

MICROARCHITECTURE CHEAT SHEET

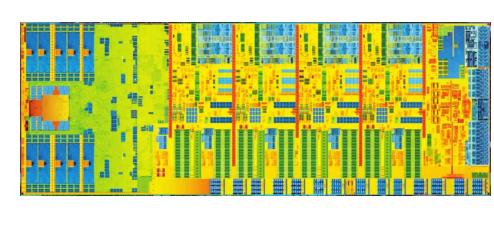
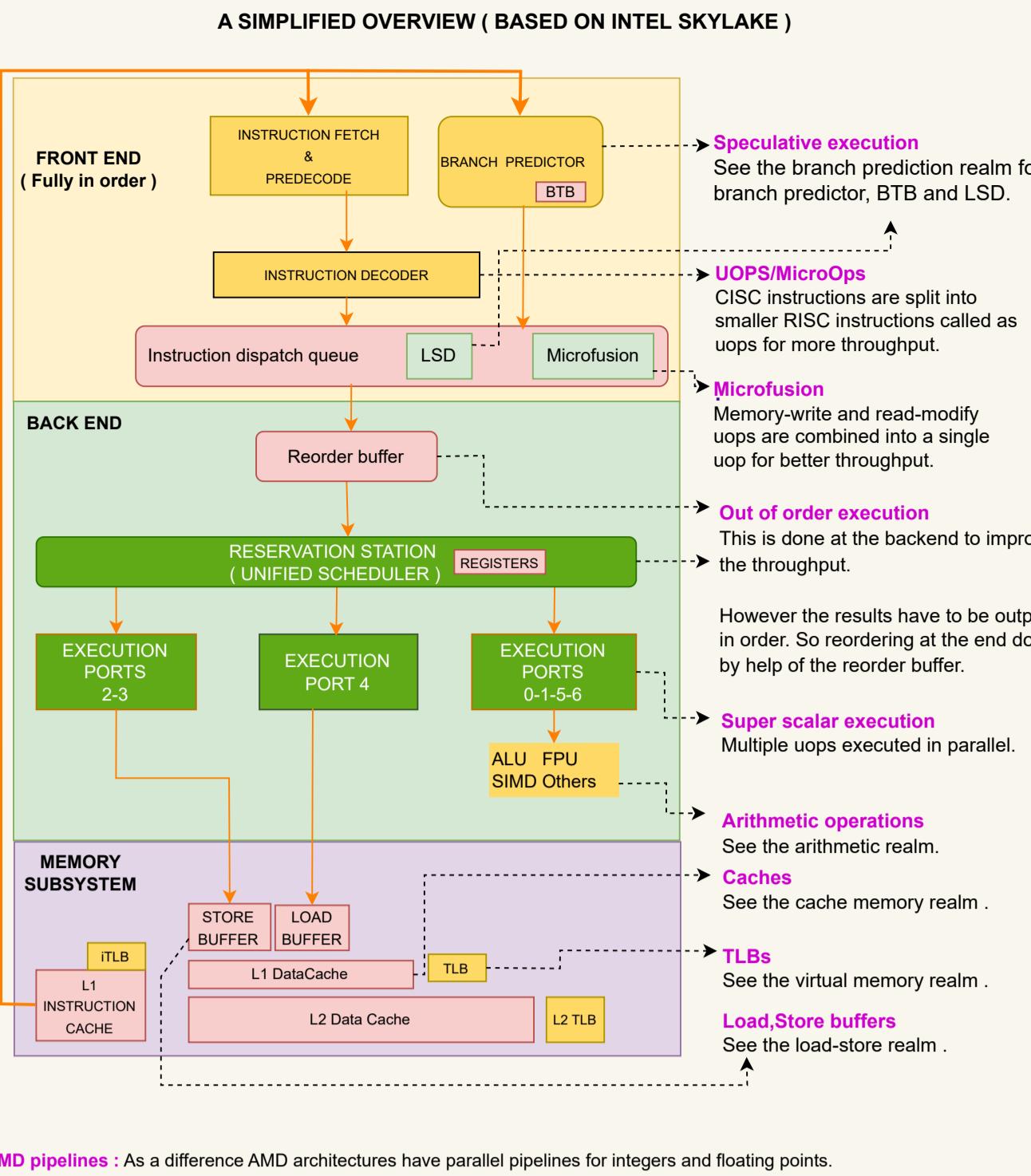
X86 CPUs & Performance

