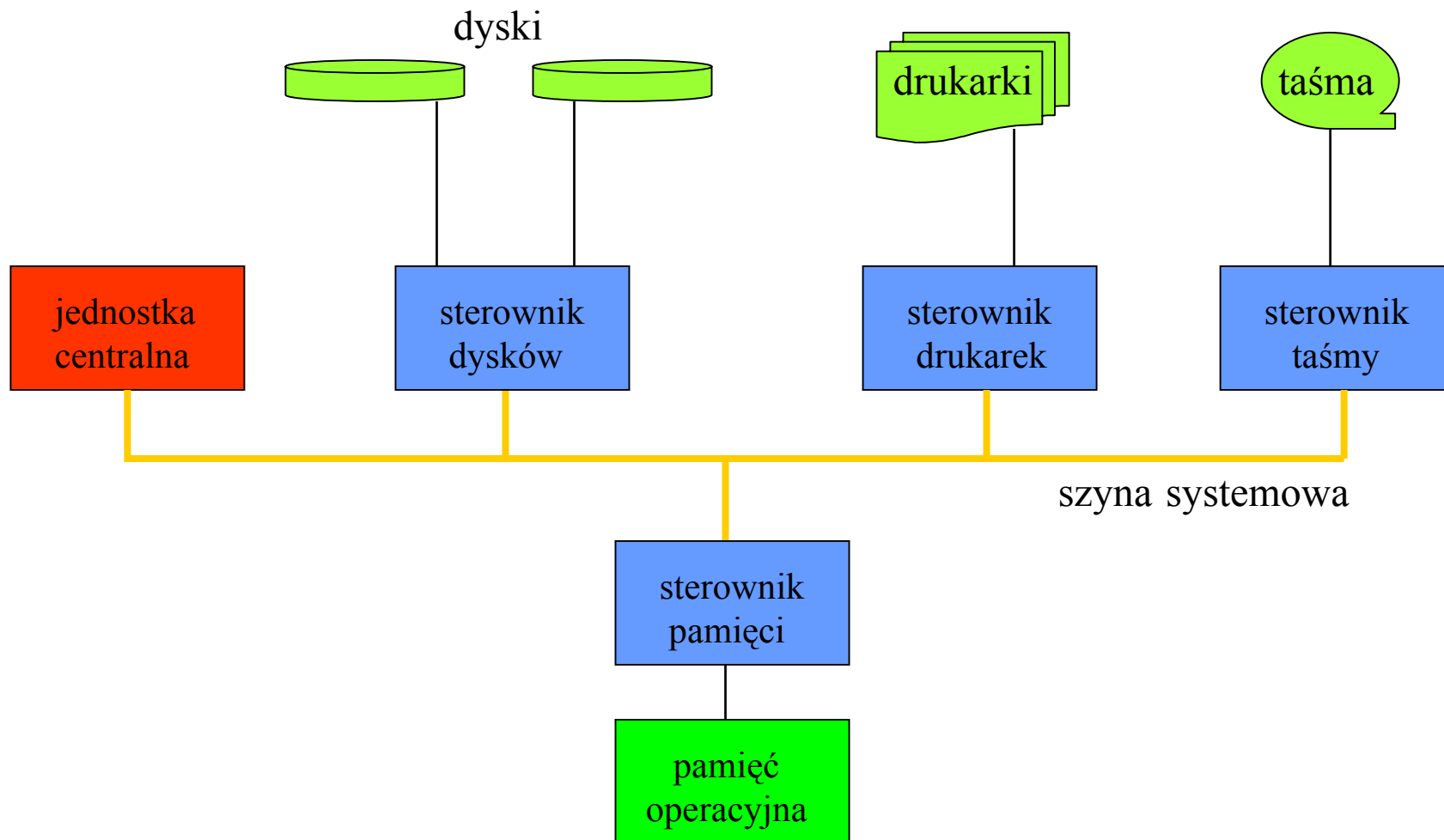
Motywacja

- ❖ *„Komputer stanowi podstawę techniki komputerowej. Bez niego większość dyscyplin tej techniki pozostawałaby dziedziną matematyki teoretycznej. Aby dzisiaj być profesjonalistą w jakiegokolwiek dyscyplinie techniki obliczeniowej, nie można komputera traktować jedynie jako czarnej skrzynki wykonującej programy w jakiś magiczny sposób. Wszyscy studium techniki obliczeniową powinni osiągnąć pewien stopień zrozumienia zespołów funkcjonalnych **systemu komputerowego**, ich właściwości, wydajności i współdziałania. Istnieją również pewne implikacje praktyczne. Studenci powinni rozumieć architekturę komputerów, żeby byli w stanie nadawać programom taką strukturę, aby te funkcjonowały skuteczniej w rzeczywistych komputerach. Wybierając system przeznaczony do użytku, powinni oni rozumieć kompromisy między właściwościami różnych zespołów, takimi jak szybkość zegara jednostki centralnej a wielkość pamięci.”*
- ❖ Computing Curricula 2001, Computer Science, IEEE & ACM

System komputerowy

- ❖ Sprzęt (ang. hardware) - zasoby o specyficznej architekturze oraz organizacji zarządzane przez system operacyjny
- ❖ System operacyjny - program, który nadzoruje i koordynuje dostęp programów do zasobów
- ❖ Programy użytkowe - realizują potrzeby użytkowników systemu komputerowego (kompilatory, bazy danych, gry,....)
- ❖ Użytkownicy - ludzie, maszyny, komputery..

Sprzęt komputerowy (c.d.)



Architektura komputera

- ❖ Architektura komputera odnosi się do atrybutów systemu widocznych dla programisty
 - ❖ atrybuty architektury: lista rozkazów, liczba bitów użytych do reprezentacji typów danych, mechanizmy we/we, metody adresowania pamięci
 - ❖ zagadnienie architektury: Czy w komputerze jest rozkaz mnożenia?

Organizacja komputera

- ❖ Organizacja komputera odnosi się do jednostek operacyjnych i ich połączeń, które stanowią specyfikację typu architektury
 - ❖ atrybuty organizacji (ukryte dla programisty): sygnały sterujące, interfejsy, technologia pamięci.
 - ❖ zagadnienie organizacji: Czy istnieje jednostka mnożąca (sprzętowa), czy mnożenie jest realizowane przez jednostkę sumującą?

Architektura a organizacja

- ❖ Rodzina komputerów Intel x86 ma tą samą podstawową architekturę
- ❖ Rodzina komputerów IBM System/370 ma tą samą podstawową architekturę
- ❖ Po co?
 - ❖ Kompatybilność oprogramowania (przynajmniej dla starszych wersji)
- ❖ Organizacja jest różna dla różnych modeli tej samej rodziny (architektury) x86 lub IBM/370

Struktura i funkcja

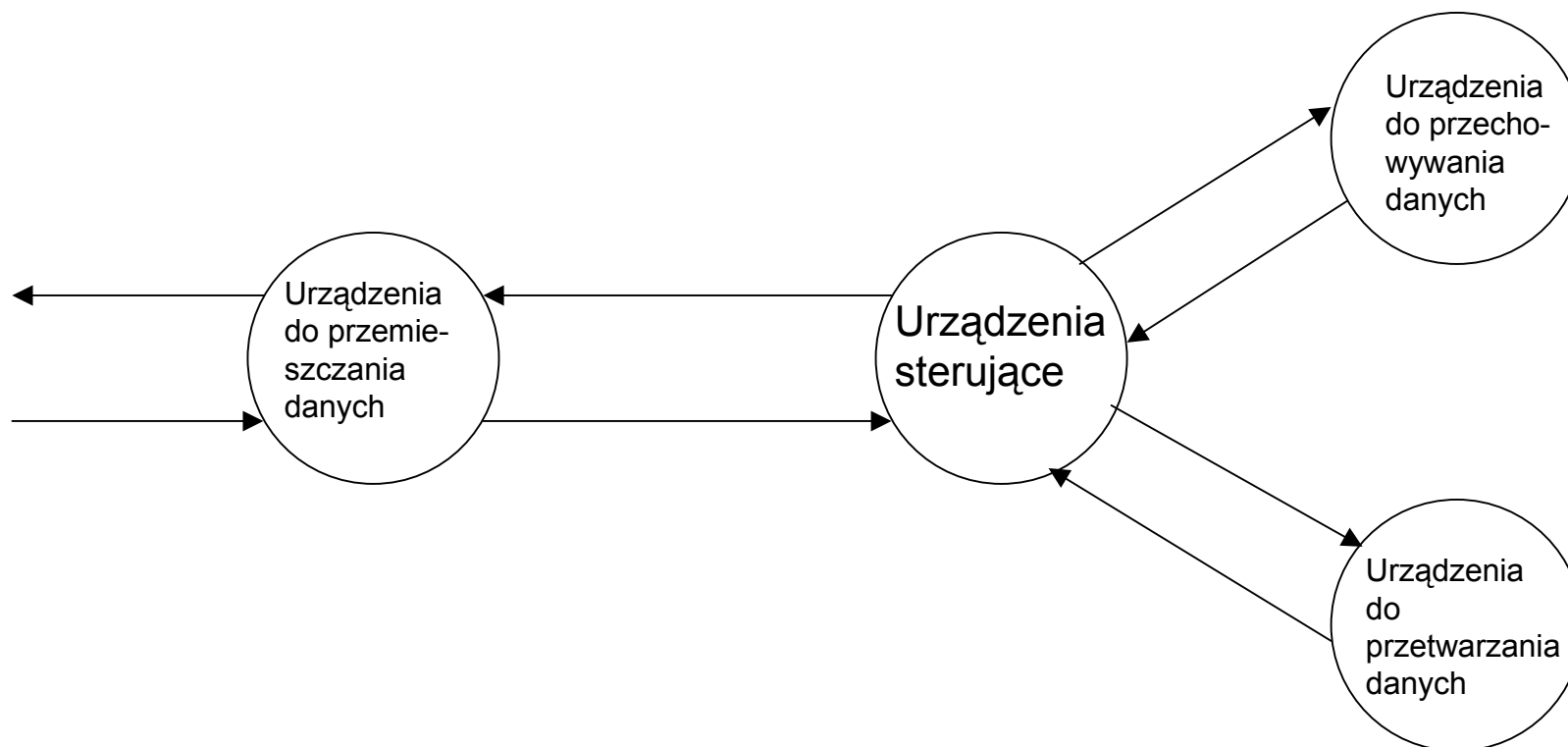
- ❖ Jak poznać tak złożony system jak komputerowy?
- ❖ System hierarchiczny jest układem wzajemnie powiązanych podsystemów, z których każdy ma także strukturę hierarchiczną, itd.. aż do osiągnięcia końcowego systemu elementarnego
 - ❖ podsystem opisujemy metodą od góry do dołu (ang. topdown)
 - ❖ np. CPU->JS
- ❖ Struktura to sposób wzajemnego powiązania składników danego (pod)systemu
- ❖ Funkcje określają działania poszczególnych składników jako części struktury

Funkcje

- ❖ Są tylko cztery funkcje ogólne komputera:
 - ❖ przetwarzanie danych - różne formy i zakres wymagań
 - ❖ przechowywanie danych
 - ❖ krótkotrwałe
 - ❖ długotrwałe
 - ❖ przenoszenie danych - proces wejścia/wyjścia między komputerem a urządzeniem peryferyjnym (zewnętrznym)
 - ❖ sterowanie - wszystkimi powyższymi funkcjami

