SIMD

Organización de Computadores

Felipe Garay

Universidad de Santiago

19 de noviembre de 2014

Temas a tratar

UdeSantiago de Chile

Introducción

- El pipeline paraliza la ejecución de distintas instrucciones.
- El cache mantiene los datos más utilizados más cerca del procesador.

Introducción

- El pipeline paraliza la ejecución de distintas instrucciones.
- El cache mantiene los datos más utilizados más cerca del procesador.
- ¿Y los datos?

Paralelizar los datos

- La idea es ahora paralelizar al nivel de datos.
- SIMD: Single Instruction Multiple Data

Figura: Escalar vs Vectorial

SIMD

- Instrucciones de ensamblador.
- Intrinsics para utilizar estas instrucciones en C.

Versiones

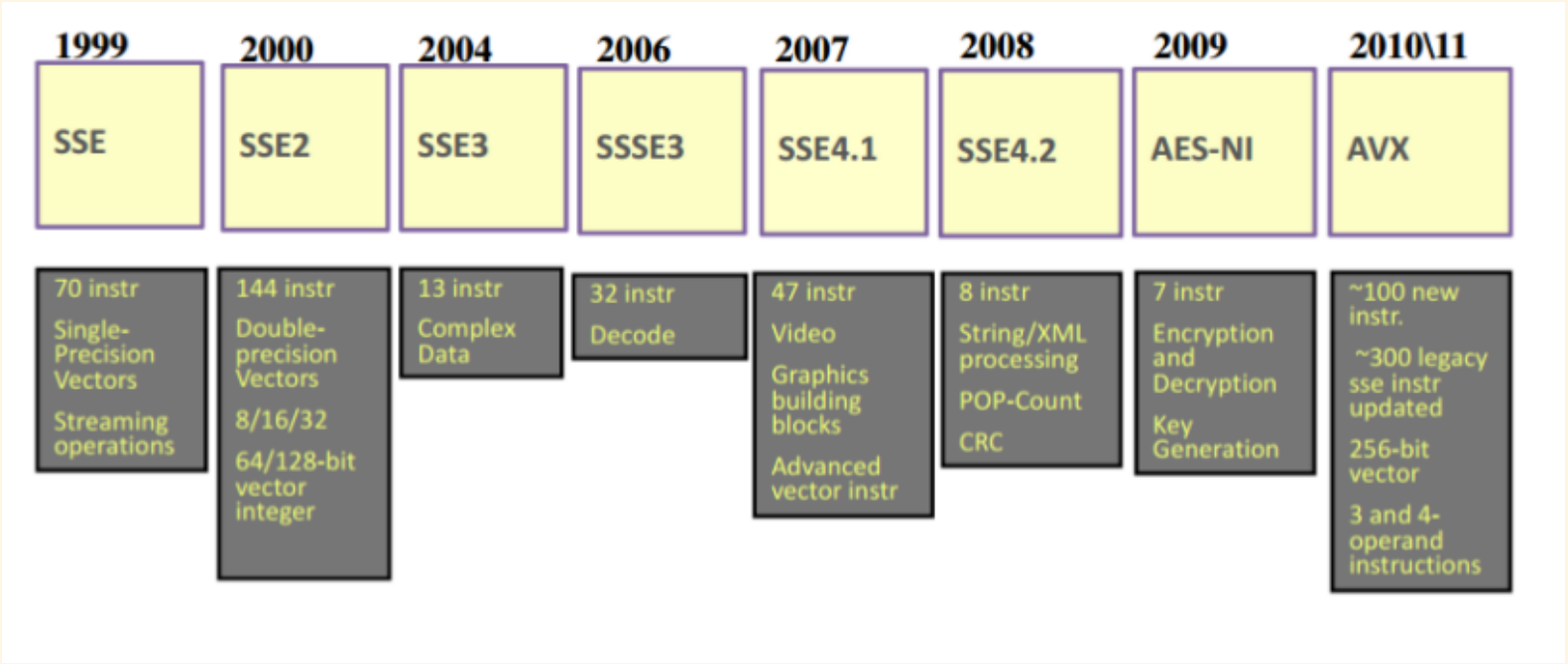

Figura: Versiones de SIMD en Intel

En C

- Existen headers que nos permiten utilizar SIMD en C.
- xmmintrin.h para SSE
- emmintrin.h para SSE 2
- pmmintrin.h para SSE 3
- Para compilar se debe utilizar
  gcc -std=c11 ejemplo.c -o ejemplo -msse -msse2 -msse3

Tipos de datos: Vectores

- __m64: Vector de 64 bits (8 enteros de 8 bits).
- __m128: Vector de 128 bits (16 enteros de 8 bits, 4 flotantes de precisión simple).
- Otros: Depende de la versión de SSE soportada por el procesador.

Cargando datos

```
float a[4] __attribute__((aligned(16)));

__mm128 = _mm_load_ps(a);
```

Tipos instrucciones

```
v = _mm_add_ps(a, b)
v = _mm_add_pi(a, b)
v = _mm_add_pd(a, b)
```

Instrucciones

```
v = _mm_add_ps(a, b)
v = _mm_sub_ps(a, b)
v = _mm_mul_ps(a, b)
v = _mm_div_ps(a, b)
v = _mm_sqrt_ps(a, b)
v = _mm_min_ps(a, b)
v = _mm_max_ps(a, b)
```