

CSC 411

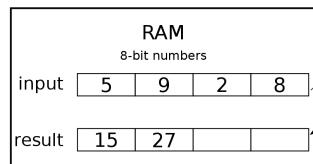
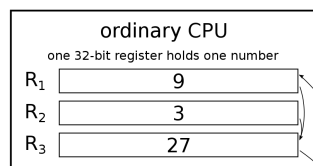
Computer Organization (Fall 2024)
Lecture 20: SIMD instructions

Prof. Marco Alvarez, University of Rhode Island

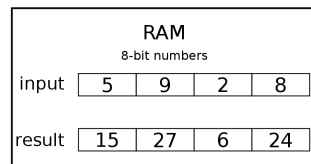
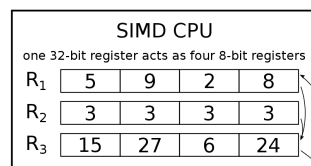
SIMD instructions

- ▶ Graphics and media processing operate on values of 8-bit and 16-bit lengths
 - can use a 128-bit adder (with partitioned carry chain) and perform operations in **parallel** — e.g. sixteen 8-bit operations, eight 16-bit operations, or four 32-bit operations
- ▶ SIMD (**single-instruction, multiple-data**)
 - a.k.a. data level parallelism, vector parallelism

Tripling four 8-bit numbers



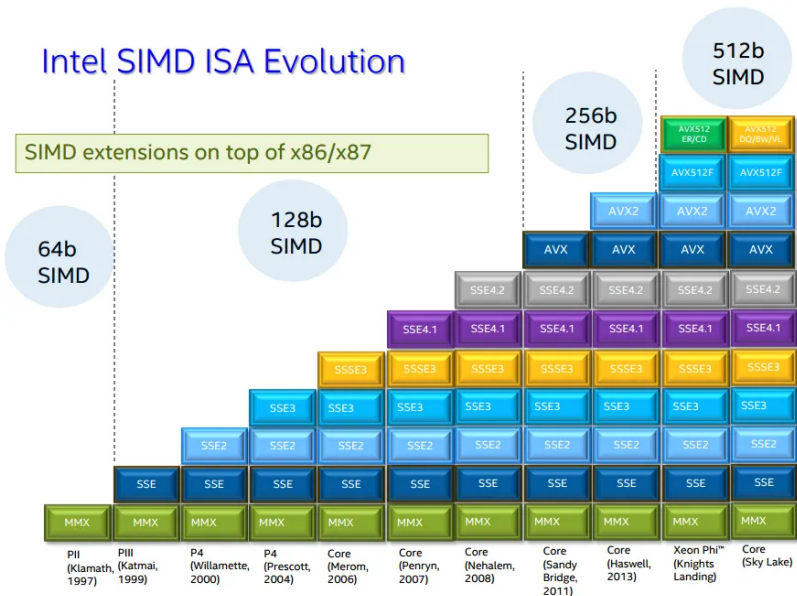
Operation count:
4 loads, 4 multiplies, and 4 saves



Operation count:
1 load, 1 multiply, and 1 save

https://en.wikipedia.org/wiki/Single_instruction,_multiple_data

Intel SIMD ISA Evolution



<https://en.algorithmica.org/hpc/simd/>

How to check for instruction set extensions?

- **Identify** your intel® Processor and note the processor number.
- **Go to** the [product specification page](#) and **enter** the number of the Intel Processor in the search box.
- **Look** in the *Advanced Technologies* section and **look for** *Instruction Set Extensions*

EXample:

Product Specifications

Products Home

09-9900K

20

Specifications		
Essentials		
CPU Specifications		
Supplemental Information	Intel® Optane™ Memory Supported ?	Yes
Memory Specifications	Intel® Turbo Boost Technology ?	2.0
Processor Graphics	Intel® Hyper-Threading Technology ?	Yes
Expansion Options	Intel® Transactional Synchronization Extensions ?	Yes
Package Specifications	Intel® 64 ?	Yes
Advanced Technologies	Instruction Set ?	64-bit
Security & Reliability	Instruction Set Extensions ?	Intel® SSE4.1, Intel® SSE4.2, Intel® AVX2
Ordering and Compliance	Idle States ?	Yes
Compatible Products	Enhanced Intel SpeedStep® Technology ?	Yes

<https://www.intel.com/content/www/us/en/support/articles/000090473/processors/intel-core-processors.html>

C code

```
#include <stdio.h>
#include <immintrin.h>

int main() {
    if (__builtin_cpu_is("intel")) printf("Intel CPU\n");
    if (__builtin_cpu_is("amd")) printf("AMD CPU\n");

    if (__builtin_cpu_supports("avx512f")) printf("AVX-512 supported\n");
    if (__builtin_cpu_supports("avx2")) printf("AVX2 supported\n");
    if (__builtin_cpu_supports("avx")) printf("AVX supported\n");
    if (__builtin_cpu_supports("sse2")) printf("SSE2 supported\n");
    if (__builtin_cpu_supports("sse")) printf("SSE supported\n");

    return 0;
}
```