Obraz funkcjonalny

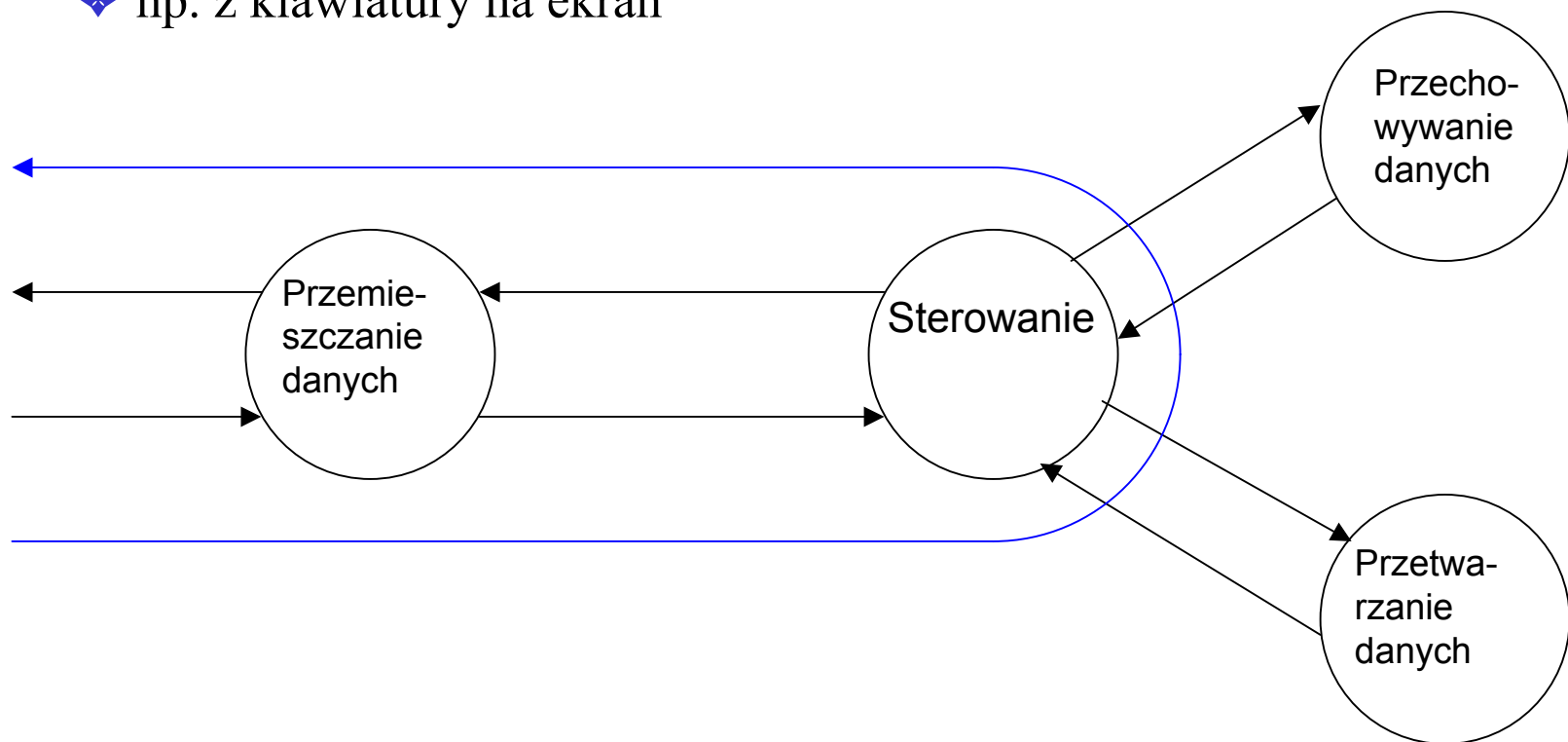
❖ Obraz funkcjonalny komputera



Operacje (1)

❖ Przenoszenie danych

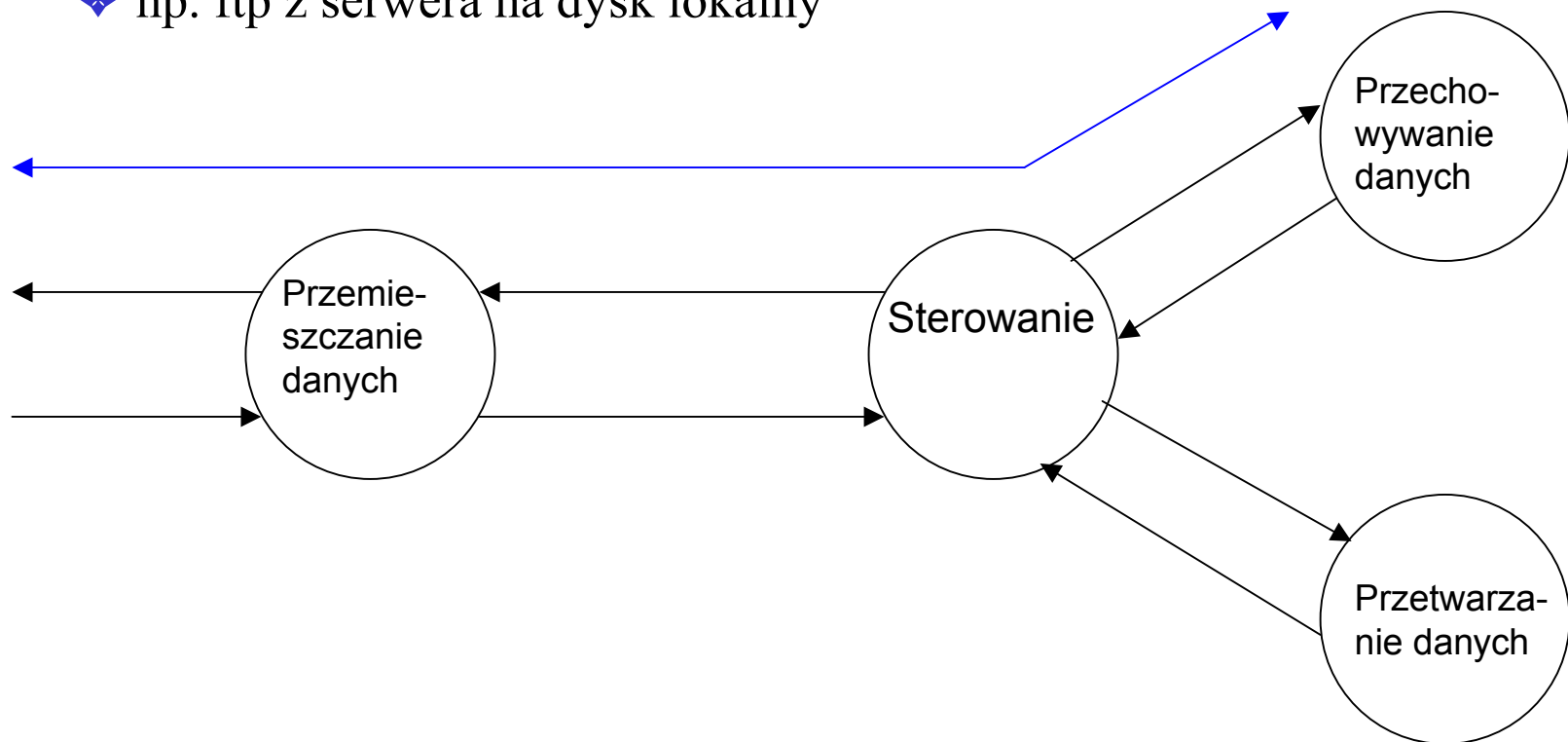
❖ np. z klawiatury na ekran



Operacje (2)

❖ Przechowywanie danych

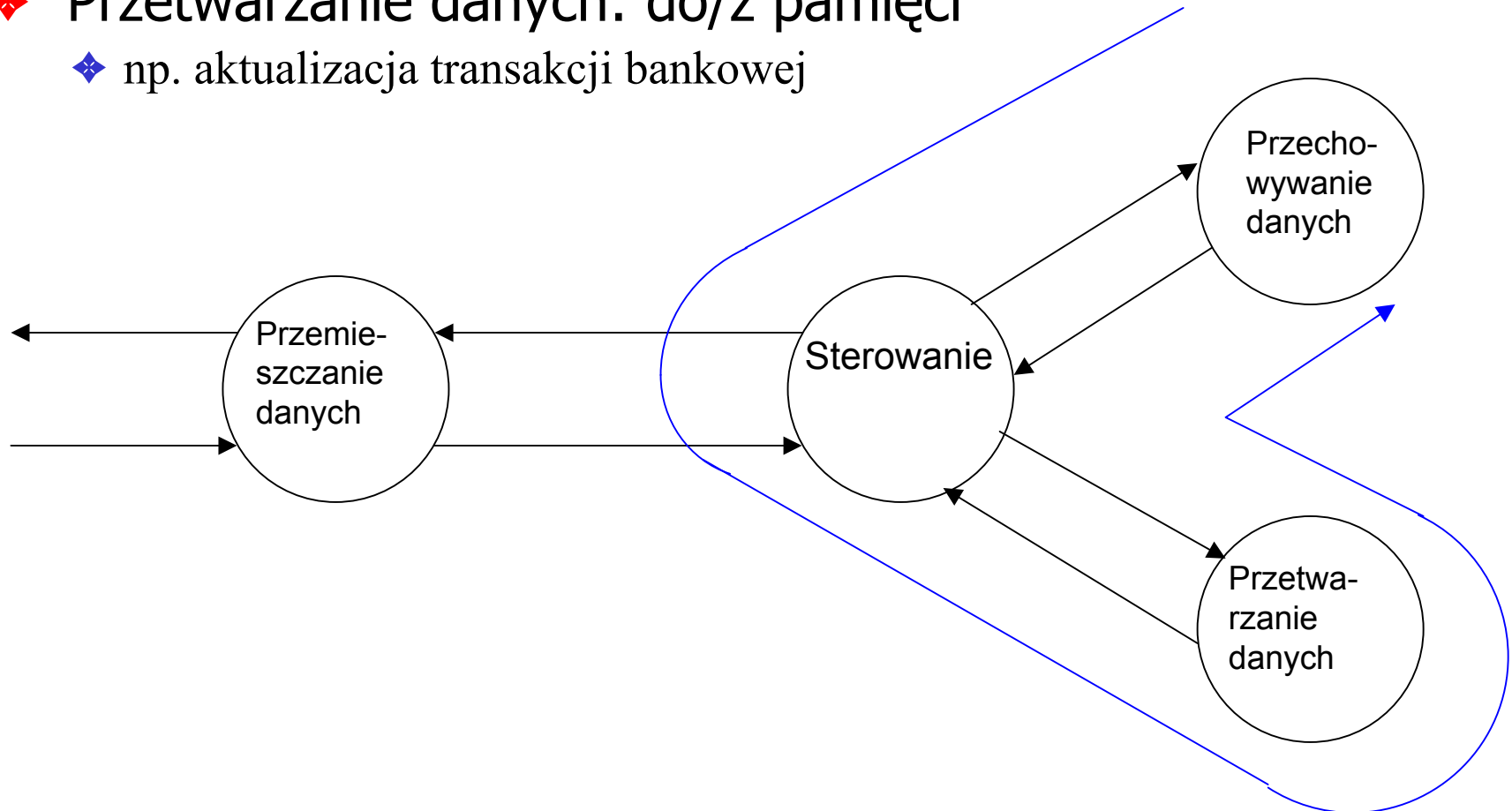
❖ np. ftp z serwera na dysk lokalny



Operacje (3)

❖ Przetwarzanie danych: do/z pamięci

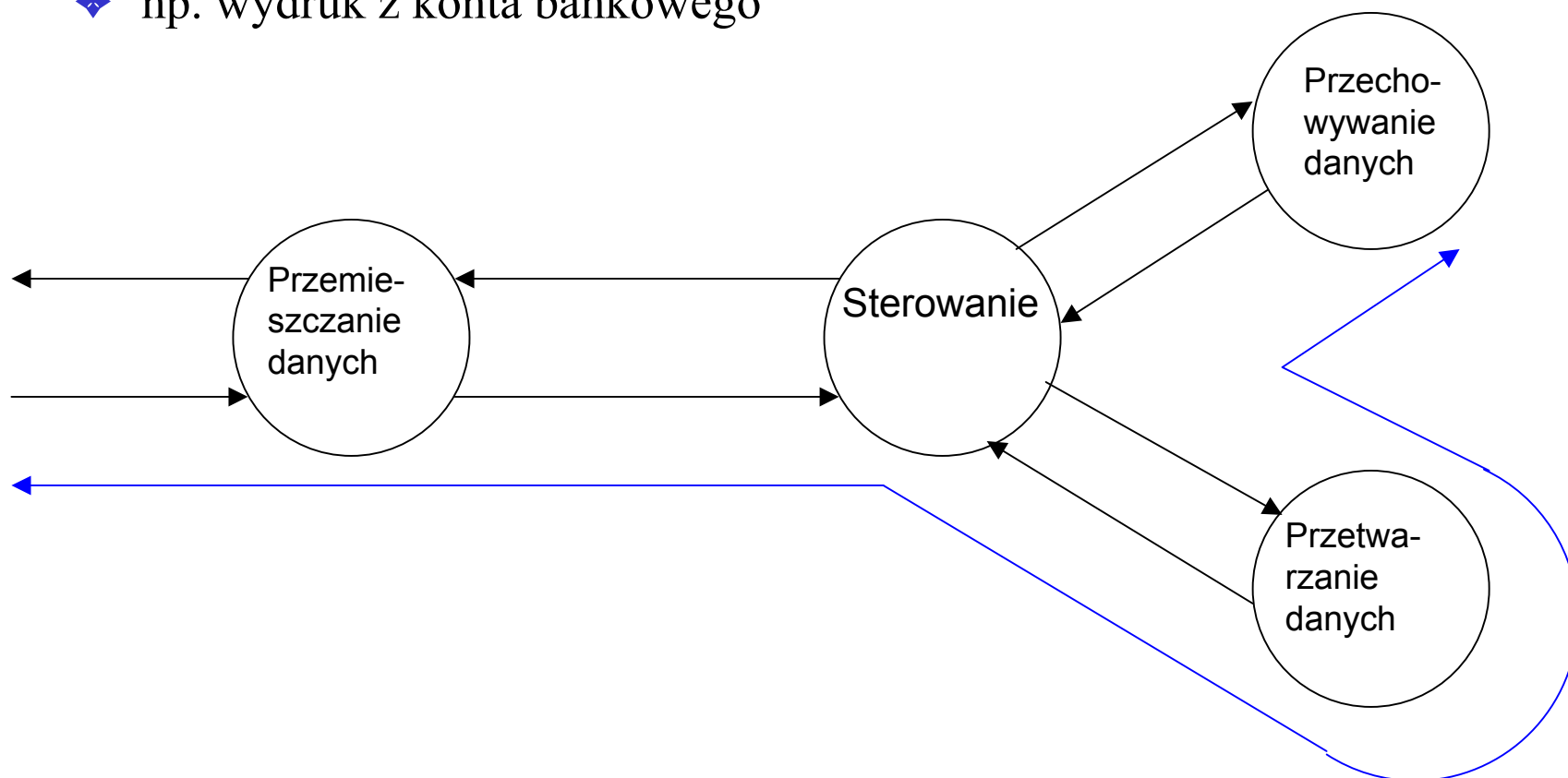
❖ np. aktualizacja transakcji bankowej



Operacje (4)

