

SIMD

Modern processors have capabilities for SIMD (single instruction multiple data) like SSE or AVX instruction sets on x64 or NEON on ARM architecture. Now, in 2022, C/C++ compilers still have problems at automatic vectorization to generate these vector instructions. Assembler or compiler intrinsic functions can be used to enforce the usage ..

CPU architectures/infos

- <https://en.wikichip.org/wiki/x86/extensions>
- <https://en.wikichip.org/wiki/arm/versions>
- https://en.wikichip.org/wiki/arm_holdings#Microarchitectures
- https://en.wikipedia.org/wiki/Raspberry_Pi#Specifications
 - https://en.wikipedia.org/wiki/Comparison_of_ARMv8-A_processors
 - does contain the L1/L2/L3 cache sizes
 - Raspberry Pi 4 has Broadcom Chip BCM2711

Libraries

- <https://www.intel.com/content/www/us/en/docs/intrinsics-guide/index.html>
- <https://github.com/VcDevel/Vc>
- <https://github.com/hayguen/MIPP>
 - it's origin <https://github.com/aff3ct/MIPP>
- SSE/AVX to Neon
 - <https://github.com/DLTcollab/sse2neon>
 - <https://github.com/kunpengcompute/AvxToNeon>
- Neon for x86/SSE
 - https://github.com/intel/ARM_NEON_2_x86_SSE
- SIMD Everywhere
 - <https://simd-everywhere.github.io/blog/>
 - <https://github.com/simd-everywhere/simde>
 - <https://github.com/jpcima/simde>

From:

<https://codingspirit.de/dokuwiki/> - **coding spirit**

Permanent link:

<https://codingspirit.de/dokuwiki/doku.php?id=development:simd>

Last update: **2022/12/24**

