Procesory

# CPU

* =Central Processing Unit
* Integrovaný obvod, který provádí strojové instrukce
* Skládá se z:
  + Řadiče (dekóduje instrukce)
  + ALU (operace s čísly, logické operace)
  + Registry (vnitřní malá ale rychlá paměť pro různé účely)
  + Cache

# Architektura CPU

* Architektura instrukční sady
* Mikroarchitektura

## Architektura instrukční sady

* Definuje počet registrů, syntax strojového kódu, seznam instrukcí, datové typy
* Nemá velký vliv na výkon procesoru
* Sady jsou většinou CISC nebo RISC (sdružení podle podobného návrhu)
* x86
  + Architektura instrukční sady pro 16/32/64bitové CPU (dnes se hlavně používá rozšíření x86-64)
  + Vznikla v 70. letech, přes četná vylepšení má prakticky úplnou zpětnou kompatibilitu
  + Ve vestavěných systémech (micro:bit) se používá hlavně rodina instrukčních sad ARM (advanced RISC machines)
  + Variabilní délka instrukce
    - v prvním bytu musí být zmíněna délka instrukce
    - primitivní instrukce zabírají méně místa
  + CISC
    - Navrženo v době, kdy bylo nutné šetřit pamětí
    - Jedna instrukce dokáže provádět komplexní operace (rozkládá se na posloupnost mikroinstrukcí)
    - Variabilní délka instrukce
    - Velký počet instrukcí a adresačních režimů
  + Zpětná kompatibilita
  + Bajtové adresovaní
    - Byte je nejmenší adresovatelná jednotka
    - Každý bajt má dvou adresu
    - Opak je word addressing (velikost slova závisí na architektuře)
  + SIMD – Single Instruction Multiple Data
    - Instrukce pracuje současně s vícero daty
  + Mikroarchitektura
  + Harvardská architektura
    - L1 je dělena na Data Cache a Instruction Cache
* ARM
  + nízká spotřeba
  + mobilní telefony, vestavěné systémy
  + RISC

## Mikroarchitektura

* =způsob implementace instrukční sady

# Rysy moderních CPU

* Paralelní zpracování
* Změna pořadí provádění instrukcí
* Tepelná ochrana
* Správa napájení

# Techniky optimalizace provádění instrukcí

## Techniky zvýšení výkonu

* Větší počet tranzistorů
* Větší frekvence
* Menší šířka spojů

## Rozšíření bitové šířky zpracovávaných dat

* Rozšíření počtu bitů, které mikroprocesory dokáží v každé instrukci zpracovávat

## Zvýšení počtu pracovních registrů

## Pipelining (třetězené provádění instrukcí)

* Současné zpracování většího množství instrukcí
* Každá instrukce se nachází v jiném stavu zpracování:
  + Načtení instrukce z operační paměti (fetch)
  + Dekódování instrukce v řadiči (decode)
  + Přenos obsahů pracovních registrů (read)
  + Vlastní provedení instrukce (execute)
  + Uložení výsledků zpět do pracovních registrů (write)

## Prediktory skoků

* Řešení problému „co se má stát s instrukcemi, které se nachází v rozpracovaném stavu v moment skoku“
* Odhadnutí, zda se skok provede či nikoli
* 1bitový prediktor si pamatuje výsledek skoku
* 2bitový prediktor si pamatuje výsledek skoku a předchozí stav predikce

# Techniky snižování spotřeby

* Zmenšení výrobní technologie
  + Vzdálenost mezi tranzistory
  + Kratší cesta mezi tranzistory => kratší cesta mezi elektrony => nižší spotřeba
* Snížení napájecího napětí
* Intel Turbo Boost nebo ARM Turbo Core
  + „Dynamické přetaktování“ – frekvence (a spotřeba) se zvýší, pokud je výkon potřeba
* AMD Cool and Quiet, Intel SpeedStep

# Rozšířené instrukční sady

* Skupina nových instrukcí, které rozšiřuj danou instrukční sadu
* Přidávají nové jednotky integrovaného obvodu nebo registry
* Skalár – hodnota proměnné plně určena jediným číselným údajem
* 7Vektor – obsahuje několik hodnot stejného typu

## x87

* Dnes se již považuje za součást x86
* Přidává operace pro práci s čísly s plovoucí řádovou čárkou

## MMX (Multi Media Extension)

* První rozšiřující sada obsahující SIMD operace pro x86
* Zaměření na multimédia
* Od Intelu

## 3DNow!

* Odpověď na MMX od AMD
* Podobné MMX, pracuje s plovoucí desetinou čárkou

## SSE (Streaming SIMD Extensions)

* Od společnosti Intel
* Odpověď na 3DNow!
* Nové instrukce, 128bitové registry

## x86-64

* 64bitový režim a režim kompatibility