❖ Przetwarzanie danych: z pamięci na we/wy

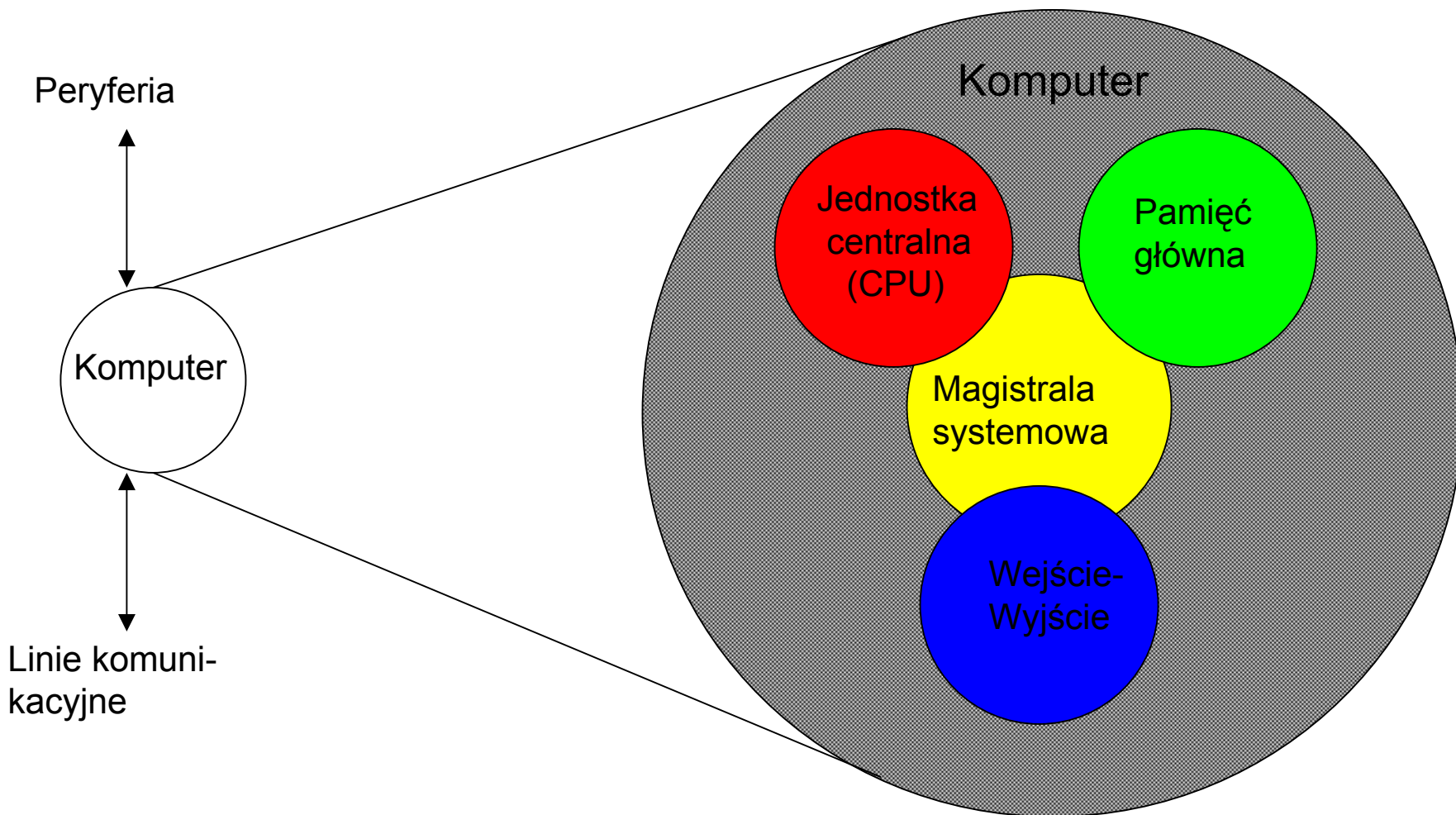
❖ np. wydruk z konta bankowego



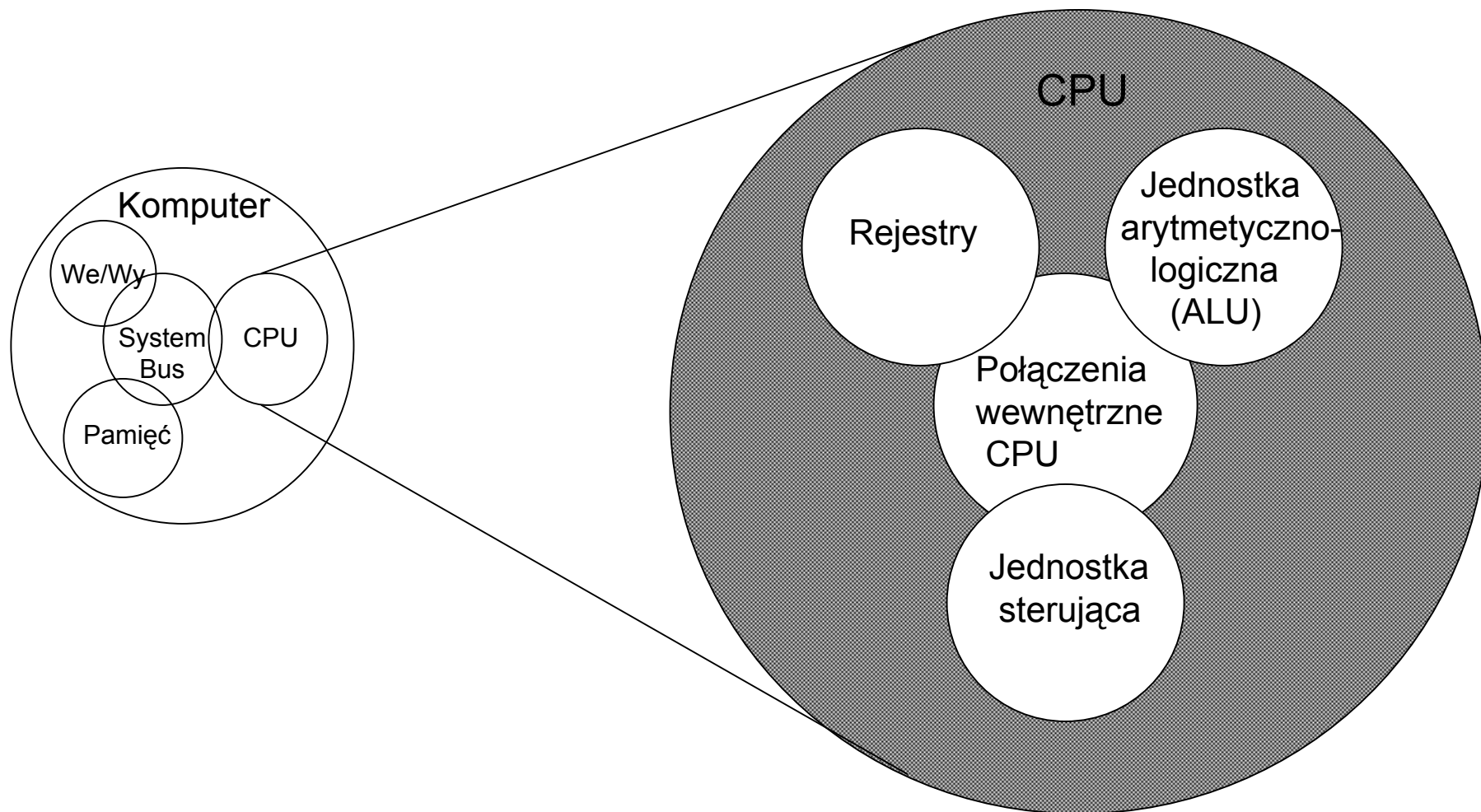
Struktura

- ❖ **Jednostka centralna (CPU, procesor)** - steruje działaniem komputera i realizuje funkcje przetwarzania danych
- ❖ **Pamięć główna** - przechowuje dane
- ❖ **Wejście/Wyjście** - przenosi dane między komputerem a światem zewnętrznym
- ❖ **Połączenia systemu** - mechanizmy zapewniające łączność między procesorem, pamięcią główną a wejściem/wyjściem

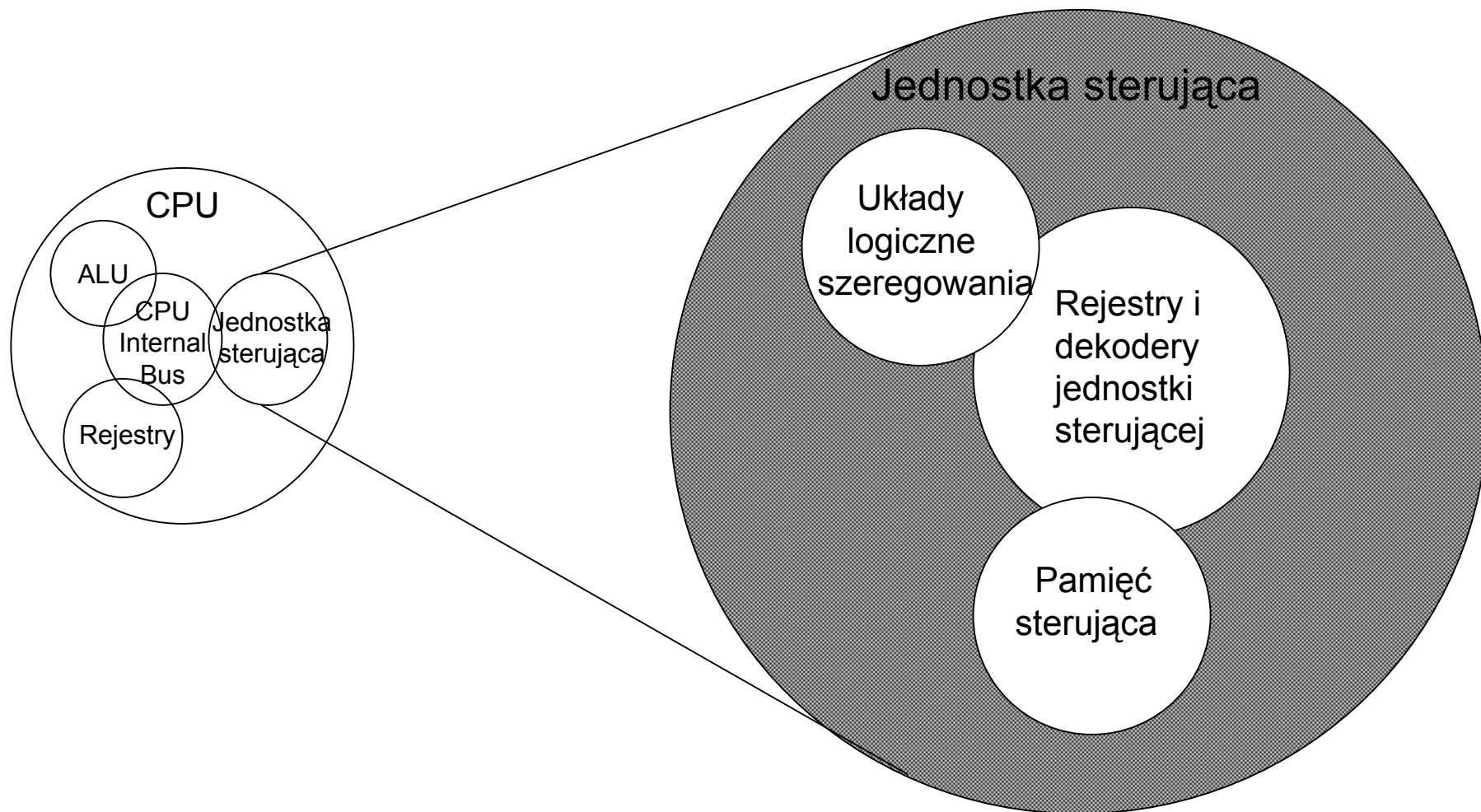
Struktura komputera



Struktura jednostki centralnej



Struktura jednostki sterującej



System dwójkowy

❖ 11001 binarnie (bin) to:

$$\begin{aligned} & 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 16 + 8 + 0 + 0 + 1 = 25 \end{aligned}$$

