

# Program

- opakování číselných soustav
- jednotky objemu dat
- počítání s číselnými soustavami
- návrh sčítačky odčítačky
- algoritmizace

# Návrh instrukční sady procesoru a návrh počítače

- Navrhněte počítač mající 8KB operační paměti (RAM) a 4KB ROM paměti.
- procesor může „oadresovat“ 64KB paměti
- navrhněte zapojení sběrnice
- Navrhněte instrukce pro procesor i registry počítače.

# Stručná historie procesorů x86

- Intel
  - [http://en.wikipedia.org/wiki/List\\_of\\_Intel\\_microprocessors](http://en.wikipedia.org/wiki/List_of_Intel_microprocessors)
  - 8086
    - představen 8 června 1978
    - 5, 8 a 10 MHz (0,33 0,66 0,75 MIPS – megainstrukce za sekundu)
    - 16 bitů data, 20 bitů adresa
  - 8088
    - osmibitová sběrnice (data) aby se mohli použít podpůrné obvody z :  
“osmibitů“
    - jinak identický s předchozím
    - Použit v IBM PC XT

# Stručná historie procesorů x86

- Intel
  - 80286
    - představen 1. února 1982
    - 6, 8 a 12.5 MHz (0,9 1,5 2,66 MIPS)
    - 3 až 6x rychlejší než 8086
    - 24 bitů adresa, 16 bitů datová sběrnice
    - nové instrukce přidány pro “chráněný režim”

# Stručná historie procesorů x86

- Intel
  - 80386
    - zvětšila se datová sběrnice na 32 bitů
    - adresová sběrnice má plných 32 bitů
    - SX – 1988
      - adresová sběrnice časově multiplexovaná – jen 16 bitů
      - 16, 20, 25, 33 Mhz (2,5 2,5 2,7 2,9 MIPS)
    - DX – 1985
      - plnokrevník
      - 16, 20, 25, 33 Mhz (5, 7, 8, 11.4 MIPS)
  - 80486 - 1989
    - DX – plnokrevník s integrovaným matematickým koprocесorem
      - 25, 33, 50 Mhz (20, 27, 41 MIPS)
    - SX – deaktivován (rok výroby 1991)
    - 50x rychlejší než 8086

# Stručná historie procesorů x86

- Intel
  - Pentium I (MMX, Pro)
    - změny ve vnitřní architektuře procesoru (superscalární)
    - 5x rychlejší než 486 na 33Mhz
  - Pentium II, Pentium III, Pentium 4
    - změny v architektuře procesoru, přidány SIMD instrukce MMX a SSE
    - 11000 MIPS

# Mikroprocesory intel x86 a klony

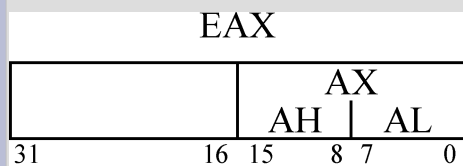
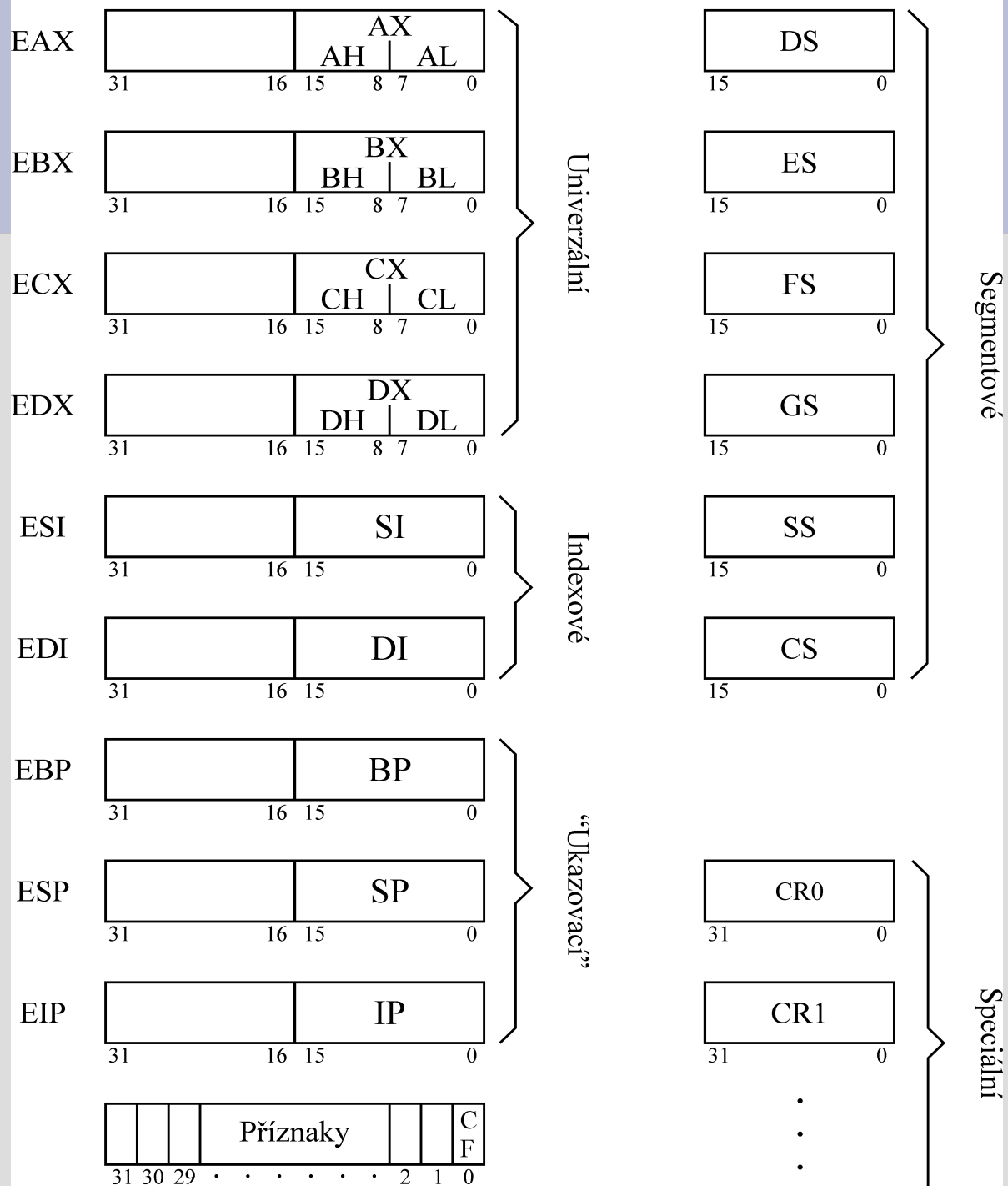
- Firmy AMD, NEC, Cyrix aj. okopírovali procesor a prodávali jej s vlastními vylepšeními ovšem zachovávající instrukční kompatibilitu
- Vydrželo AMD a stává se stále významnějším konkurentem Intelu
- [http://en.wikipedia.org/wiki/List\\_of\\_AMD\\_microprocessors](http://en.wikipedia.org/wiki/List_of_AMD_microprocessors)
- AMD:
  - AMD K5 současník prvního Pentia
  - AMD K6 současník Pentia II (později přidán 3DNow)
  - AMD K7
    - Athlon, Duron, Athlon XP, Sempron + “mobilní” varianty
  - AMD K8
    - Opeteron, Athlon 64

