

Jednostka centralna

dr hab. inż. Krzysztof Patan, prof. PWSZ

Instytut Politechniczny
Państwowa Wyższa Szkoła Zawodowa w Głogowie
k.patan@issi.uz.zgora.pl

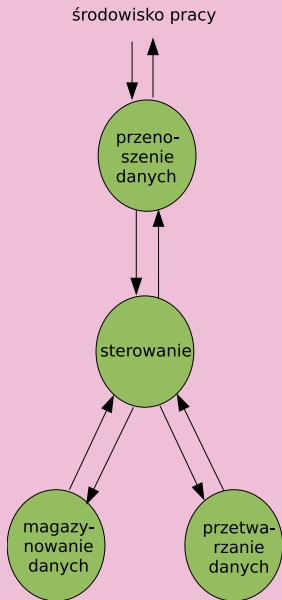
Architektura i organizacja komputerów

- **Architektura** – atrybuty widoczne dla programisty
 - zbiór instrukcji
 - reprezentacja binarna danych (liczba bitów)
 - mechanizmy wejścia/wyjścia
 - techniki adresowania
- **Organizacja** – sposób zaimplementowania właściwości
 - sygnały sterujące
 - interfejsy
 - techniki wykonania pamięci
- Wszystkie procesory rodziny Intel x86 mają tę samą podstawową architekturę
- Rodzina systemów IBM 370 ma wspólną podstawową architekturę
- Organizacja systemu różni się w poszczególnych wersjach

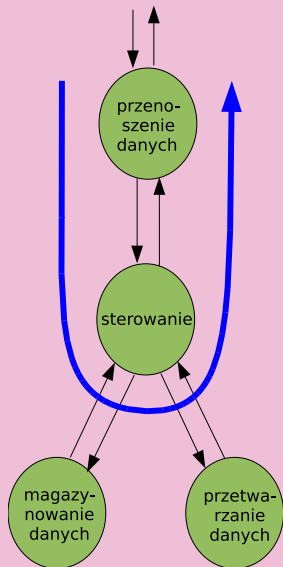
Struktura i funkcje systemu komputerowego

- **Struktura** – sposób powiązania komponentów systemu ze sobą
- **Funkcja** – funkcjonowanie pojedynczego komponentu jako części w ramach całej struktury
- Funkcje komputera
 - przetwarzanie danych
 - przechowywanie danych
 - przenoszenie danych
 - sterowanie

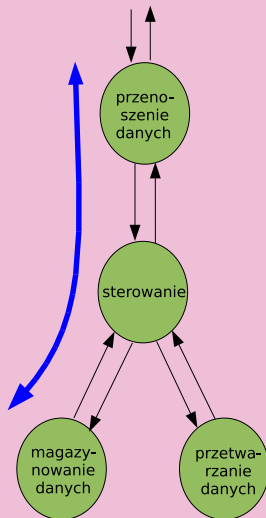
Widok funkcjonalny systemu komputerowego