❖ Inny zapis:

$$11001_2 = 25_{10}$$

Algorytm rozwinięcia liczby w ułamek łańcuchowy

- ❖ $\lfloor u \rfloor$ - najmniejsza liczba całkowita mniejsza lub równa u
 - ❖ $\lfloor -1/2 \rfloor = -1, \lfloor \pi \rfloor = 3$
- ❖ liczba rzeczywista x_0 ma rozwinięcie $[a_0; a_1, a_2, a_3, \dots]$
 - ❖ $a_n = \lfloor x_n \rfloor; x_{n+1} = 1/(x_n - a_n); n=0,1,2,\dots;$
- ❖ Prz. $1281/243 = [5; 3, 1, 2, 7]$
 - ❖ $a_0 = \lfloor 1281/243 \rfloor = 5; a_1 = \lfloor 1/(1281/243 - 5) \rfloor = \lfloor 243/66 \rfloor = 3;$
 - ❖ $a_2 = \lfloor 1/(243/66 - 3) \rfloor = \lfloor 66/45 \rfloor = 1; a_3 = \lfloor 1/(66/45 - 1) \rfloor = \lfloor 45/21 \rfloor = 2$
 - ❖ $a_4 = \lfloor 1/(45/21 - 2) \rfloor = 7; x_5 \rightarrow \text{stop dzielenie przez } (7-7)=0!$
- ❖ Prz. $\sqrt{3} = [1; 1, 2, 1, 2, 1, 2, 1, 2, \dots]$
 - ❖ $a_0 = \lfloor \sqrt{3} \rfloor = 1; a_1 = \lfloor 1/(\sqrt{3} - 1) \rfloor = \lfloor \frac{1}{2}(\sqrt{3} + 1) \rfloor = 1;$
 - ❖ $a_2 = \lfloor 1/(\frac{1}{2}(\sqrt{3} + 1) - 1) \rfloor = \lfloor \sqrt{3} + 1 \rfloor = 2;$
 - ❖ $a_3 = \lfloor 1/((\sqrt{3} + 1) - 2) \rfloor = \lfloor \frac{1}{2}(\sqrt{3} + 1) \rfloor = 1; \text{pętla: } 1, 2, 1, 2, \dots$
- ❖ Prz. $\pi = [3; 7, 15, 1, 292, 1, 1, 1, 2, 1, 3, 1, 14, 2, 1, 1, \dots]$
- ❖ liczba niewymierna ma (jedyne) nieskończone rozwinięcie

Ewolucja komputerów (c.d.)

❖ Wilhelm Schickard (1592-1635)

- ❖ profesor astronomii i języków biblijnych w Tybindze
- ❖ czterodziałaniowa maszyna do liczenia (dla Keplera) (spłonęła w 1624)

❖ Blaise Pascal (1623-1662)

- ❖ maszyna dwudziałaniowa (1642), bardzo popularna (do dziś)
- ❖ przeniesienie dziesiątków za pomocą kółek zębatach ponumerowanych 0,1...,8,9

❖ Gottfried Wilhelm von Leibniz (1646-1716)

- ❖ projekt czterodziałaniowego arytmetru (1671) - skonstruowany w 1694
- ❖ wdrożenia arytmetru Leibniz'a
 - ❖ T. Odhner (1874), S. Baldwin (1875), C.-X. Thomas de Colmar (1882), P. Czebyszew (1882)
- ❖ arytmetyka binarna, początki rachunku logicznego (algebry Boole'a)
- ❖ początki analizy matematycznej (1672) i rachunku prawdopodobieństwa
 - ❖ J.Bernoulli, L.Euler, J.Stirling, I.Newton, A. de Moivre, B.Taylor, P. Czebyszew

Materializowanie logiki i arytmetyki

❖ Francja II połowa XVIII wieku

- ❖ maszyny tkackie sterowane taśmą (Bouchon, 1725) i kartą papierową (Falcon, 1728) i Jacquard (1801): 24 000 kart z 1050 możliwościami perforacji
- ❖ Wykorzystanie idei: Hertz (1888) - fale -> Tesla (1893); Marconi (1896) - radio

❖ Charles Babbage (1791-1871) – synteza mechaniki, logiki i arytmetyki

- ❖ difference engine (tablice) – maszyna różnicowa (arytmometr), 1822-1833
- ❖ analytical engine (1834-1871) – maszyna analityczna
 - ❖ arytmometr (młyn, ang. mill), pamięć (magazyn, ang. store), urządzenie do wczytywania kart papierowych, układ uruchamiania maszyny, urządzenie do drukowania wyników
- ❖ metoda programowania za pomocą kart perforowanych
- ❖ Ada Augusta hr. Lovelace (1815-1852) (córka Byrona, uczennica de Morgana)
 - pierwsza programistka:
 - ❖ „...*Maszyna analityczna będzie tkąła wzory algebraiczne...*”
- ❖ brak odpowiedniej technologii dla realizacji świetnych pomysłów

Charles Babbage (1791-1871)



Ewolucja komputerów (c.d.)

❖ Ameryka Północna

❖ Herman Hollerith (1860-1929)

- ❖ licząco-sortujące urządzenie mechanograficzne (tabulator) (1884)
 - ❖ dane czytane z kart perforowanych – przepływ prądu przez otwór w karcie powodował przesunięcie wskazówki licznika o jedną działkę
- ❖ spis ludności w 1890 r. w USA - zaoszczędzenie 5M \$
- ❖ założył firmę Tabulating Machine Co. (1896)
 - ❖ Computer Tabulating Recording Co. (1911)
 - ❖ CTR w 1924 przyjęła nazwę IBM

❖ Thomas Watson (1874-1956) – pierwszy prezes IBM

- ❖ tabulator z drukarką (1920)
- ❖ 80 kolumnowe karty (IBM standard, 1928)
- ❖ tabulator alfabetyczny i maszyna mnożąca (1932)
- ❖ elektroniczna maszyna kalkulacyjna (1946)

Materializowanie logiki i arytmetyki

❖ Technologia elektromechaniczna

- ❖ Konrad Zuse (1938-1941) – komputery Z-1, Z-3
 - ❖ wyłączniki elektromechaniczne z przekaźnikiem; pamięć: 64x22bity, arytmetyka zmiennopozycyjna
 - ❖ http://irb.cs.tu-berlin.de/%7Ezuse/Konrad_Zuse/en/Rechner_Z1.html
- ❖ George Stibitz (1904-1995) – maszyna zbudowana w Bell Laboratories (1940)
 - ❖ Bell Relay Computer (1946) – 10 ton na 105 m², 9000 przekaźników; mnożenie 2 liczb 7-cyfrowych w 1s
- ❖ Howard Aiken (1900-1973) + IBM (1944) – maszyna Mark I, Harvard University, realizacja maszyny analitycznej
 - ❖ zarekwirowana przez US Navy dla obliczeń balistycznych
 - ❖ urządzenie mechaniczne z elektromagnetycznymi przekaźnikami
 - ❖ wprowadzenie mechanizmu umożliwiającego przejścia warunkowe
 - ❖ 5 ton, 750 000 części (m.in. 1000 łożysk kulkowych, 850 km przewodów elektrycznych)
 - ❖ zegar o okresie 0,015 s: dodawanie - 0,3s; mnożenie - 0,6s; obliczenie sinusa - 60s (bardziej wiarygodna niż ENIAC ale wolniejsza)

Materializowanie logiki i arytmetyki

❖ Technologia lamp próżniowych (I generacja)

- ❖ wykorzystanie "zjawiska termojonowego" zachodzącego w lampie próżniowej T. A. Edisona (1847-1931)
 - ❖ dioda (1904) - J. A. Fleming (1849-1945), trioda (1907) – L. De Forest (1873-1961)
 - ❖ flip-flop (1919) – przerzutnik dwójkowy, W.H. Eccles i F.W. Jordan
 - ❖ J. Atanasoff (1903-1995), C. Berry (1918-1963) – kalkulator cyfrowy (do rozwiązywania równań liniowych) ABC, zbudowany pomiędzy 1937 i 1941 w Iowa State College (projekt znany twórcom ENIACa)
 - ❖ <http://www.scl.ameslab.gov/ABC/>
- ❖ Bletchley Park, Anglia (1943) – COLOSSUS
 - ❖ pierwszy komputer elektroniczny (8.12.1943)
 - ❖ odszyfrował ok. 80 000 depesz zakodowanych przez Lorenz SZ40
 - ❖ złamanie szyfru Enigmy: Marian Rejewski, 1932
 - ❖ <http://www.picotech.com/applications/colossus.html>

ENIAC - komputer I generacji

❖ Electronic Numerical Integrator And Computer

- ❖ pierwsza elektroniczna maszyna analityczna
- ❖ John Presper Eckert (1919-1995) i John William Mauchly (1907-1980)
- ❖ Początek projektu - 1943, wdrożenie w 1946
 - ❖ John von Neumann (1903-1957) zainteresował się projektem (1944) w związku z budową bomby atomowej
- ❖ Moore School of Electrical Engineering, University of Pennsylvania
- ❖ Tablice strzelnicze - tory lotu i zasięgu pocisków

❖ ENIAC obliczył liczbę π z dokładnością do 2037 znaków w 70 godzin (1949)

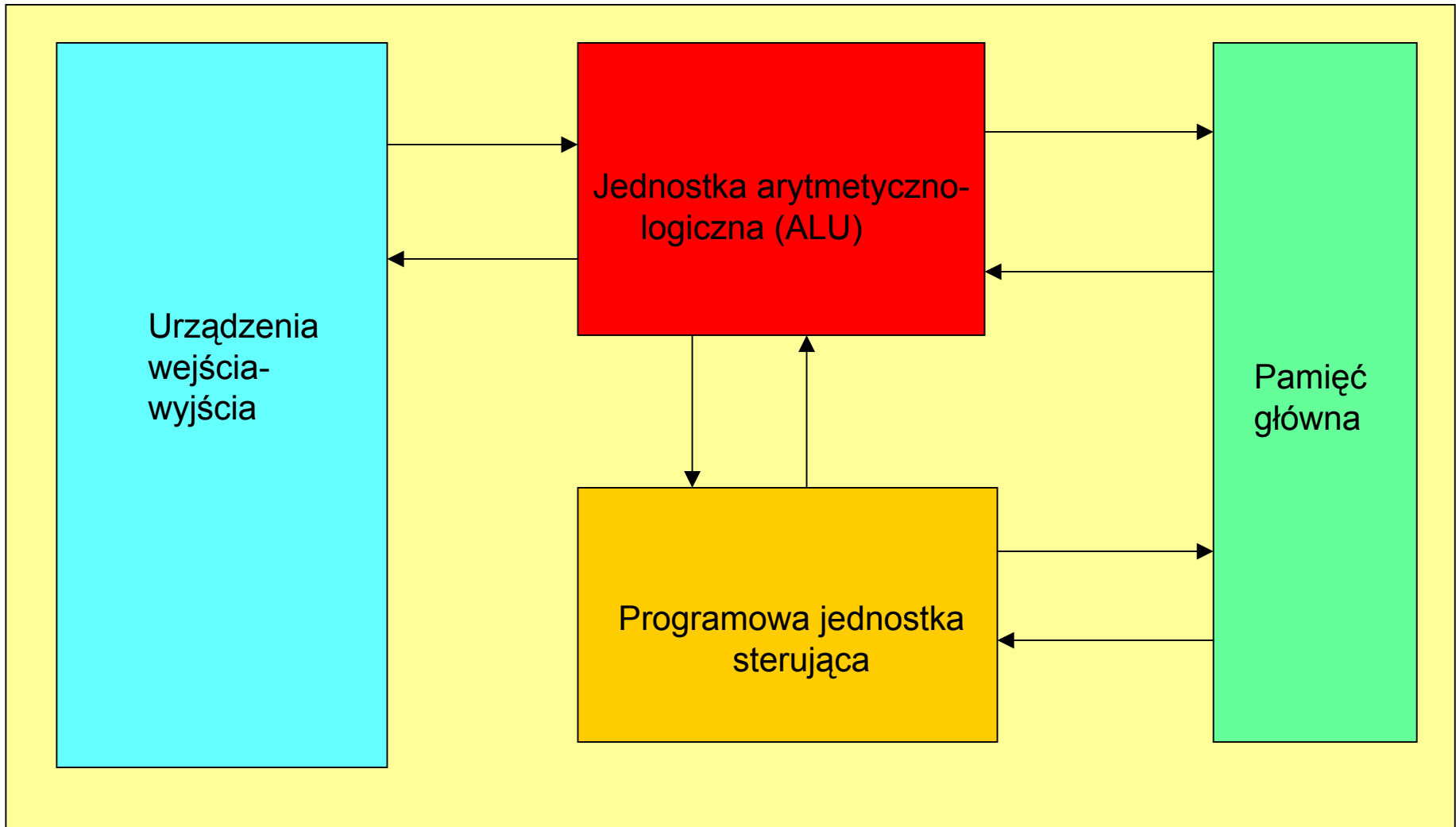
- ❖ Anglik W. Shanks w latach 1855-1874 obliczył liczbę π z dokładnością do 707 znaków (popełnił jednak błąd na 528 miejscu (Ferguson, 1945))
- ❖ Maple: `evalf(Pi,2037);`

