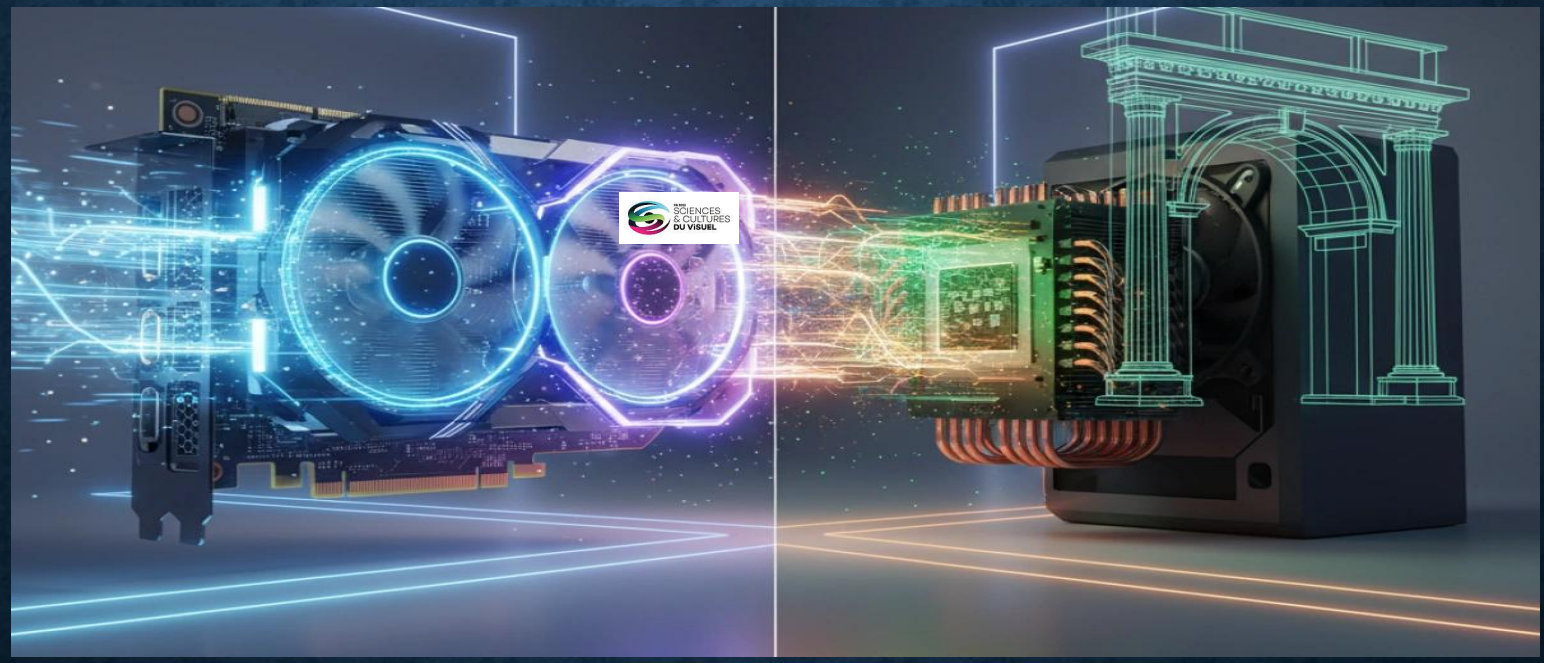
Université de Lille

FR 2053 SCIENCES & CULTURES DU VISUEL

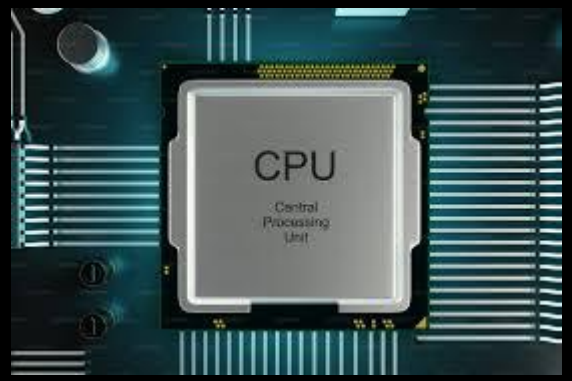
POURQUOI LES GPU DOMINENT LE CALCUL HAUTE PERFORMANCE MODERNE



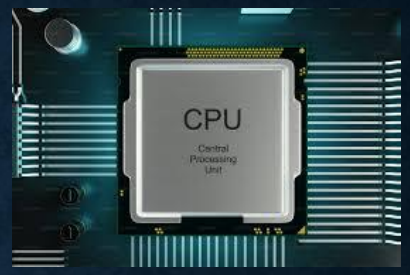
Audain Desrosiers, PhD
IGR expert en calcul scientifique