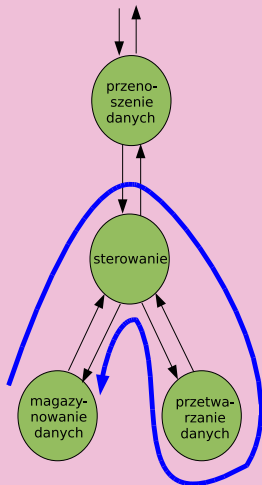
przenoszenie danych



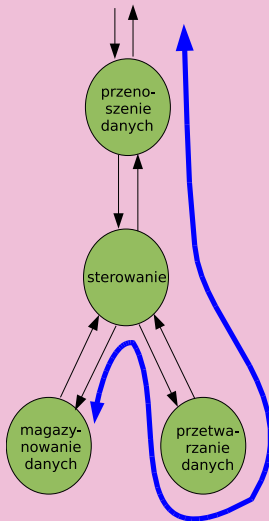
magazy-nowanie danych



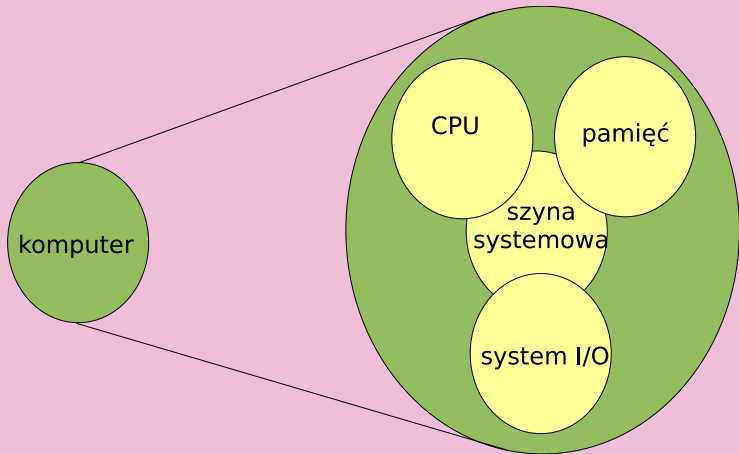
przetwarzanie danych z/do magazynu



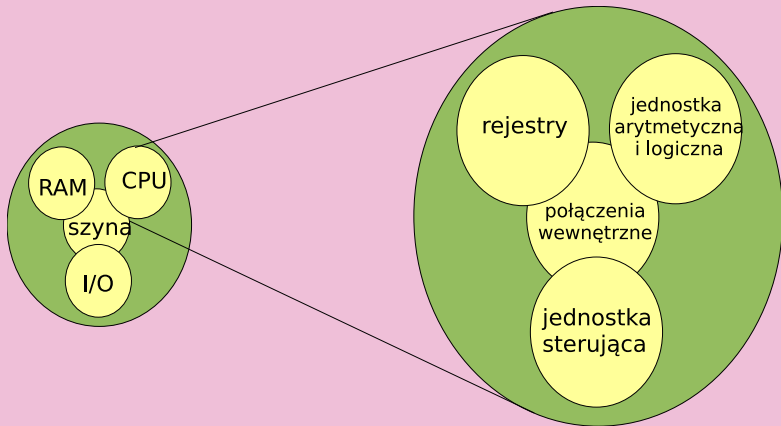
przetwarzanie danych z magazynu do I/O



Struktura systemu



Struktura procesora



ENIAC – Electronic Numerical Integrator and Computer

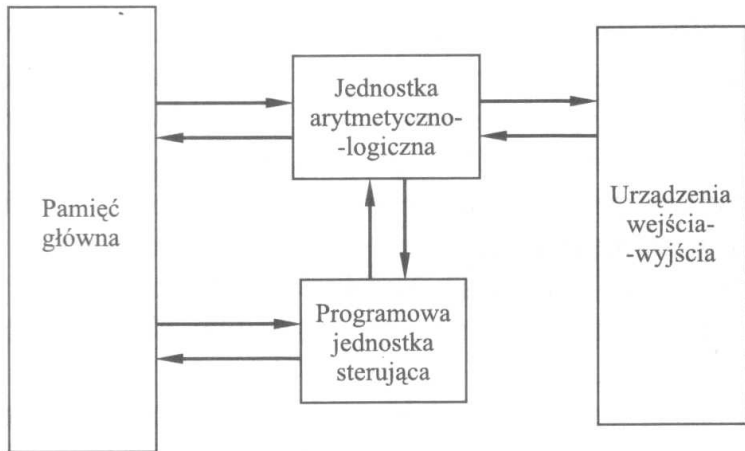
- Pierwszy całkowicie elektroniczny kalkulator ogólnego przeznaczenia (1946r.)
- Waga: 30 ton, zajmowana powierzchnia: $72 m^2$, zużycie energii 200KW
- Wydajność: 5000 operacji dodawania na sekundę, operacja dodawania – $200\mu s$, mnożenia – $6ms$
- Przetwarzanie dziesiętne (nie binarne)
- 20 akumulatorów dziesięciocyfrowych
- Programowanie manualne za pomocą przełączników
- Zastosowanie: przeliczanie trajektorii pocisków

