Compute terminology translation guide

CPU (with an x86 bias)	NVidia CUDA	Khronos (OpenCL, SYCL, Vulkan…)
SIMD lane	CUDA core	Processing Element
SIMD vector	Warp	Subgroup
Simultaneous Multi-Threading / Hyper-Threading	Hardware multithreading	*Not named, hidden inside PE/CU model*
Advanced Matrix eXtensions, Neural Processing Unit	Tensor core	*No standard name (only usable via Vulkan vendor extensions)*
No equivalent, ray tracing is not special	RTX core	*Not named, exposed via VK_KHR extensions*
Intel Data Streaming Accelerator	Tensor Memory Accelerator (kind of [1])	*Not named, exposed via OpenCL async copy API*
Core	Streaming Multiprocessor	Compute Unit
Core complex / Sub-NUMA cluster	Thread block cluster (kind of [2])	*Not exposed yet*
NUMA node	Device	Device
SIMD registers, with manual [3] spill to caches/RAM	Local memory, with automatic spill to global memory	Private memory, with automatic spill to global memory
L1/L2 cache, with automatic spill to lower caches/RAM	Shared memory	Local memory
No equivalent, all memory is writable	Constant memory	Constant memory
No equivalent, images are not special	Texture memory	*Not named, exposed via image API*
Cache hierarchy + RAM	Global memory	Global memory
SIMD lane	Threads within the same warp	Work-items within the same subgroup
SIMD vector	Warp	Subgroup
Thread	Threads within the same thread block	Work-items within the same work-group
No equivalent, all threads can synchronize with each other	Thread block	Work-group
No equivalent, all threads can synchronize with each other	Thread block cluster	*Not exposed yet*
No equivalent, threads are independent from each other	Grid	NDrange
No equivalent, can spawn threads anytime	Stream	Command queue
No equivalent, can spawn threads anytime	Graph	Command buffer *(only exposed in Vulkan)*
Buffer	Buffer	Buffer
No equivalent, images are not special	Texture	Image
No equivalent, images are not special	Texture	Sampler
RAM	Memory	Device memory
No equivalent, all useful memory is reachable	Unified memory	Shared memory
No equivalent, all useful memory is reachable	Host/pinned memory	Host memory

[1] Both TMA and DSA allow offloading memory copy work from the compute cores, but Intel DSA is more focused on device DMA and Nvidia TMA on VRAM-cache copies.
[2] On a hardware level, it's the same idea, but CUDA only allows some forms of synchronization when threads belong to the same thread block cluster.
[3] Speaking from the perspective of machine code here. Of course, the compiler of most programming langages will automate it for you.