❖ ENIAC działał do roku 1955 w Ballistic Research Center, Aberdeen

Idea komputera

- ❖ Schickard, 1623; Babbage, 1833; Stibitz, 1940; Zuse, 1941, Atanasoff-Eckert, 1942; Aiken, 1944;
- ❖ Synteza von Neumanna, 1945: realizacja idei Leibniza, 1671
 - ❖ pamięć główna przechowuje dane i programy
 - ❖ jednostka arytmetyczno-logiczna (ALU) działa na danych binarnych
 - ❖ jednostka sterująca interpretuje oraz wykonuje rozkazy z pamięci
 - ❖ urządzeniami we/wy steruje jednostka sterująca
- ❖ koncepcja programu wprowadzanego
- ❖ Pierwsze realizacje idei von Neumanna
 - ❖ EDSAC, M. V. Wilkes (1913-), Cambridge, Anglia, uruchomiony 6.V.1949
 - ❖ komputer IAS, Princeton Institute for Advanced Studies
 - ❖ początek projektu 1946, ukończony 10.06.1952
 - ❖ 2300 lamp próżniowych (ang. vacuum tubes)
- ❖ **Komputery współczesne są jedynie ulepszeniem IAS-a**
 - ❖ komputery działające zgodnie z zasadami fizyki klasycznej

Struktura maszyny von Neumanna



IAS - architektura

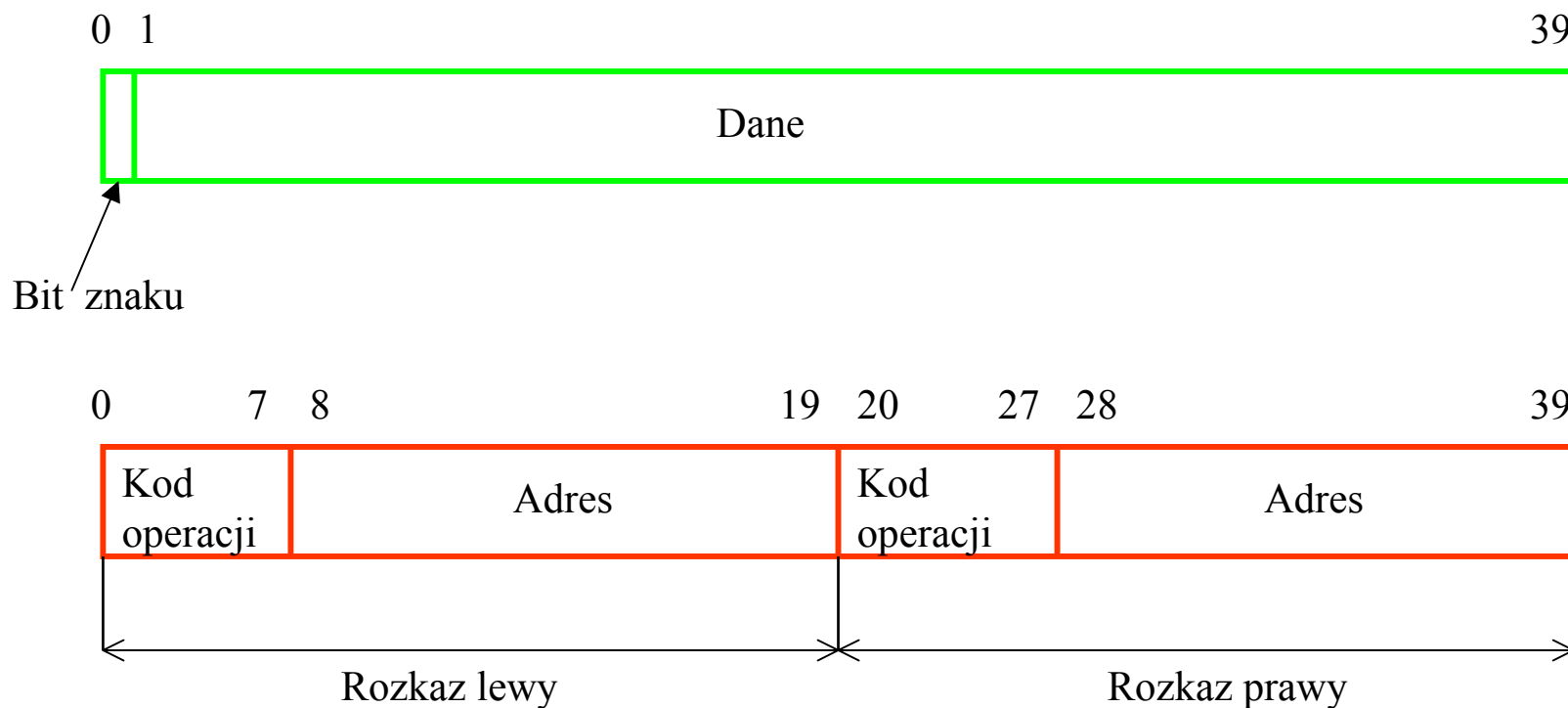
❖ Pamięć: 1K słów 40 bitowych

- ❖ Reprezentacja binarna w kodzie dopełnieniowym (uzupełnienia do dwóch) (ang. two's complement, U2)
 - ❖ negujemy bity liczby bez znaku i dodajemy binarnie 1
- ❖ 2 x 20 bitowe rozkazy

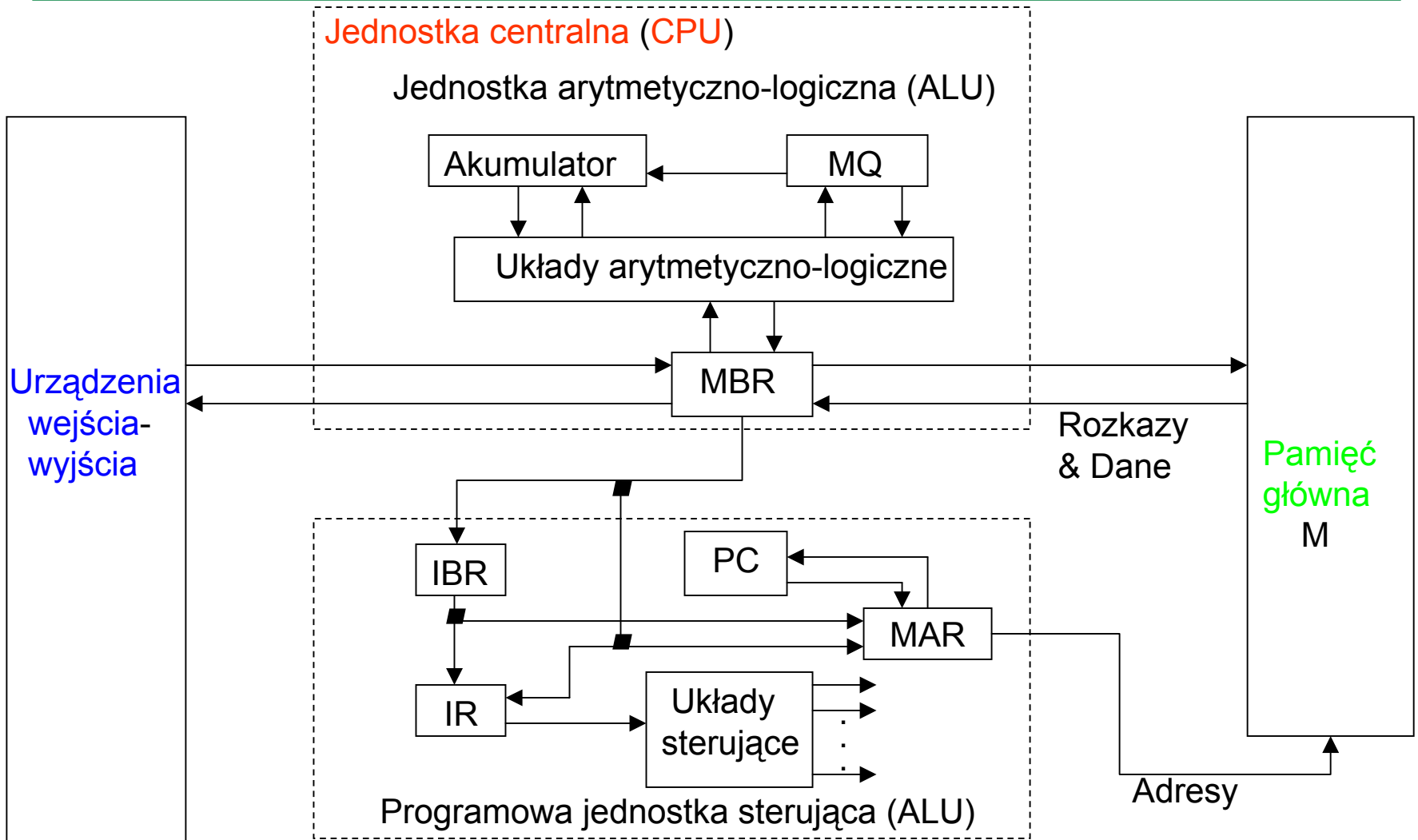
❖ Rejestry (pamiętane w CPU)

- ❖ Rejestr buforowy pamięci - MBR: Master Buffer Register
- ❖ Rejestr adresowy pamięci - MAR: Memory Address Reg.
- ❖ Rejestr rozkazów - IR: Instruction Register
- ❖ Rejestr buforowy rozkazów - IBR: Instruction Buffer Reg.
- ❖ Licznik programu - PC : Program Counter
- ❖ Akumulator - AC: Acumulator
- ❖ Rejestr mnożenia-dzielenia -MQ: Multiper-Quotier

IAS - format danych i rozkazów



Struktura komputera IAS



Rejestry IAS

- ❖ MBR - zawiera słowo, które ma być przechowywane w pamięci, lub też jest wykorzystywany do pobierania słów z pamięci
- ❖ MAR - określa adres słowa w pamięci, które ma być zapisane lub odczytane do/z MBR
- ❖ IR - zawiera 8-bitowy kod wykonywanego rozkazu
- ❖ IBR - rejestr roboczy do przechowywania prawej połowy słowa rozkazu

Rejestry IAS (c.d.)