`kąt = f(położenie, siła wiatru, gęstość
powietrza, temperatura, waga pocisku, materiał
napędowy, ...)`

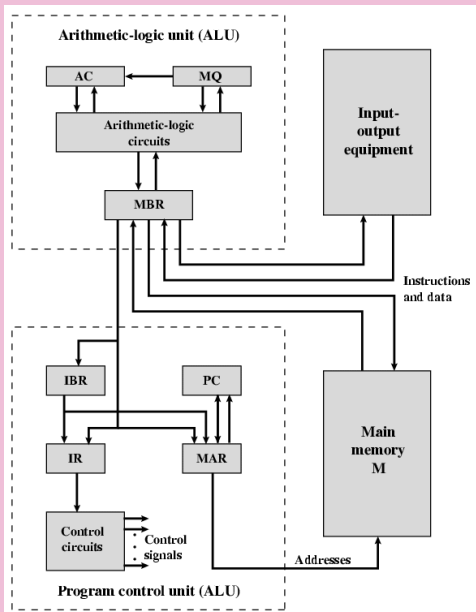
Architektura von Neumanna

- Projekt ukończony w Princeton Institute for Advanced Studies (IAS) w roku 1952
- Koncepcja programu przechowywanego, pamięć typu read/write
- Pamięć główna przechowuje programy i dane
- Jednostka arytmetyczna operuje na danych binarnych
- Jednostka sterująca interpretuje instrukcje pobrane z pamięci i następnie je wykonuje
- Sprzęt wejścia/wyjścia obsługiwany jest przez jednostkę sterującą
- Rejestry procesora (pamięć CPU)
 - MBR (ang. *Memory Buffer Register*)
 - MAR (ang. *Memory Address Register*)
 - IR (ang. *Instruction Register*)
 - IBR (ang. *Instruction Buffer Register*)
 - MQ (ang. *Multiplier Quotient*)
 - Licznik rozkazów PC (ang. *Program Counter*)
 - Akumulator (ang. *Accumulator*)

Architektura von Neumanna



Widok szczegółowy komputera IAS



- 1953 – IBM 701, pierwszy komputer z pamięcią read/write do obliczeń naukowych
- 1955 – IBM 702, komputer do zastosowań biznesowych
- 1957 – DEC PDP-1, komputer tranzystorowy
- 1964 – seria IBM 360, pierwsza rodzina komputerów
- 1964 – DEC PDP-8, pierwszy minikomputer
- 1970 – pierwsza pamięć półprzewodnikowa
- 1971 – Intel 4004, pierwszy mikroprocesor
- 1972 – Intel 8008, 8 bitowy
- 1974 – Intel 8086, pierwszy procesor ogólnego zastosowania

Generacje komputerów

- lampowe – 1946-1957
- tranzystorowe - 1958-1964
- mała skala integracji – od 1964 do 100 urządzeń w procesorze
- średnia skala integracji – do 1971 od 100 do 3000 urządzeń w procesorze
- duża skala integracji – 1971-1977 od 3000 – 100000 urządzeń w procesorze
- bardzo duża skala integracji – 1978–1991 od 100000 do 100000000 urządzeń w procesorze
- ultra duża skala integracji – od 1991 ponad 1000000000 urządzeń w procesorze

Zestaw instrukcji

- Zestaw instrukcji komputera – kluczowy element architektury komputera
- Zestaw instrukcji jest interfejsem pomiędzy sprzętem, a oprogramowaniem
- Zestaw instrukcji można rozpatrywać jako atrybuty komputera widziane z poziomu programisty, np. struktura konceptualna i zachowanie funkcjonalne o odróżnieniu od przepływu danych czy fizycznej implementacji, itd.
- ISA – architektura zestawu instrukcji (ang. *Instruction Set Architecture*)
- Zagadnienie kompatybilności ISA – ten sam zbudowany program można uruchomić na poprzednich kompatybilnych wersjach systemu, np. IBM 360/370, Intel x86 (IA32), IA64