LAST UPDATE DATE : 02 JAN 2026

FOR LATEST VERSION: [www.github.com/akinh/microarchitecture-cheatsheet](https://github.com/akinh/microarchitecture-cheatsheet)

AUTHOR: AKIN OCAL skin_ocal@hotmail.com

Pipeline Realm : Inside An Individual Core



Copyright of Intel

Pipeline Parallelism & Performance

Pipeline diagrams: The diagrams below are outputs from an online microarchitecture analysis tool UCA, and they represent parallel execution through cycles.

Rows are multiple instructions being executed at the same time.

Columns display how instruction state changes through cycles.

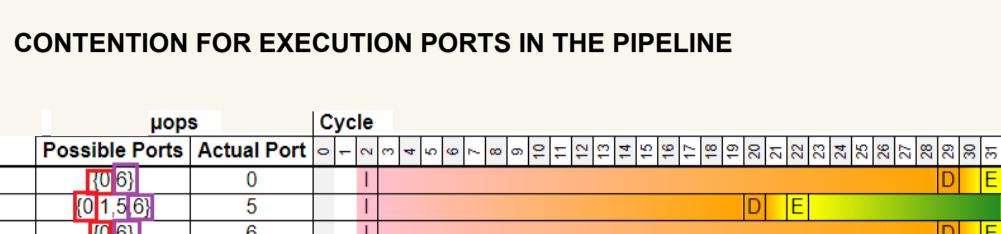
IPC: As for pipeline performance, typically IPC is used. It stands for "instructions per cycle".

A higher IPC usually means a better throughput.

You can measure IPC with perf: <https://perf.wiki.kernel.org/index.php/Tutorial>

Rate of retired instructions: Apart from IPC, number of retired instructions should be checked. Retired instructions are not committed instructions which were wrongly speculated. On the other hand executed instructions are the ones which were finalized. Therefore a high rate of retired instructions indicates low branch prediction rate.

Instruction latency states in UCA diagrams



In the example above, all instructions are working on different registers, but SHR, ADD, DEC instructions are competing for ports 0 & 6. SHR and DEC are getting executed after ADD instruction.

Also notice that there is longer time between E(executed) and R(referred) states of instruction ADD as retirement has to be done in-order whereas execution is out-of-order.

Reference : Dennis Bakhvalov's article

INSTRUCTION STALLS DUE TO DATA DEPENDENCY



In the example above, there are 2 dependency chains, each marked with a different colour. In the first red coloured one, 2 instructions are competing for RAX register and notice that the second instruction gets executed after the first one. And the same applies to the 2nd purple pair.

Reference : Dennis Bakhvalov's article

ARITHMETIC REALM

X86 user IEEE 754: IEEE 754 standard for floating point numbers. A 32 bit floating point consists of 3 parts

In the memory layout, below you can see all bits of 32 bit IEEE 754 visualized : <https://bartsch.github.io/ieee754-visualisation/>

Floating Points: A floating point's value is calculated as: $sign \times mantissa \times 2^{\text{exponent}}$

IEEE754 also defines **denormal numbers**. They are very small / near zero numbers.

As floating point numbers are denormalized, some special cases are needed to avoid underflow of a float but a float b = 0. Without denormal support the code to right handle this would be much more complex. Reference : Bruce Dawson's article

X86 EXTENSIONS: SIMD

SSE (Streaming SIMD Extensions) is the most popular level that provides SIMD parallelization. SIMD stands for single instruction multiple data. SIMD instructions use wider registers to execute more work in a single go.

