CS 110 Discussion 15

Programming with SIMD Intrinsics

Yanjie Song

School of Information Science and Technology

May 7, 2020

Table of Contents

Table of Contents

Introduction on Intrinsics

Definition

In computer software, in compiler theory, an **intrinsic function** (or **builtin function**) is a function (subroutine) available for use in a given programming language whose implementation is handled specially by the compiler.

Intrinsics in C/C++

Compilers for C and C++, of Microsoft, Intel, and the GNU Compiler Collection (GCC) implement intrinsics that map directly to the x86 single instruction, multiple data (SIMD) instructions (MMX, Streaming SIMD Extensions (SSE), SSE2, SSE3, SSSE3, SSE4).

x86 SIMD instruction set extensions

- MMX (1996, 64 bits)
- 3DNow! (1998)
- Streaming SIMD Extensions (SSE, 1999, 128 bits)
  - SSE2 (2001)
  - SSE3 (2004)
  - SSSE3 (2006)
  - SSE4 (2006)
- Advanced Vector eXtensions (AVX, 2008, 256 bits)
  - AVX2 (2013)
- F16C (2009)
- XOP (2009)
- FMA
  - FMA4 (2011)
  - FMA3 (2012)
- AVX-512 (2015, 512 bits)

SIMD extensions in other ISAs

There are SIMD instructions for other ISAs as well, e.g.

- NEON (ARM)
- MIPS-3D

Intel(R) Intrinsics Guide

Intel(R) Intrinsics Guide

Table of Contents

Enable SIMD Intrinsics in GCC

One way is to enable the flags of corresponding extension set. For example, add `-mavx2` to enable AVX2.
SSE and SSE2 are enabled by default on x86-64 machines.

Enable SIMD Intrinsics in GCC

Another approach is to select the microarchitecture of your CPU. For example, your CPU belongs to Skylake, then add `-march=skylake` to GCC.

For more options, please refer to x86 Options.

Optimization level and Vectorization

There are three levels of optimizations in GCC. The compiler will only do automatic vectorization on the third level.
Different to GCC, Intel's compiler (`icc`) will do automatic vectorization on the second level.

Table of Contents

AVX-512

Even though AVX-512 seems to be very powerful, only Intel High-end Desktop and Server CPU supports it.
How can I test my program if I use AVX-512?

Intel(R) SDE

Intel(R) Software Development Emulator provides software emulation for instruction like AVX-512.
Please refer to the website for its usage.
Since it is software emulation, the speed is rather slow.

Table of Contents

Horizontal sum

The packed data in the big register is called a vector. The horizontal sum of a vector is the sum of elements in that vector.
For example, suppose 8 packed integers are stored in a, then the horizontal sum of a is $\sum_{i=0}^{7} a_i$.

Horizontal sum in AVX-512

Compilers have provided helper functions that can do reduce operations on the vector.

For example, `_mm512_reduce_add_epi32` returns the sum of 16 integers in the vector.

This is just a helper function, not actual instructions.

More general approach

There is a more general approach:

1. Extract the high half and the low half.
2. Add them up.
3. Repeat until get the result.

Take AVX as an example:

```
int hsum_8x32(__m256i v)
{
    __m128i sum128 = _mm_add_epi32(
        _mm256_castsi256_si128(v),
        _mm256_extracti128_si256(v, 1));
    return hsum_epi32_avx(sum128);
}
```

Inside 128-bit vector

Shuffle inside 128-bit vector is more efficient, so we usually stop at this stage.

```
int hsum_epi32_avx(__m128i x)
{
    __m128i hi64  = _mm_unpackhi_epi64(x, x);
    __m128i sum64 = _mm_add_epi32(hi64, x);
    __m128i hi32  = _mm_shuffle_epi32(sum64, 177);
    __m128i sum32 = _mm_add_epi32(sum64, hi32);
    return _mm_cvtsi128_si32(sum32);
}
```

$$(10110001)_2 = (177)_{10}$$

Reference

The code is mainly referenced from Peter Cordes' answer from Stack Overflow.