- ❖ PC - zawiera adres następnej pary rozkazów, która ma być pobrana z pamięci
- ❖ AC i MQ - rejestry robocze do czasowego przechowywania argumentów oraz wyników operacji prowadzonych przez ALU
 - ❖ wynik mnożenia liczb 40-bitowych jest 80-bitowy; 40 starszych bitów jest pamiętanych w AC, 40 młodszych bitów jest pamiętanych w MQ
- ❖ Oznaczenie $M(X, A:B)$ - zawartość komórki pamięci o adresie X od bitu A do bitu B; $M(X) = M(X, 0:39)$
 - ❖ $M(5, 0:7)$ - kod instrukcji w szóstym słowie
- ❖ Lista 21 rozkazów

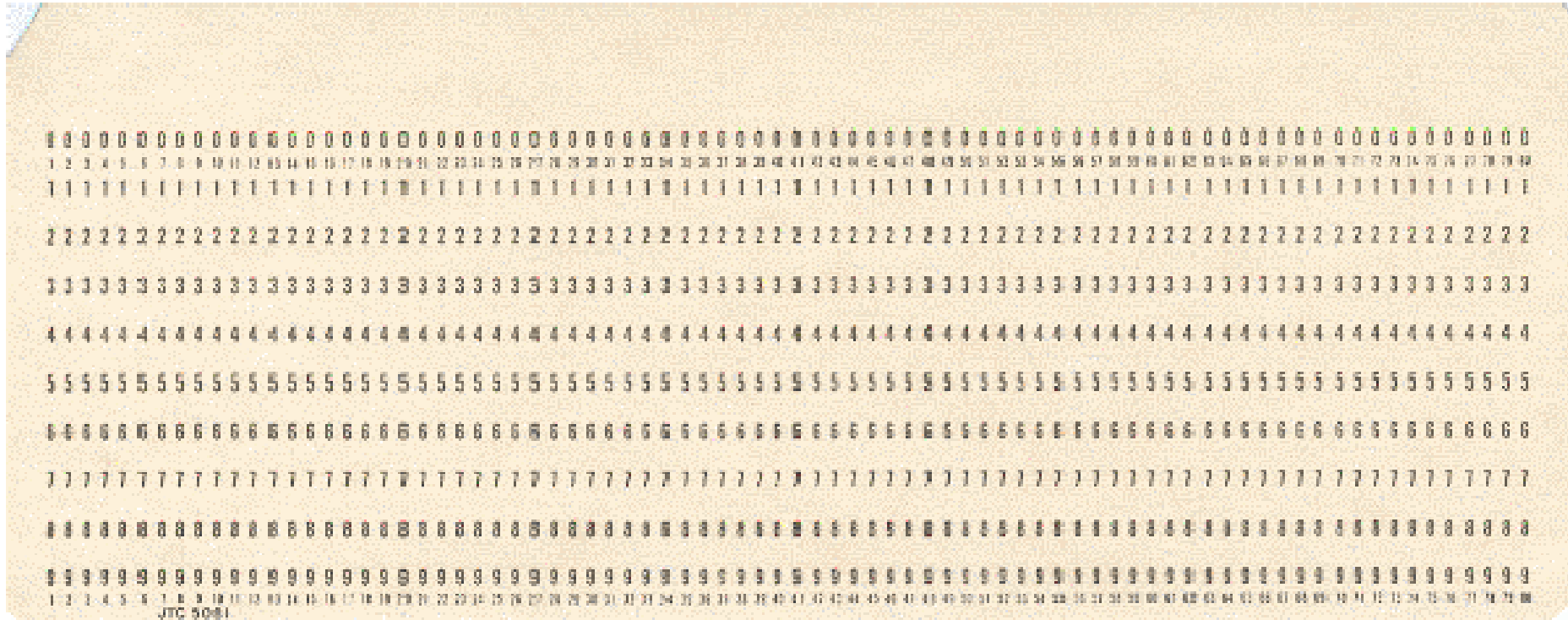
Komputery komercyjne

- ❖ 1947 - Eckert - Mauchly Computer Corporation
- ❖ UNIVAC I (Universal Automatic Computer)
 - ❖ pierwszy dostępny w handlu komputer (1951)
- ❖ US Bureau of Census : spis ludności w 1950
- ❖ EMCC przejęta przez Sperry-Rand Corporation
- ❖ Koniec lat 50-tych - UNIVAC II
 - ❖ Większa wydajność
 - ❖ Większa pamięć

IBM

- ❖ Urządzenia do przetwarzania danych wprowadzanych z kart perforowanych
- ❖ Seria IBM/700 - CPU: lampy próżniowe
 - ❖ IBM 701 (1953), pamięć elektrostatyczna 2KB; pierwszy komputer do zastosowań naukowych
 - ❖ wykonany na potrzeby wojny w Korei
 - ❖ IBM 704 (1955), pamięć rdzeniowa 32KB
 - ❖ IBM 709 (1958), nakładanie wejścia/wyjścia (ang. I/O)

Karta perforowana (IBM)



Komputery II generacji

- ❖ Komputery tranzystorowe
- ❖ NCR & RCA produkują pierwsze małe komputery II generacji
- ❖ IBM/7000
 - ❖ IBM 7094: separacja CPU i I/O: multiplexor, data channels
 - ❖ system operacyjny IBSYS
- ❖ DEC - 1957
 - ❖ Wyprodukowanie PDP-1
- ❖ Liczba tranzystorów w komputerach urosła do setek tysięcy
- ❖ Pierwszy superkomputer CDC 6600 – Seymour Cray
- ❖ 12.09.1958 - wynaleziono układ scalony (IC)
 - ❖ Jack Kilby (1903-2005), Texas Instruments
 - ❖ nagroda Nobla z fizyki w 2000r.

Komputery III generacji

- ❖ Układy scalone (Jack Kilby, 1958)

 - ❖ http://nobelprize.org/educational_games/physics/integrated_circuit/history/

- ❖ Mikroelektronika, dosłownie "mała elektronika"

- ❖ Komputer składa się z bramek, komórek pamięci oraz połączeń między nimi

 - ❖ bramka (ang. gate) implementuje funkcję boolowską, np. AND

 - ❖ komórka (ang. cell) pamięta bit danych: 0 lub 1

- ❖ Można to zrealizować w półprzewodniku

 - ❖ np. krzemowa płytką (ang. silicon wafer)

 - ❖ EPROM (ponad 1000 tranzystorów) ->

- ❖ Setki tranzystorów, rezystorów i przewodników

 - ❖ na jednej płytce krzemowej



Skale integracji

- ❖ Mała skala integracji - od 1965
 - ❖ do 100 elementów w mikroukładzie (ang. chip)
- ❖ Średnia skala integracji - do 1971
 - ❖ 100-3,000 elementów w mikroukładzie
- ❖ Wielka skala integracji - 1971-1977
 - ❖ 3,000 - 100,000 urządzeń w mikroukładzie
- ❖ Bardzo wielka skala integracji - 1978 do dziś
 - ❖ 100,000 - 100,000,000 urządzeń w mikroukładzie
- ❖ Ultrawielka skala integracji
 - ❖ ponad 100,000,000 urządzeń w chipie

Prawo Moore'a

- ❖ Wzrastające upakowanie w mikroukładzie
- ❖ Gordon Moore - współtwórca firmy Intel
- ❖ Liczba tranzystorów na chipie podwaja się co roku (1965)
- ❖ Od lat 70-tych rozwój ten uległ spowolnieniu
 - ❖ liczba tranzystorów na chipie podwaja się co 18 miesięcy (1970)
 - ❖ 30 tranzystorów (1965); ok. 200M tranzystorów (2005)
- ❖ Konsekwencje prawa Moore'a:
 - ❖ koszt chipa pozostaje prawie niezmienny
 - ❖ większe upakowanie oznacza skrócenie elektrycznych obwodów, tzn. większą wydajność
 - ❖ mały rozmiar = większa swoboda
 - ❖ redukcja zasilania oraz mniejsze chłodzenie
 - ❖ mniejsza ilość połączeń = większa niezawodność
- ❖ <http://www.intel.com/technology/mooreslaw/index.htm>

Komputery III generacji – IBM/360

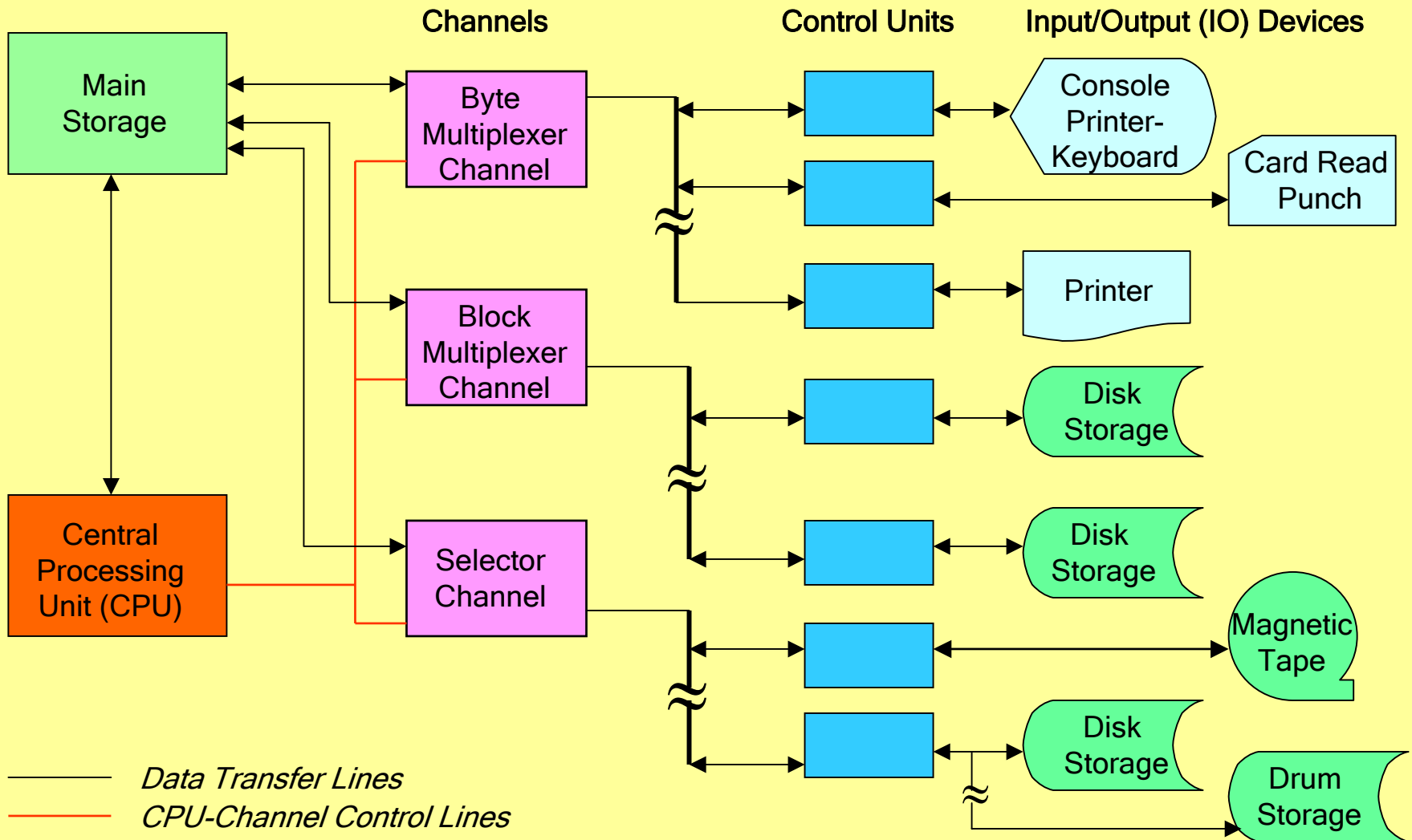
- ❖ 1964 - zastąpienie (niekompatybilne) modeli 7000
 - ❖ to samo może mieć miejsce (kiedyś) z architekturą Intela
- ❖ Pierwsza zaplanowana "rodzina" komputerów
 - ❖ podobna lub identyczna lista rozkazów
 - ❖ podobny lub identyczny system operacyjny
 - ❖ rosnąca szybkość
 - ❖ rosnąca liczba urządzeń we/wy (więcej terminali)
 - ❖ rosnący rozmiar pamięci
 - ❖ rosnąca cena
 - ❖ koniec produkcji w 1977 roku
- ❖ 1970 - System IBM/370 (MS->W95,W98,W2K)

System IBM/360

❖ Organizacja IBM/360

- ❖ Model 30 - pamięć: 64kB, transfer I/O: 250kBps
- ❖ Model 40 - pamięć: 256kB, transfer I/O: 400kBps
- ❖ Model 50 - pamięć: 256kB, transfer I/O: 800kBps
- ❖ Model 75- pamięć: 512kB, transfer I/O: 1250kBps
- ❖ CENTRALNE STEROWANIE
 - ❖ http://www.beagle-ears.com/lars/engineer/comphist/ibm_nos.htm
- ❖ 16x32b rejestrów, 4x64b rejestrów zmiennoprzecinkowych
- ❖ kodowanie znaków 8-bitowe: EBCDIC
- ❖ system operacyjny - MFT, MVT, MVT/TSO
- ❖ dyski twarde - 2311 (7.25MB), 2314 (29.2 MB)

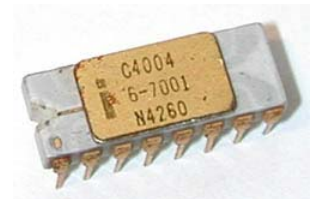
System 370 - organizacja



Komputery III generacji - Intel

❖ Procesory x86

- ❖ 1971 - 4004 - pierwszy mikroprocesor: 2.3K tranzystorów, 46 instrukcji



- ❖ wszystkie elementy procesora w jednym chipie
- ❖ dodawanie liczb 4-bitowych, mnożenie przez dodawanie
- ❖ zegar: 108kHz, szyna: 4b, pamięć: 640B