X86 EXTENSIONS : SIMD DETAILS

The most recent SIMD instruction sets for Intel CPUs are :

AVX: Up to 256 bits

AVX2: Up to 256 bits

AVX512: Up to 512 bits

AVX10: Up to 512 bits

As for programming, there are also wider data types. The data type diagrams below are for 128 bit operations:

m128: 4 x 32 bit floating points

m128d: 8 x 16 bit doubles

m256: 16 x 16 bit ints

m256l: 32 x 16 bit longs

m512: 64 x 16 bit longs

m512l: 128 x 16 bit longs

m512ll: 256 x 16 bit longs

m512lll: 512 x 16 bit longs

m512llll: 1024 x 16 bit longs

m512lllll: 2048 x 16 bit longs

m512llllll: 4096 x 16 bit longs

m512lllllll: 8192 x 16 bit longs

m512llllllll: 16384 x 16 bit longs

m512lllllllll: 32768 x 16 bit longs

m512lllllllll: 65536 x 16 bit longs

m512llllllllll: 131072 x 16 bit longs

m512llllllllll: 262144 x 16 bit longs

m512llllllllll: 524288 x 16 bit longs

m512llllllllll: 1048576 x 16 bit longs

m512llllllllll: 2097152 x 16 bit longs

m512llllllllll: 4194304 x 16 bit longs

m512llllllllll: 8388608 x 16 bit longs

m512llllllllll: 16777216 x 16 bit longs

m512llllllllll: 33554432 x 16 bit longs

m512llllllllll: 67108864 x 16 bit longs

m512llllllllll: 134217728 x 16 bit longs

m512llllllllll: 268435456 x 16 bit longs

m512llllllllll: 536870912 x 16 bit longs

m512llllllllll: 1073741824 x 16 bit longs

m512llllllllll: 2147483648 x 16 bit longs

m512llllllllll: 4294967296 x 16 bit longs

m512llllllllll: 8589934592 x 16 bit longs

m512llllllllll: 17179869184 x 16 bit longs

m512llllllllll: 34359738368 x 16 bit longs

m512llllllllll: 68719476736 x 16 bit longs

m512llllllllll: 137438953472 x 16 bit longs

m512llllllllll: 274877906944 x 16 bit longs

m512llllllllll: 549755813888 x 16 bit longs

m512llllllllll: 1099511627776 x 16 bit longs

m512llllllllll: 2199023255552 x 16 bit longs

m512llllllllll: 4398046511104 x 16 bit longs

m512llllllllll: 8796093022208 x 16 bit longs

m512llllllllll: 17592186044416 x 16 bit longs

m512llllllllll: 35184372088832 x 16 bit longs

m512llllllllll: 70368744177664 x 16 bit longs

m512llllllllll: 140737488355328 x 16 bit longs

m512llllllllll: 281474976710656 x 16 bit longs

m512llllllllll: 562949953421312 x 16 bit longs

m512llllllllll: 1125899906842624 x 16 bit longs

m512llllllllll: 2251799813685248 x 16 bit longs

m512llllllllll: 4503599627370496 x 16 bit longs

m512llllllllll: 9007199254740992 x 16 bit longs

m512llllllllll: 18014398509481984 x 16 bit longs

m512llllllllll: 36028797018963968 x 16 bit longs

m512llllllllll: 72057594037927936 x 16 bit longs

m512llllllllll: 144115188075855872 x 16 bit longs

m512llllllllll: 288230376151711744 x 16 bit longs

m512llllllllll: 576460752303423488 x 16 bit longs

m512llllllllll: 1152921504606846976 x 16 bit longs

m512llllllllll: 2305843009213693952 x 16 bit longs

m512llllllllll: 4611686018427387904 x 16 bit longs

m512llllllllll: 9223372036854775808 x 16 bit longs

m512llllllllll: 18446744073709551616 x 16 bit longs

m512llllllllll: 36893488147419103232 x 16 bit longs

m512llllllllll: 73786976294838206464 x 16 bit longs

m512llllllllll: 147573952589676412928 x 16 bit longs

m512llllllllll: 295147905179352825856 x 16 bit longs

m512llllllllll: 590295810358705651712 x 16 bit longs

m512llllllllll: 118059162071741130344 x 16 bit longs

m512llllllllll: 236118324143482260688 x 16 bit longs

m512llllllllll: 472236648286964521376 x 16 bit longs

m512llllllllll: 944473296573929042752 x 16 bit longs

m512llllllllll: 1888946593147858085504 x 16 bit longs

m512llllllllll: 3777893186295716161008 x 16 bit longs

m512llllllllll: 7555786372591432322016 x 16 bit longs