

The single-core, multi-core and many-core CPU



Multi-core CPU from 2011

CPUs are now split up in 3 types, depending on the number of cores: single (1), multi (2-8) and many (10+).

I find it more important now to split up into these three types, as the types of problems to be solved by each is very different. Based on the problem-differences I'm even expecting that the number of cores between multi-core CPUs and many-core CPUs will grow.

Below are the three types of CPUs discussed and a small discussion on many-core processors we see around.

1 core CPUs

Single core CPUs can currently only be found on low-power solutions, but even there the minimum seems to be 2 (one for the task, one for the OS).

The main reason has been the frequency wall of 2006, which made it impossible to get to 10 MHz processors and up. The i7-2600 from 2011 from the above photo has a score of 1921 on the [Passmark single thread benchmark](#), where the fastest CPU, the 2017 Intel Core i7-7740X, gives 2652.

So we simply gave up on 1-core CPUs, but with [new materials that replace silicon](#) they could make a comeback.

2-8 core CPUs

For slightly over a decade 2 cores are quite standard. Intel with Core 2 Duo, AMD with Athlon 64 X2 in 2005/2006. ARM came years later with their first commercial dual-core Cortex-A9. While operating systems could schedule different tasks to different cores, software packages took years to catch up. Software could delay, because in CPUs there were still improvements possible by increasing the instructions per clock tick (longer pipelines and such). 4-core CPUs took off only a few years ago. 8-core CPUs are still a luxury CPU.

We can expect the core-count for desktops, laptops, tablets and smartphones to keep between 4 and 8 cores for the years to come.

10-64 core CPUs

Not HPC, but virtualization has been a driving force behind many-core CPUs. Virtualized CPUs need to handle many threads simultaneously.



For HPC this could be the one thing that breaks the chicken-egg problem for increased core-counts. Till now CPUs simply had a too high \$/GFLOPS to be competitive on massively parallel code. With AMD halving the prices of many-core processors with Epyc, CPUs can better compete with GPUs again on both single precision and double precision. It depends on what opportunities AMD, Intel and IBM see how this will evolve exactly.

We're expecting a steady growing market for [CPU code modernization](#).]

Manycore processors

A CPU is a processor, but a processor not always a CPU - this is especially true when the core-count goes up. GPUs, DSPs and special purpose processors already often have 100's or 1000's of cores. For example the low-power many-core processors we call [Grid-processors](#) - these perform very well on tasks like video encoding, signal processing, cryptography and neural networks. Or Google's TPU which has [65 thousand 8-bit integer multipliers](#).

A special mention is the LGA 3647 socketed X200f Xeon Phi. It's technically a coprocessor, but it has a CPU that can handle the booting and other "normal" CPU tasks. It has 6 channels to the system DDR4, theoretically reaching 115.2 GB/s. Still, this is not a CPU, but a CPU+accelerator in one package.

CPUs have taken several techniques from GPUs, like AVX and FMA. Expected is that CPU will keep adopting new techniques. The question is how close the gap will be in a few years.