- ❖ 1972 - 8008, 3.5K tranzystorów

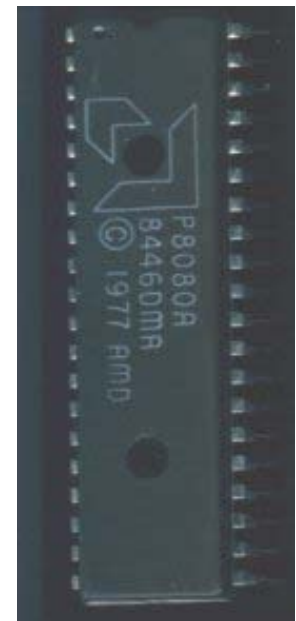
- ❖ zegar: 108kHz, szyna: 8b, pamięć: 16kB
- ❖ oba zaprojektowane do specyficznych zastosowań

- ❖ 1974 - 8080, 6K tranzystorów

- ❖ pierwszy mikroprocesor ogólnego przeznaczenia
- ❖ zegar: 2MHz, szyna: 8b, pamięć: 64kB

- ❖ 1978 - 8086, 29K tranzystorów

- ❖ zegar: 10MHz, szyna: 16b, pamięć: 1MB



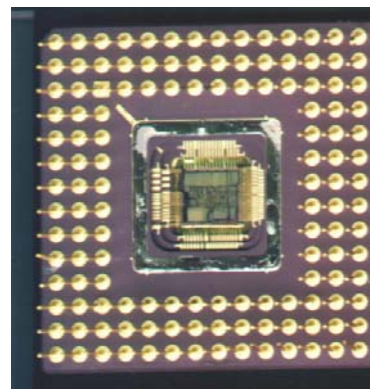
Komputery IV generacji - Intel

❖ 1982 - 80286, 134K tranzystorów

- ❖ zegar: 12MHz, szyna: 16b, pamięć: 1GB, wirtualna - 1GB

❖ 1985 - 386DX, 275K tranzystorów

- ❖ zegar: 16-33MHz, szyna: 32b, pamięć: 4GB, wirtualna - 64TB
- ❖ pierwszy procesor zdolny wykonać kod systemu Unix (Linux)
- ❖ IA32 - Intel Architecture 32-bit



❖ 1989 - 486DX, 1.9M tranzystorów

- ❖ zegar: 25-50MHz, szyna: 32b, pamięć: 4GB, wirtualna - 64TB

Komputery IV generacji – Intel (c.d.)

❖ 1993 - Pentium, 3.1M tranzystorów

- ❖ zegar: 60-166MHz, szyna: 32b, pamięć: 4GB, wirtualna - 64TB
- ❖ Pentium (gr. penta) - U.S. Trademark office nie rejestruje liczb
- ❖ <http://developer.intel.com/design/PentiumII/manuals/243191.htm>

❖ 1997 - Pentium II, 7M tranzystorów

- ❖ zegar: 300-450MHz, szyna: 64b, pamięć: 4GB, wirtualna - 64TB

❖ 1998-2007

- ❖ Pentium III, P6 (1999), 600-1000MHz, 8.2M tranzystorów
- ❖ Pentium 4, P7 (2000) - 1.5-3.0 GHz, 42M tranzystorów
- ❖ Itanium (2002) – 800MHz, **Intel Architecture 64-bit**
- ❖ Pentium 4 „E” (2004) – 3.8 GHz, 125M tranzystorów (skala ultra wielka)
- ❖ Penryn (2007) – 410M tranzystorów (dwurdzeniowy)

❖ <http://www.intel.com/technology/silicon/sp/index.htm>

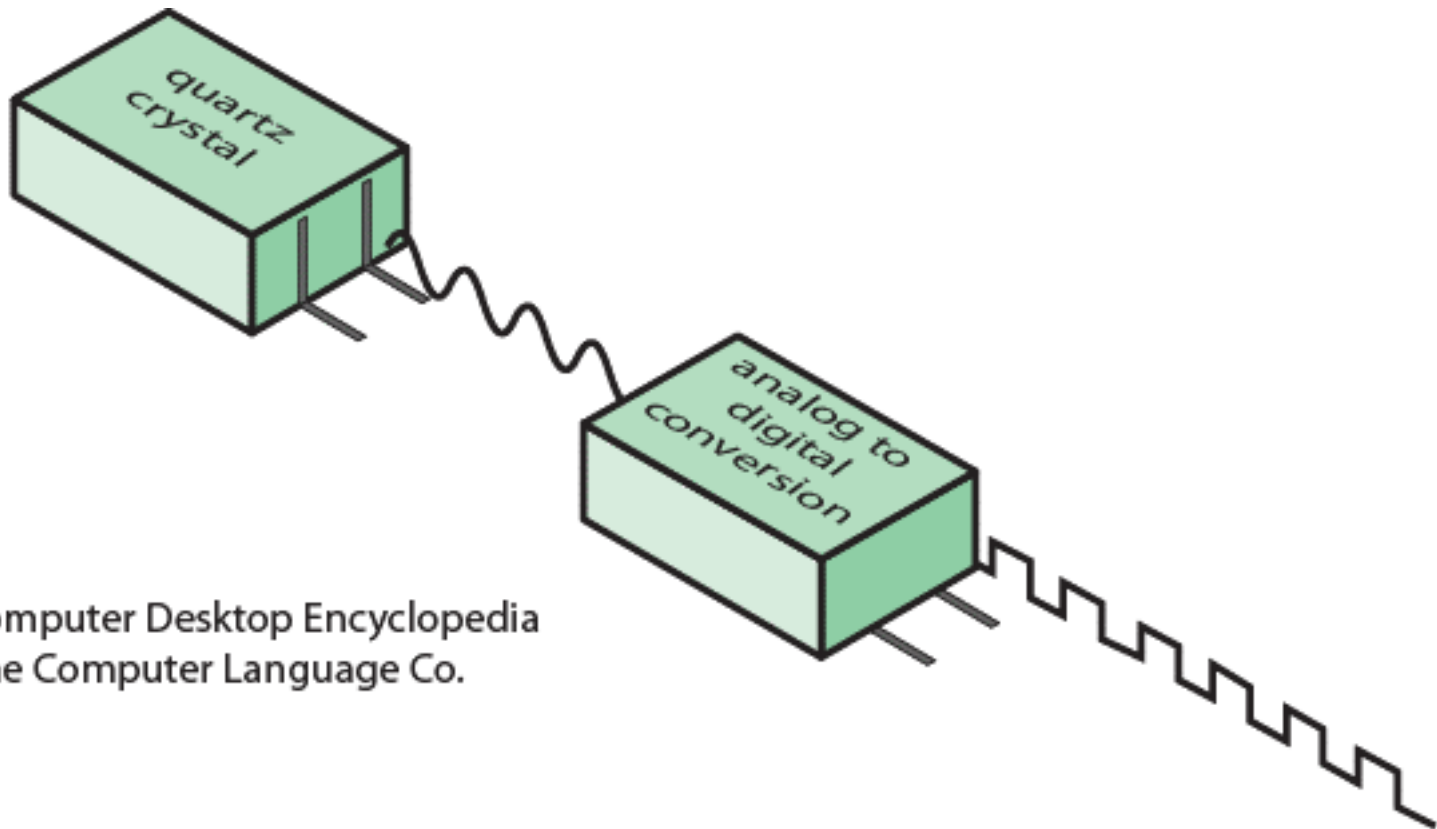
❖ *“Nic nie jest trwałe, oprócz zmiany” (Heraklit)*

❖ Zad. Ile tranzystorów będzie w procesorze x86 w 2012r?

Zegar

- ❖ Zegar - układ wysyłający regularne impulsy o stałej szerokości (τ) i częstotliwości (f)
 - ❖ http://en.wikipedia.org/wiki/Computer_clock
 - ❖ umożliwia kontrolę relacji czasowych w CPU
 - ❖ pobieranie, dekodowanie, wykonanie instrukcji zaczyna się po impulsie („tyknięcie” (ang. tick)) zegara
- ❖ Odstęp między dwoma impulsami to cykl zegara (ang. clock cycle time) $\tau = 1/f$
 - ❖ f - zwykle 1MHz-3GHz
- ❖ I_c – liczba wykonanych instrukcji maszynowych (w odcinku czasu)
- ❖ CPI - średnia ilość cykli potrzebnych do wykonania instrukcji
 - ❖ CPI_i - średnia ilość cykli na instrukcję typu i ; I_i – liczba instrukcji typu i
 - ❖ $CPI = (I_c)^{-1} \sum_i CPI_i * I_i$

Zegar systemowy



From Computer Desktop Encyclopedia
1998, The Computer Language Co.

Miary wydajności

❖ T – czas CPU potrzebny do wykonania programu

- ❖ $T = I_c * CPI * \tau$

- ❖ $T = I_c * [p + m * k] * \tau$

- ❖ p – liczba cykli procesora do zdekodowania i wykonania instrukcji;
m – liczba odwołań do pamięci; k – stosunek cyklu pamięci do cyklu procesora

❖ MIPS - Millions of instructions per second

- ❖ ile milionów instrukcji CPU wykonuje w ciągu sekundy

- ❖ $MIPS = (T * 10^6)^{-1} * I_c = (CPI * 10^6)^{-1} * f$

- ❖ miara wydajności CPU

❖ Ex. 2M instrukcji, f=400MHz:

- ❖ 60% - CPI=1 (ALU); 18% - CPI=2 (cache load/store); 12% - CPI=4 (skoki); 10% - CPI=8 (RAM)

- ❖ $CPI = 0.6 + 2 * 0.18 + 4 * 0.12 + 8 * 0.1 = 2.24$

- ❖ $MIPS = (2.24 * 10^6)^{-1} * (400 * 10^6) \approx 178$

Miary wydajności (c.d.)

❖ MFLOPS - Millions of floating point instructions per second

- ❖ ile milionów instrukcji zmiennoprzecinkowych CPU wykonuje w ciągu sekundy
- ❖ $\text{MFLOPS} = (\text{czas wykonania} * 10^6)^{-1} * (\text{liczba wykonanych w programie instrukcji zmiennoprzecinkowych})$
- ❖ miara stosowana przy oszacowaniu wydajności gier i programów obliczeniowych

❖ Miary zależne od zbioru instrukcji, architektury i organizacji CPU, hierarchii pamięci, sposobu tworzenia kodu maszynowego przez kompilator

- ❖ CISC: add B(X), C(X), A(X); RISC: L R1, B(X); L R2, C(X); A R1,R2,R3: ST R3, A(X)
- ❖ czas wykonania nadaje się do porównań jednakowych architektur

Benchmarking

- ❖ Benchmark - program do testowania wydajności
 - ❖ http://en.wikipedia.org/wiki/Benchmark_%28computing%29
 - ❖ napisany w języku wysokiego poziomu i przenaszalny na różne komputery
 - ❖ reprezentuje rodzaj programowania: systemowe, numeryczne,....
 - ❖ łatwy w pomiarze oraz szeroko rozpowszechniony
- ❖ Ex. System Performance Evaluation Corporation (SPEC)
 - ❖ <http://www.spec.org/benchmarks.html>
 - ❖ CPU2006 – zestaw do pomiaru aplikacji ograniczonych przez CPU
 - ❖ 17 zmiennoprzecinkowych programów w C, C++, Fortran
 - ❖ 12 programów na liczbach całkowitych w C, C++
 - ❖ 3 miliony linii kodu
 - ❖ $R_A = m^{-1} * \sum_{i \leq m} R_i$ – śr. arytmetyczna; $R_H = m * (\sum_{i \leq m} R_i^{-1})^{-1}$ – śr. harmoniczna
 - ❖ R_i – tempo wykonywania instrukcji dla i-tego benchmarku
 - ❖ dwie metryki: speed i rate
 - ❖ pojedyncze zadanie i przepustowość (ang. throughput)

SPEC Speed

- ❖ Pojedyncze zadanie
- ❖ Bazowy czas wykonania na maszynie referencyjnej (ang. reference machine)
- ❖ Wyniki jako stosunek czasu referencyjnego do czasu wykonania
 - ❖ T_{ref_i} – czas wykonania benchmarku i na maszynie referencyjnej
 - ❖ T_{sut_i} – czas wykonania benchmarku i na maszynie testowej

$$r_i = \frac{T_{ref_i}}{T_{sut_i}}$$

- ❖ Całkowita wydajność obliczana za pomocą średniej geometrycznej

$$r_G = \left(\prod_{i=1}^n r_i \right)^{1/n}$$

SPEC speed (przykład)

❖ Sun Blade 6250

- ❖ 400.perlbench – 17.5
- ❖ 401.bzip2 – 14.0
- ❖ 403.gcc – 13.7
- ❖ 429.mcf – 17.6
- ❖ 445.gobmk – 14.7
- ❖ 456.hmmer – 18.6
- ❖ 458.sjeng – 17.0
- ❖ 462.libquantum – 31.3
- ❖ 464.h264ref – 23.7
- ❖ 471.omnetpp – 9.23
- ❖ 473.astar – 10.9
- ❖ 483.xalancbmk – 14.7
- ❖ $(17.5 * 14 * 13.7 * 17.6 * 14.7 * 18.6 * 17 * 31.3 * 23.7 * 9.23 * 10.9 * 14.7)^{1/12} \approx 18.5$