PLAN

- ❖ CPU vs GPU
- ❖ Cluster GPU
- ❖ Architecture CPU (Intel, AMD, Apple)
- ❖ GPU (Volta, Turing)
- ❖ Architecture GPU (Ada 6000)
- ❖Formule générale FLOPS d'un GPU
- ❖Avantage et désavantage des GPU
- ❖ Modèle entraîné sur GPU et CPU
- ❖Modèles impossibles à entraîner sur CPU

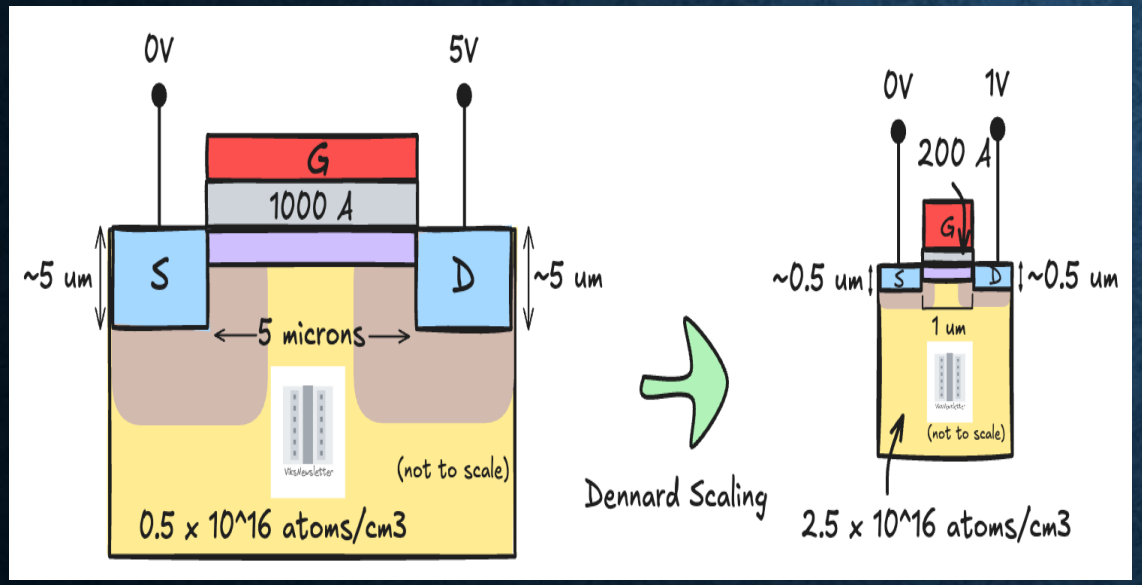


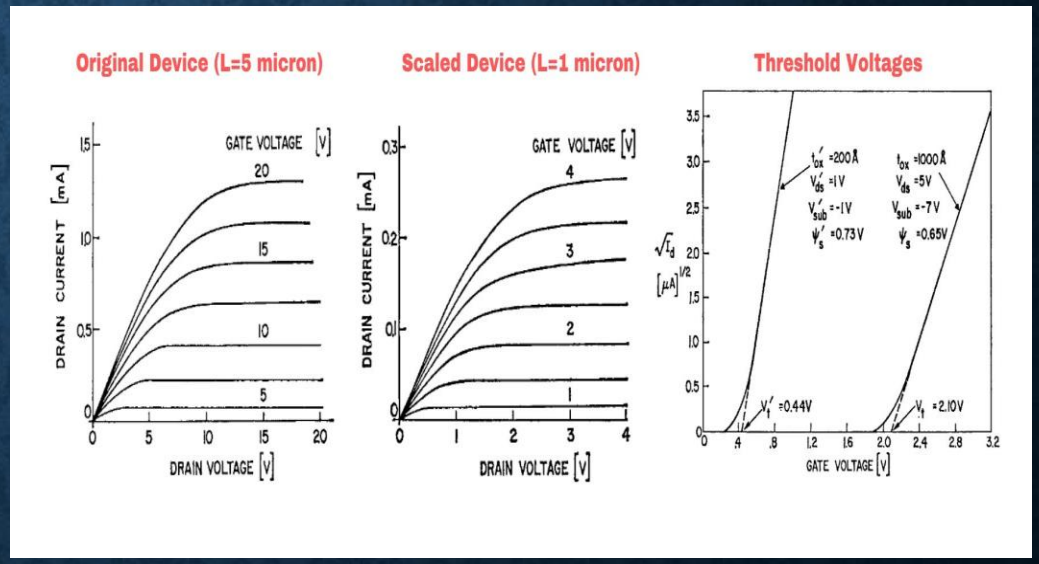
CPU VS GPU



➢1 - La loi de Dennard scaling

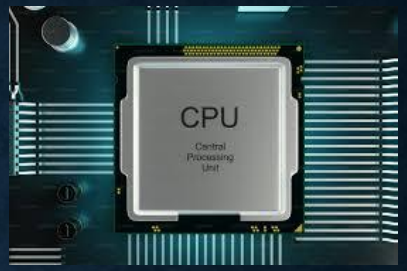
(Robert H. Dennard et al.1974, IBM Journal of Research and Development).