- ISA definiuje możliwości implementacji
- ISA określa format instrukcji (kodowanie binarne) oraz semantykę instrukcji (składnia)
- przykłady: MIPS, x86, IBM 360, JVM
- wiele możliwości implementacji jednego zestawu instrukcji
 - linia IBM 360: model 30 (rok 1964), model z900 (rok 2001)
 - platforma x86: 8086 (rok 1978), 80186, 286, 386, 486, Pentium, Pentium Pro, Pentium-4 (rok 2000), AMD Athlon, Transmeta Crusoe, SoftPC
 - implementacje MIPS: R2000, R4000, R10000
 - implementacje JVM: HotSpot, PicoJava, ARM Jazelle

Problem kompatybilności w firmie IBM

W latach 60 firma IBM posiadała 4 niekompatybilne linie komputerów

701, 650, 702, 1401

Każdy system posiadał własny

- zbiór instrukcji
- podsystem wejścia/wyjścia i pamięć pomocniczą
- asemblery, kompilatory, biblioteki
- rynek zbytu: zastosowania biznesowe, naukowe, czasu rzeczywistego

Jak wybrać dobry zestaw poleceń?

- zestaw poleceń zapewniający prosty interfejs programowy, a także umożliwiający szybkie i efektywne oprogramowanie sprzętu
- kompatybilność z poprzednimi wersjami systemu
- przykłady występujących trudności:
 - współczesne komputery posiadają procesory zawierające rejestry o większej pojemności niż cała pamięć główna wczesnych komputerów
 - współczesny procesor zawiera setki razy więcej tranzystorów niż cały komputer z lat 60

Wczesny zbiór instrukcji

LOAD	$AC \leftarrow M[x]$
STORE	$M[x] \leftarrow (AC)$
ADD	$AC \leftarrow (AC) + M[x]$
SUB	
MUL	rejestr MQ
DIV	
SHIFT LEFT	$AC \leftarrow 2 \times (AC)$
SHIFT RIGHT	
JUMP	$PC \leftarrow x$
JGE	if $(AC) \geq 0$ then $PC \leftarrow x$
LOAD ADR	$AC \leftarrow \text{dokładny adres}(M[x])$
STORE ADR	

Zazwyczaj około 20 instrukcji

Wydajność procesora

$$CW = IPP \times CPI \times TPC$$

- CW – czas wykonania programu
- IPP – ilość instrukcji programu (ang. *Instructions per program*) zależy od kodu źródłowego, kompilatora i architektury zbioru instrukcji ISA
- CPI – ilość cykli na instrukcję (ang. *Cycles per instruction*) zależy od architektury zbioru instrukcji ISA i mikroarchitektury
- TPC – czas cyklu (ang. *Time per cycle*) zależy od mikroarchitektury i technologii wykonania

Mikroarchitektura	CPI	TPC
mikrocodowanie	>1	krótki
pojedynczy-cykl	1	długi
“pipelined”	1	krótki

Rejestry procesora

W procesorach znajdują się rejestry udostępniające szybszą pamięć o mniejszej pojemności w stosunku do pamięci operacyjnej

Rejestry widoczne dla użytkownika

Procesor może odwoływać się do rejestru widocznego dla użytkownika za pomocą języka maszynowego, który proces wykonuje

