NVIDIA:
Sm89: Ada (4090,80,70) -> cc: 8.9
Sm90: Hopper (H100,200) -> cc:9
Sm100: Blackwell datacenter (B200) -> cc:10
Sm120: Blackwell consumer (5090,80,70) -> cc:12

References:

https://docs.nvidia.com/cuda/parallel-thread-execution/#warp-level-matrix-instructions

https://news.ycombinator.com/item?id=45280592

https://docs.nvidia.com/cuda/cuda-c-programming-guide/#compute-capabilities

https://docs.nvidia.com/cutlass/media/docs/cpp/blackwell_functionality.html

Jarmusch, A., Graddon, N. and Chandrasekaran, S. (2025) “Dissecting the NVIDIA Blackwell Architecture with Microbenchmarks.” arXiv. Available at: https://doi.org/10.48550/arXiv.2507.10789.

Luo, W. et al. (2024) “Benchmarking and Dissecting the Nvidia Hopper GPU Architecture.” arXiv. Available at: https://doi.org/10.48550/arXiv.2402.13499.

Luo, W. et al. (2025) “Dissecting the NVIDIA Hopper Architecture through Microbenchmarking and Multiple Level Analysis.” arXiv. Available at: https://doi.org/10.48550/arXiv.2501.12084.

Abdelkhalik, H. *et al.* (2022) “Demystifying the Nvidia Ampere Architecture through Microbenchmarking and Instruction-level Analysis.” arXiv. Available at: https://doi.org/10.48550/arXiv.2208.11174.

Luhnen, T., Marschner, T. and Lal, S. (no date) “Benchmarking Thread Block Cluster - 2CTA MMA.”

AMD: To do
https://rocm.docs.amd.com/en/latest/compatibility/compatibility-matrix.html
https://rocm.docs.amd.com/en/latest/reference/gpu-arch-specs.html

https://rocm.blogs.amd.com/software-tools-optimization/matrix-cores-cdna/README.html -> Instruction of low bit matmul

One thing that I noticed is that in AMD, they called it matrix core instead of tensor core

https://github.com/ROCm/composable_kernel -> equivalent with Cutlass in NVIDIA

https://github.com/ROCm/aiter -> Seems like full of kernel, but I found that they also use some fp8 matmul here. Which imply AMD recent hardware has some fp8 matrix core (recall that matrix core is a tensor core in NVIDIA)

https://gpuopen.com/learn/wmma_on_rdna3/ -> iiuc, this WMMA is the same as WGEMM in NVIDIA

Features:
Warp, warp groups, and warp specialization.

TMA, distributed smem, tensor memory (TMEM).

Hardware supported mxfp8, FP8, mxfp4, nvfp4 v/s simulated.

Tools:
Triton and Gluon support for Warp Specialization
PyTorch and TLX for warp specialization.

JAX team which express that their Pallas was able to do it as well :
https://docs.jax.dev/en/latest/pallas/gpu/blackwell_matmul.html#warp-specialization

Some links:
https://github.com/facebookexperimental/triton/tree/tlx
https://pytorch.org/blog/fast-2-simplicial-attention-hardware-efficient-kernels-in-tlx/
https://github.com/triton-lang/triton/tree/main/python/tutorials/gluon

I know there's the CuTe DSL -> but no AMD support.

Well I am not sure too that AMD has TMA or not. Not sure as well if this TDA is TMA equivalent in AMD : https://github.com/triton-lang/triton/pull/8333