# Mikroprocesor 80386

- typický reprezentant x86
- pozdější procesory z hlediska programátora:
  - rychlejší
  - multimediální instrukce (MMX, 3DNow, SSE)
  - nové registry pro multimediální instrukce
  - přidáno pár nových instrukcí
  - integrován matematický koprocessor
- režimy procesoru
  - realný
    - kompatibilita s 8086
  - chráněný
    - podpora virtuální paměti
    - ochrana paměti
    - podpora multitaskingu

# Registry 80386

- Univerzální
- Indexovací
- “Ukazovací” - pointer
- Segmentové



Registr EAX

# Přerušeni

- událost v systému, která způsobí přerušeni právě probíhajícího programu (většinou kontroluje se mezi dokončenou a nově započatou instrukcí)
- proces začne zpracovávat program obsluhy přerušeni, po jeho dokončení se vrátí na přerušené místo
- Čtení knihy a vyrušení telefonem – stejný princip

# Přerušeni II

- typy přerušeni:
  - hardwarová
    - od periférií
      - od řadiče přerušeni IRQ0 – IRQ15
    - od procesoru (trap)
      - přerušeni se provádění instrukce pokud se zjistí chyba (neexistující paměť, neplatná instrukce apod)
  - softwarová
    - vektor přerušeni - ukazatel adresu programu obsluhy přerušeni
    - tabulka vektorů přerušeni - “pole vektoru přerušeni s adresami obslužných rutin
    - vyvolávají se instrukcí INT (nebo obdobnou) operand (argument) je číslo požadovaného přerušeni (na x86 0-255)

# Operandy instrukcí

- data se kterými instrukce pracuje
- píšší se za instrukci
- typy:
  - přímé – jsou součástí instrukce
  - registr – požadovaná hodnota je/bude uložena v registru
  - odkaz do paměti – součástí instrukce je ukazatel do paměti, kde se hodnota (bude) nachází
  - fixní – instrukce očekává operand přímo v nějakém neměnitelném registru
- podle “velikosti” operandu
  - 8, 16, 32 bitové

# Adresace paměti

- i program data je uložen v operační paměti na určité adrese (viz von Neumannova koncepce počítače)
- programovací jazyky schovávají konkrétní adresy proměnných za jejich název (programátora nezajímá, že obsah proměnné X je uložen např. na 0x256644AC)
- na úrovni instrukcí samozřejmě konkrétní adresy vidíme
- to že je operand uložen v paměti prozradí hranaté závorky [ xxxx ]
- adresovat lze:
  - přímo hodnotou adresy [0x12345678]
  - pomocí hodnoty uložené v nějakém registru [edi] nebo [eax] apod.
  - složitěji: [0x123456789+edi+eax\*6]

# Instrukce JAS

## (jazyka symbolických adres)

- odpovídá jedné (strojové) instrukci procesoru
- má nula nebo více operandů
- operandy zapisujeme za instrukci
- formát instrukce
  - **symb. název instr.** *nápověda* operandy
- *Nápověda* napovídá velikost dat (byte, word, dword) se kterými se pracuje
- operandy jsme si už představili

# Skupiny instrukcí

- Instrukce MOV
- Aritmetické instrukce (ADD, SUB, ADC, SBB, MUL, DIV, NEG)
- Instrukce porovnání (CMP, TEST)
- Instrukce pracující se zásobníkem PUSH, POP, CALL, RET
- Instrukce měnící běh programu (skoky)
  - JMP
  - JZ, JNZ ..
- Logické instrukce (AND OR XOR NOT)
- Instrukce posuvů a rotací (SHR SHL ROL ROR RCR RCL)