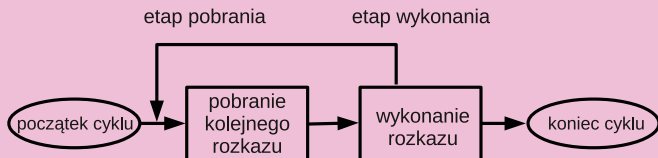
- rejestry danych – są to rejestry ogólnego przeznaczenia, które są wykorzystywane przez dowolne instrukcje przetwarzające dane. Mogą istnieć rejestry dedykowane do operacji zmiennoprzecinkowych czy do operacji na liczbach całkowitych
- rejestry adresowe – zawierają adresy danych lub rozkazów w pamięci operacyjnej. Mogą to być rejestry
 - rejestr indeksowy – adresowanie indeksowe jest typowym sposobem adresowania, który dodaje indeks do wartości bazowej
 - wskaźnik segmentu – rejestr przechowuje adres początkowy segmentu; w takim przypadku rejestrów jest więcej
 - wskaźnik stosu – rejestr wskazuje lokalizację na szczycie stosu; stos jest zestawem lokalizacji umieszczanej jedna na drugiej

Rejestry sterowania i stanu

- Do kontrolowania działania procesora służy wiele różnych rejestrów
- W większości systemów komputerowych są to rejestry niewidoczne dla użytkownika
- MAR – rejestr adresowy pamięci – określa adres w pamięci dla kolejnej operacji odczytu bądź zapisu
- MBR – rejestr buforowy pamięci – zawiera dane, które mają zostać zapisane/odczytane w pamięci/z pamięci
- I/O AR – rejestr adresowy wejścia/wyjścia – określa urządzenie wejścia/wyjścia
- I/O BR – rejestr buforowy wejścia/wyjścia – używany do wymiany danych pomiędzy modułem wejścia/wyjścia, a CPU
- PC – licznik rozkazów – zawiera adres kolejnego rozkazu do pobrania
- IR – rejestr rozkazów – zawiera ostatnio pobrany rozkaz
- kody warunków (flagi) – bity ustawiane przez osprzęt procesora w rezultacie wykonywania działań

Wykonanie programu

- Program można traktować jako sekwencję rozkazów
- Dla każdego rozkazu wykonywane są operacje arytmetyczne i logiczne
- Dla każdej operacji, wymagany jest różny zbiór sygnałów sterujących
- Operacje związane z przetworzeniem jednego rozkazu noszą nazwę *cyklu rozkazu*
- Dwie fazy wykonywania rozkazu:
 - 1 faza pobrania
 - 2 faza wykonania



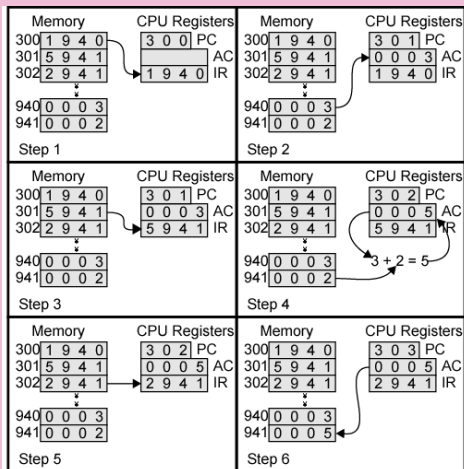
Faza pobrania

- Licznik programu (PC) przechowuje adres kolejnej instrukcji do pobrania
- Procesor pobiera instrukcję z jednostki pamięci wskazanej przez rejestr PC
- Zostaje zwiększona wartość licznika programu (PC)
- Pobrany rozkaz zostaje załadowany do rejestru rozkazów (IR)
- Procesor interpretuje rozkaz i wykonuje żądane operacje

Faza wykonania

- Operacje wykonywane przez procesor można podzielić na cztery kategorie:
 - ① Procesor–pamięć – dane można przenosić z CPU do pamięci lub odwrotnie
 - ② Procesor–wejście/wyjście – dane mogą być przenoszone pomiędzy CPU, a modułem wejścia/wyjścia
 - ③ Przetwarzanie danych – CPU może realizować na danych działania arytmetyczne lub logiczne
 - ④ Sterowanie – rozkaz może zawierać dyrektywę określającą zmianę kolejności wykonywania rozkazu, np. skok
- W ogólności procesor może wykonać działanie będące kombinacją wyżej wymienionych