SPEC Rate

- ❖ Mierzy przepustowość maszyny wykonującej wiele programów
- ❖ Wiele kopii benchmarków uruchamianych jednocześnie
 - ❖ zwykle tyle ile procesorów
- ❖ Wyniki obliczane jako:
 - ❖ T_{ref_i} - czas wykonania benchmarku i na maszynie referencyjnej
 - ❖ N – liczba kopii wykonywanych jednocześnie
 - ❖ T_{sut_i} – czas który upłynął od startu benchmarku i na wszystkich N procesorach do momentu zakończenia wszystkich kopii programu na maszynie testowej

$$r_i = \frac{N \times T_{ref_i}}{T_{sut_i}}$$

- ❖ Całkowita wydajność obliczana za pomocą średniej geometrycznej

Prawo Amdahla

- ❖ Potencjalne przyspieszenie programu przy wykonaniu na wielu procesorach (Gene Amdahl, 1967)
 - ❖ Część f kodu zrównoleglonego (ang. infinitely parallelizable) bez szeregowania
 - ❖ Część $(1-f)$ kodu wykonywanego instrukcja po instrukcji
 - ❖ T – całkowity czas wykonania programu na procesorze
 - ❖ N – liczba procesorów wykorzystujących kod zrównoleglony

$$\text{Speedup} = \frac{\text{time to execute program on a single processor}}{\text{time to execute program on } N \text{ parallel processors}} = \frac{T(1-f) + Tf}{T(1-f) + \frac{Tf}{N}} = \frac{1}{(1-f) + \frac{f}{N}}$$

❖ Wnioski

- ❖ Jeśli małe f to małe przyspieszenie (ang. speedup)
- ❖ Jeśli $N \rightarrow \infty$ to przyspieszenie ograniczone przez $1/(1-f)$

Literatura podstawowa

- ❖ A. Silberschatz, P.B. Galvin, Podstawy systemów operacyjnych, wyd. III, WNT, 2000 (tł. 5th ed.)
- ❖ A. Silberschatz, P.B. Galvin, G. Gagne, Podstawy systemów operacyjnych, WNT, 2005, 2006 (tł. 6th ed.)
 - ❖ <http://os-book.com>
- ❖ W. Stallings, Organizacja i architektura systemu komputerowego. Projektowanie systemu a jego wydajność, PWN, Warszawa, 2000 (tł. 4th ed.)
- ❖ W. Stallings, Organizacja i architektura systemu komputerowego. Projektowanie systemu a jego wydajność, wyd. III, PWN, Warszawa, 2004 (tł. 6th ed.)
 - ❖ <http://WilliamStallings.com>
- ❖ W. Stallings – Systemy operacyjne, PWN, 2006 (tł. 5th ed.)
 - ❖ <http://williamstallings.com/OS/OS5e.html>

Literatura dodatkowa

- ◆ D. J. Struik, Krótki zarys historii matematyki, PWN, 1960
- ◆ A. Я. Хинчин, Цепные дроби, Москва, 1961
- ◆ C. B. Boyer, Historia rachunku różniczkowego i całkowego i rozwój jego pojęć, PWN, 1964
- ◆ S. M. Ulam, Maszyny liczące, *w: Matematyka w świecie współczesnym*, PWN, 1966
- ◆ H. Katzan, Jr., Computer Organization and the System/370, Van Nostrand, New York, 1971
- ◆ R. W. Gosper, Continued fraction arithmetic, unpublished, 1977
 - ◆ <http://www.tweedledum.com/rwg/cfup.htm>
- ◆ H. Kaufmann, Dzieje komputerów, PWN, 1980
- ◆ N. M. Beskin, Fascinating fractions, Mir, Moscow, 1986
- ◆ R. L. Graham, D. E. Knuth, O. Patashnik, Matematyka konkretna, PWN, 1996
- ◆ D. Wells, The Penguin Dictionary of Curious and Interesting Numbers, rev. ed., 1997
- ◆ D. H. Bailey, J. M. Borwein, R. E. Crandall, On the Khintchine Constant, *Math. Comp.*, 66, 217, 417-431, 1997
- ◆ J. H. Conway, R. K. Guy, Księga liczb, WNT, 1999
- ◆ P. Flajolet, B. Vallée, I. Vardi, Continued fractions from Euclid to the present day, preprint, 2000

Literatura (c.d.)

- ❖ S. Singh, Księga szyfrów, Albatros, 2001
- ❖ D. E. Knuth, Sztuka programowania, WNT, 2002
- ❖ J. Glenn Brookshear, Informatyka w ogólnym zarysie, WNT, 2003
- ❖ G. Johnson, Na skróty przez czas, Prószyński i S-ka, 2005
- ❖ G. Ifrah, Historia powszechna cyfr, W.A.B., 2006
- ❖ A. S. Tanenbaum, Strukturalna organizacja systemów komputerowych, Helion, 2006
- ❖ S. Y. Yan, Teoria liczb w informatyce, PWN, 2006
- ❖ T. Brabec, R. Lórencz, Arithmetic Unit based on Continued Fractions, submitted for review to ECI 2006
 - ❖ <http://service.felk.cvut.cz/anc/brabect1/pub/eci06.pdf>
- ❖ P. Loya, Amazing and Aesthetic Aspects of Analysis, Binghamton, 2006
- ❖ M. Grajek, ENIGMA. Blżej prawdy, Poznań, 2007
- ❖ <http://mathworld.wolfram.com/ContinuedFraction.html>
- ❖ <http://mathworld.wolfram.com/KhinchinsConstant.html>
- ❖ http://pl.wikipedia.org/wiki/Field_Programmable_Gate_Array
- ❖ A. Silberschatz, P.B. Gavin, G. Gagne, Operating Systems Concepts, 8th ed., Pearson, 2010
- ❖ W. Stallings, Computer Organization and Architecture, 8th ed., Pearson, 2010
- ❖ A. Tanenbaum, Systemy operacyjne, wyd. III, Helion, 2010
- ❖ <http://en.wikipedia.org/wiki/Abacus>

Podsumowanie (1)

- ❖ Architektura - atrybuty komputera widoczne dla programisty: lista rozkazów
- ❖ Organizacja - atrybuty komputera niewidoczne dla programisty: interfejsy, sygnały sterujące
- ❖ Przykłady architektur: Intel x86(IA32), IBM 360/370, Sun SPARC, SGI MIPS, Digital Alpha, HP PA, DEC VAX,.....

Podsumowanie (2)

- ❖ System komputerowy opisuje się zwykle od “góry do dołu”
- ❖ Struktura to sposób wzajemnego powiązania składników
- ❖ Funkcje określają działania poszczególnych składników jako części struktury

Podsumowanie (3)

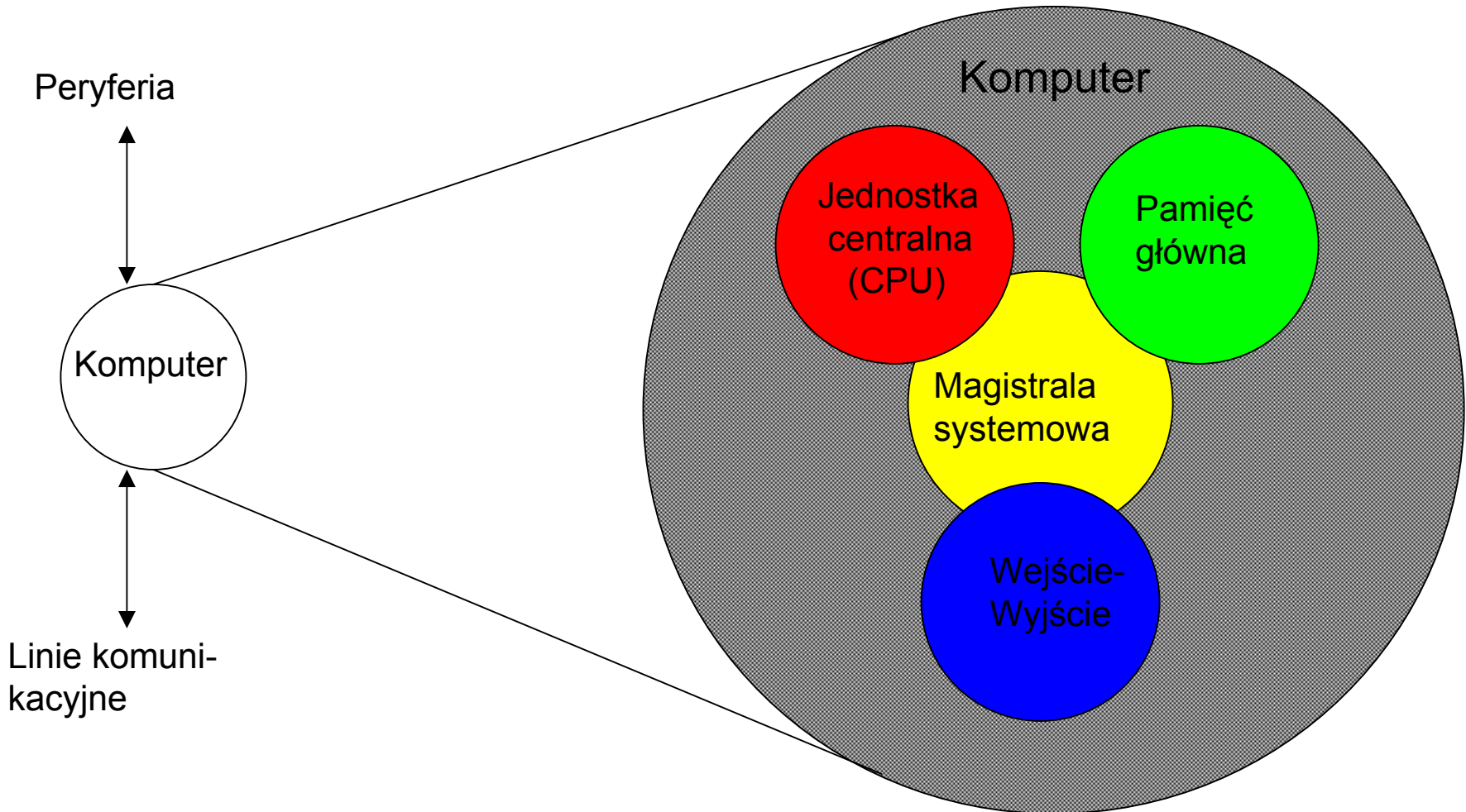
❖ Cztery funkcje komputera:

- ❖ przetwarzanie
- ❖ przemieszczanie
- ❖ przechowywanie danych
- ❖ sterowanie

❖ Struktura komputera:

- ❖ jednostka centralna (CPU, procesor)
- ❖ pamięć główna
- ❖ wejście/wyjście
- ❖ połączenia systemu

Podsumowanie (4)



Podsumowanie (5)

- ❖ Maszyna von Neumanna to koncepcja przechowywania danych i programów w pamięci głównej komputera
- ❖ Ewolucja komputerów a rozwój technologii
 - ❖ lampa próżniowa: Edison, 1873; Fleming (dioda), 1904; de Forest (trioda), 1907
 - ❖ tranzystor: Shockley, Bell Labs, 1947
 - ❖ układ scalony, Kilby, Texas Instruments, 1958
 - ❖ pamięć półprzewodnikowa: Fairchild Semiconductor, 1970
 - ❖ mikroprocesor: Intel, Texas Instruments, Garret Air Research, 1970

Podsumowanie (6)

- ❖ Zerowa generacja – 1600-1945 (Pascal, Babbage, Zuse)
- ❖ Pierwsza generacja - 1945-1955
 - ❖ lampa próżniowa: ENIAC, IAS
 - ❖ open shop
- ❖ Druga generacja - 1955-1965
 - ❖ tranzystor: mainframes; IBM7094 - wydzielenia kanałów I/O, DEC PDP-1
 - ❖ monitor, system wsadowy
- ❖ Trzecia generacja - 1965-1980
 - ❖ układy scalone: IBM/360 - I/O multiplexer, PDP-8 - magistrala OMNIBUS
 - ❖ spooling, wieloprogramowanie, podział czasu
 - ❖ CTSS, MULTICS, MFT, MVT/TSO, Berkeley UNIX
- ❖ Czwarta generacja - 1980-2007
 - ❖ mikroprocesory o bardzo wielkiej skali integracji: IBM PC, Lisa (Apple)
 - ❖ GUI, systemy rozproszone, multimedia
 - ❖ Windows, OS/2, Mac OS X, Solaris, Linux, FreeBSD, AIX, HP-UX, True64