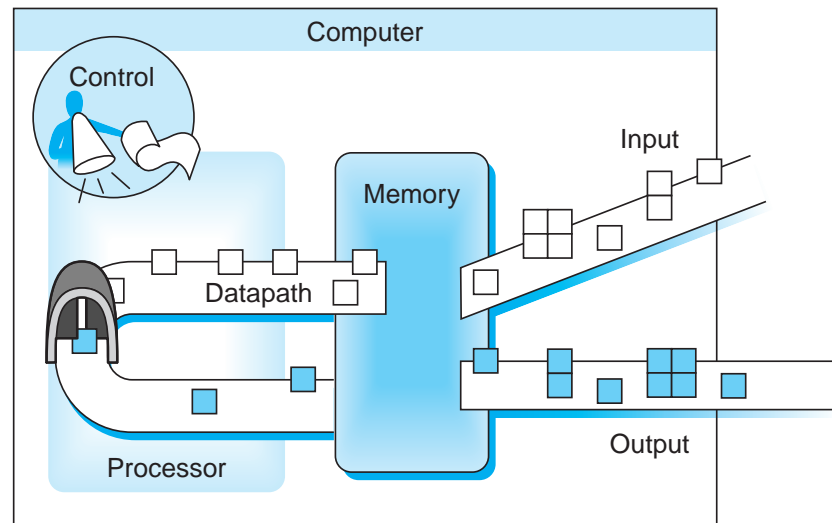


Architektury systemów komputerowych

(Wykład 1, 7.X.2009)



Literatura

1. David Patterson, John Hennessy, *Computer Organization and Design, 4th Edition*, Morgan Kaufmann 2009.
Z tej książki pochodzi większość rysunków do moich slajdów.
2. Digital Design and Computer Architecture, David Harris, Sarah Harris, Morgan Kaufmann 2007.
3. William Stallings, *Organizacja i architektura systemu komputerowego. Projektowanie systemu a jego wydajność*, WNT 2004.
4. Linda Null, Julia Lobur, *Struktura organizacyjna i architektura systemów komputerowych*, Helion 2004
5. B. S. Chalk, *Organizacja i architektura komputerów*, WNT 1998;
6. M. Morris Mano, *Architektura komputerów*, WNT 1988.

Ćwiczenia

- Na ćwiczeniach obowiązuje system z deklaracjami
- Szczegóły są podane na mojej stronie internetowej:
www.ii.uni.wroc.pl/~kiero/dydaktyka.html

Czego będziemy się uczyć?

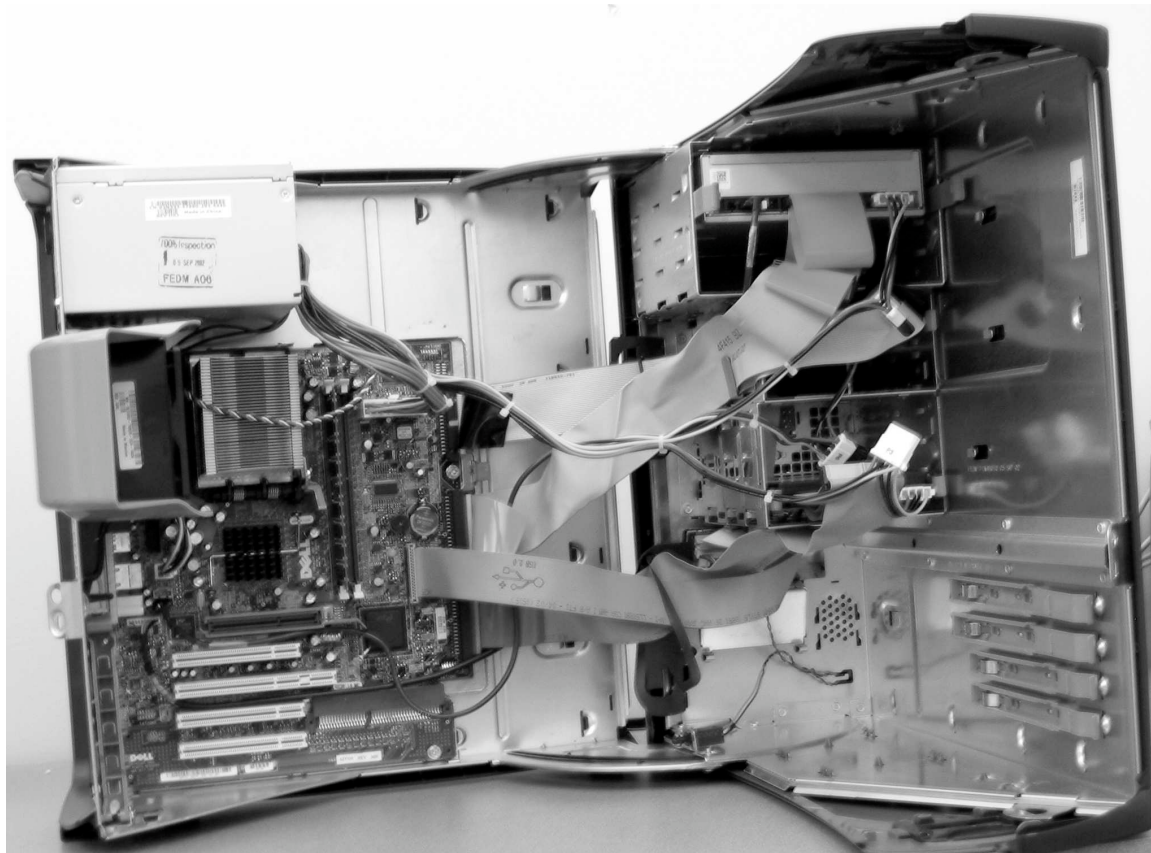
- Wykład przeznaczony jest głównie dla studentów pierwszego roku.
- Poznawać będziemy ogólne zasady budowy i działania systemów komputerowych.
- Omawiać będziemy głównie uproszczone rozwiązania modelowe.