Przykład wykonania sekwencji rozkazów



Wewnętrzne rejestry CPU
 Licznik programu (PC) = adres rozkazu
 Rejestr rozkazu (IR) = wykonanie rozkazu
 Akumulator (AC) = pamięć tymczasowa

Lista kodów operacji
 0001 = pobranie akumulatora z pamięci
 0010 = zapisanie akumulatora w pamięci
 0101 = wstawianie danych z pamięci do akumulatora

Rozwój procesorów

- Rozbudowa zbioru rozkazów
- Koprocesory arytmetyczne
- Pamięć podręczna
- Kompatybilność
- Zwiększenie zbioru rozkazów pozwala na:
 - generowanie krótszego i efektywniejszego kodu programu
 - zmniejszenie liczby odwołań do pamięci
 - zmniejszenie obszaru alokowanej pamięci
- Kwestionowanie złożonego zbioru instrukcji (koniec lat 70)
 - niewielki zbiór instrukcji może być bardziej efektywnie wykorzystywany przez kompilatory
- Przełom lat 80/90 – opracowanie procesorów o zredukowanej liście rozkazów

CISC - Complex Instruction Set Computer

- architektura dominująca w rodzinach Intel x86 i Motorola 680xx
- procesory realizują coraz większe zadania
- pojedynczy rozkaz wywołuje szereg kompleksowych działań
- czas opracowywania polecenia może dochodzić nawet do kilkudziesięciu cykli zegarowych
- kod programu zwarty
- proces transportu kodu do procesora powoduje mniej problemów
- lista rozkazów od 100 do 200
- duża liczba trybów adresowania od 5 do 20
- duży zakres rozkazów o różnych długościach i czasach wykonania
- mikroprogramowalna jednostka sterująca

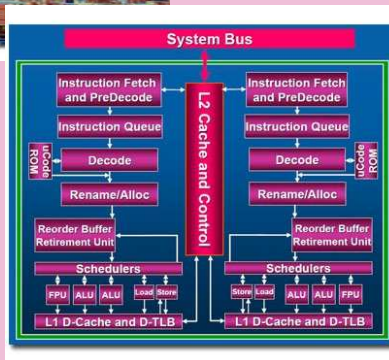
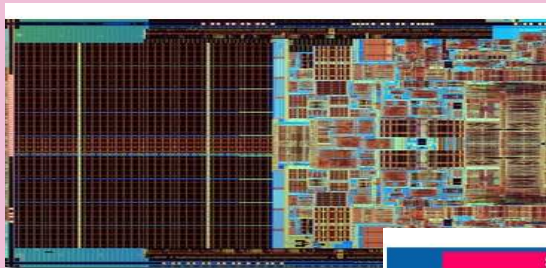
RISC - Reduced Instruction Set Computer

- powstał w toku prac nad projektem 801 firmy IBM
- rozwijany przez wielu producentów - AMD 29000, HP PA-RISC, Intel 860 i 960, IBM RS/6000
- ograniczona lista rozkazów do niewielu szybko wykonywanych instrukcji
- realizacja sprzętowa jednostki sterującej
- duże obciążenie magistrali pamięciowej
- wymagana duża przepustowość magistrali
- stała długość wszystkich mikrorozkazów
- niewielka liczba trybów adresowania (kilka)

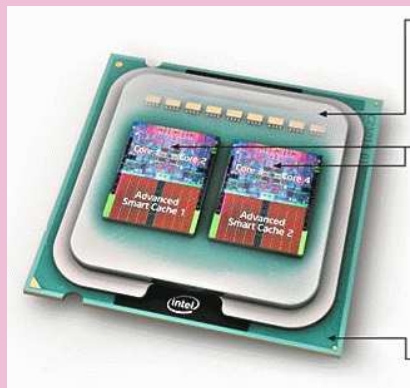
Intel Pentium 4



Intel Core Duo



Intel Core Quadro



radiator

chipy

podstawka

