Vektorové operace

SIMD paralelismus (Single Instruction Multiple Data)

- aplikace → lineární algebra → zpracování obrazu a zvuky, machine learning
  - → blokové zpracování dat
  - → tenkýž alg. pro různé vstupy

1996 Intel MMX (multimedia extensions)
- 8 64-bit. vektorových registrů MM0 – MM7 — sdílené s FPU — přepínání – pomalé!
- integerová aritmetika (i saturační)
  - add, sub, mul

1998 3DNow + floatové vektory (2011 zrušeno)

Intel: SSE → SSE2 → SSE3 → SSSE3 → SSE4.1, 4.2 ...

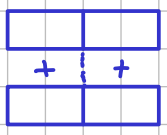
SSE = Streaming SIMD Extensions

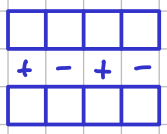
- registry XMM0 – XMM7 (v 64-bit. módu – XMM15)
  128-bit. 16×8, 8×16, 4×32, 2×64
  (4×32 = float; 16×8, 8×16, 4×32, 2×64 = int.)

floatové instrukce

ADDPD src, dest
- packed
- double

ADDSUBPS
- single





CMPEQPD – vyrobí bit. masku
- =

MAXPD

HADDPS
- horizontal

integerové instrukce

PADDW
- packed
- word

PADDSW
- saturate

PAND

PAVGB
- average

PMINB, PMAXB

PCMPEQB

přesuny

MOVAPD
- double
- packed
- aligned

MOVDQA
- aligned
- double quad

MOVUPD
- unaligned

stringy

S	32	float
D	64	float
B	8	int
W	16	int
D	32	int
Q	64	int
DQ	128	int

permutace

PSHUFD
- shuffle
- řízeno 8bit. tabulkou

PBLENDW podmíněné přiřazení
- 8bit. literál
- xmm0
- pořadí obecně

PMASKMOV ... EDI/RDI

AVX (Advanced Vector eXtensions)

- 256-bit. registry YMM0-15 (rozšíření XMMm)
- 3-operandové instrukce
- jen floatové

AVX2 – navíc integery

- nevyžaduje alignment (unaligned pomalejší)
- gather (vektorové indexování)

problémy:
- některé ops se mohou počítat nadvakrát
- vektorové patří, může překrývat CPU ↓ zpomalení hodin

AVX 512 – 512-bit. vektory

- 32 512-b. registrů ZMM0-31
- 8 opmask registrů K0–7 (K0: fixně 1…1)
  - módy: nevybrané ponechat / nulovat
  - bit. ops na Kn → mohou ovlivnit flags
  - CMP generuje opmasku
- moduly

programování (GCC)

autovektorizace

velikost vektoru v bytech 2^k !

GCC vektory

```
typedef int __attribute__((vector_size(16))) vec;
```

+, −, = po složkách

x[i]

x * 3 .. auto-extend

== != → masky (vektory)

__builtin_ia32_paddb(x,y)

Intelovské C

```
#include <emmintrin.h>
```

typy: __v4si — int, vektor, signed, #složek

funkce:

_mm_add_epi8(x,y)

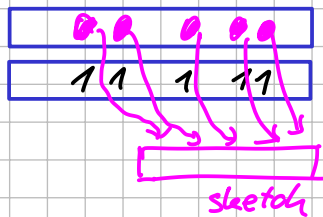
CPU

```
__builtin_cpu_init()
- cpu_is("core2")
- cpu_supports("avx2")
```

/proc/cpuinfo

Zvěřinec:

- bit. operace POPCNT …
- řetězcové porovnávání
- insert / extract field
- permutace
- 16-bit floaty
- AES, SHA-1, SHA-256
- násobení 2_2^k
- CRC32
- HW random generator
- PDEP / PEXT



- PREFETCH → kam? → L1 …
- netemporální přístup k paměti → pozor na koherenci → nutné bariéry