Hierarchiczne spojrzenie na budowę komputera



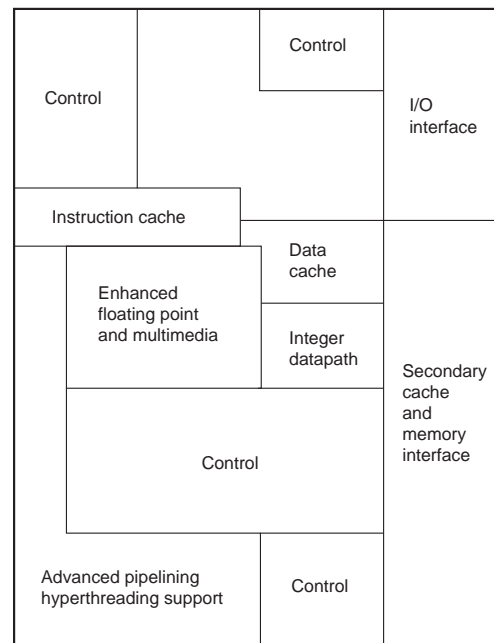
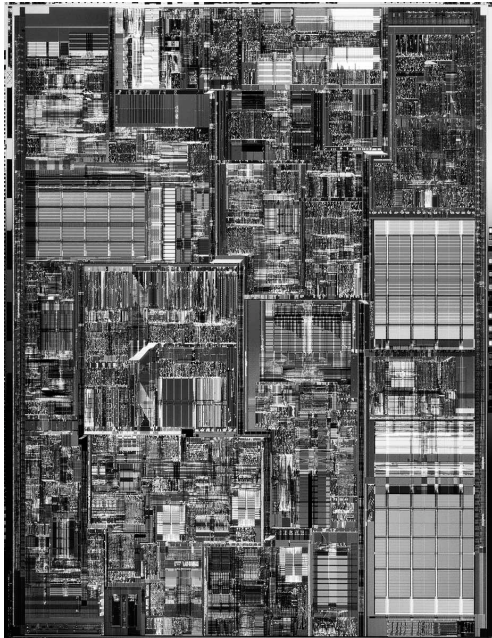
Typowy komputer osobisty.

Hierarchiczne spojrzenie na budowę komputera



Wnętrze typowego komputera osobistego.

Hierarchiczne spojrzenie na budowę komputera



PAT01F09B.eps

Rzut oka na procesor Pentium 4.

Hierarchiczne spojrzenie na budowę komputera

Poziomy postrzegania systemu komputerowego:

- Zwykły użytkownik
- Języki wysokiego poziomu
- Asemblery / języki maszynowe
- Logiczna budowa procesora, pamięci, I/O, połączenia między nimi
- Bramki logiczne

Nas interesować będą trzy najniższe poziomy.

Proces kompilacji programu

High-level
language
program
(in C)

```
swap(int v[], int k)
{int temp;
  temp = v[k];
  v[k] = v[k+1];
  v[k+1] = temp;
}
```

↓
Compiler

Assembly
language
program
(for MIPS)

```
swap:
  muli $2, $5, 4
  add  $2, $4, $2
  lw   $15, 0($2)
  lw   $16, 4($2)
  sw   $16, 0($2)
  sw   $15, 4($2)
  jr   $31
```

↓
Assembler

Binary machine
language
program
(for MIPS)

```
000000001010000100000000000011000
00000000000110000001100000100001
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
0000001111100000000000000001000
```

Architektura zbioru rozkazów

- Jeden z kluczowych poziomów abstrakcji
- Lista rozkazów procesora
- Informacje o rejestrach, sposobie dostępu do pamięci, obsłudze urządzeń I/O
- Architektura ilustrującą omawiane podczas tego wykładu zagadnienia będzie MIPS
- Przykładowe rozkazy:
 - `add $s1, $s2, $s3`
 - `sw $s1, (4)$s2`
 - `beq $s1, $s2, addr`

Jak przebiegać będzie wykład

- Poziom bramek logicznych
 - Układy kombinacyjne
 - Układy sekwencyjne
- Reprezentacja danych i arytmetyka
- Architektura MIPS
- Realizacja architektury MIPS z układów kombinacyjnych i sekwencyjnych. Cel: wyobrazić sobie jak zbudować komputer realizujący pewnien (bogaty) podzbiór listy rozkazów MIPS z uwzględnieniem działania systemu pamięci cache oraz przetwarzania potokowego.

Model von Neumanna

Koncepcja pojawiła się w latach 40-tych XX w. Podobne pomysły miał w tym samym czasie Alan Turing.

Podstawowa cecha: **program przechowywany jest w pamięci** razem z danymi.

- koncepcja programu (rozkazy wykonywane są szeregowo, z wyjątkiem skoków)
- program i dane w pamięci głównej, adresowanie
- ALU operuje na danych binarnych
- cykl pobieranie-dekodowanie-wykonywanie rozkazów
- I/O nadzorowane jest przez CPU.

Wiele dodatkowych rozszerzeń (pamięć wirtualna, dane zmiennoprzecinkowe, przerwania, potokowość, wieloprocessorowość...), ale, z grubsza rzecz biorąc, model obowiązuje do dziś.