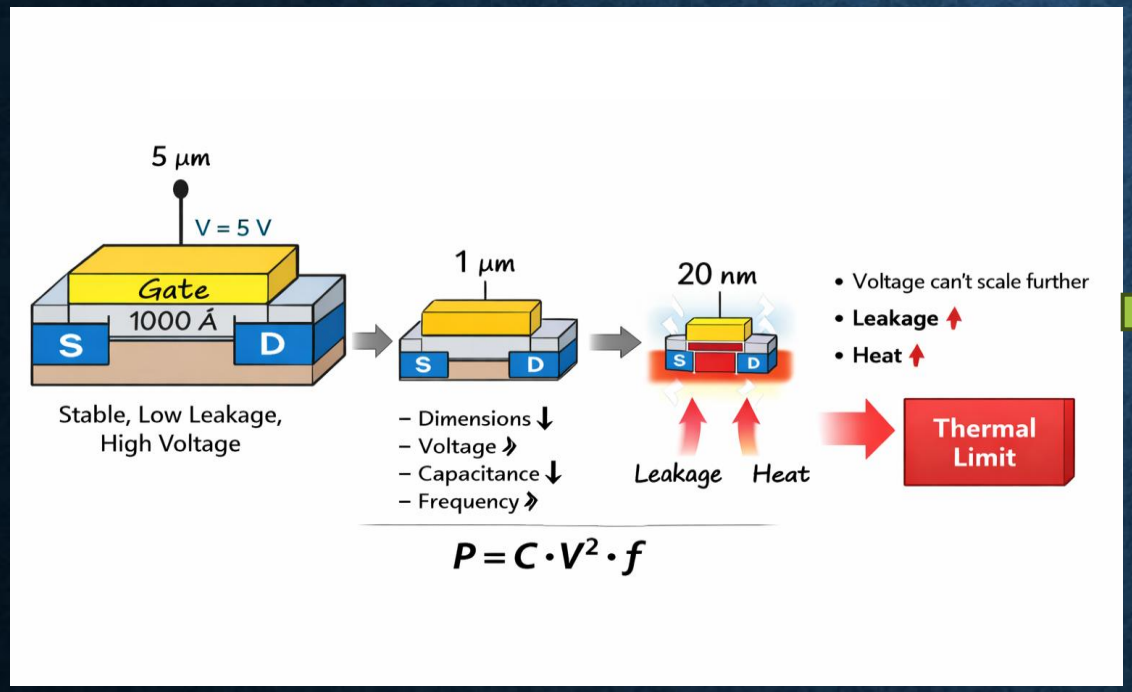


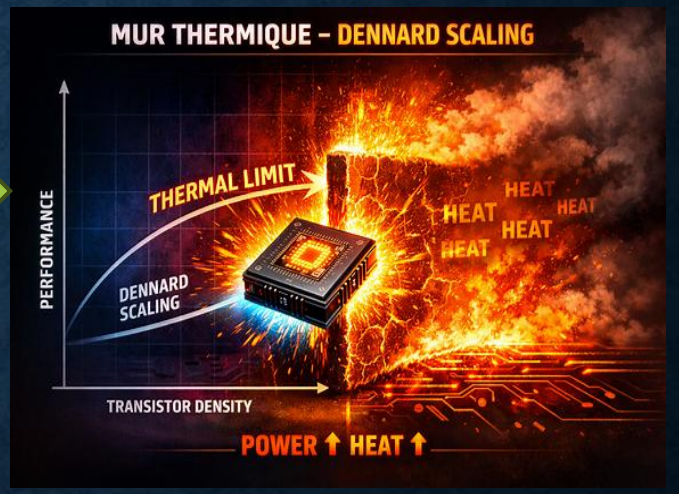
- Pendant près de 30 ans, l'industrie a appliqué le Dennard Scaling en réduisant la taille des transistors afin d'augmenter la fréquence des CPU à chaque génération sans augmenter la consommation électrique.

CPU VS GPU



- **Effondrement de la loi de Dennard scaling (2005).**





Le mur thermique correspond au point où l'augmentation de fréquence provoque une hausse de puissance dissipée plus rapide que la capacité de refroidissement. Au-delà de ~4 à 5 GHz, la densité thermique devient trop élevée. C'est pourquoi les CPU stagnent autour de 3–4 GHz (base) et ~5 GHz en boost.

CPU VS GPU

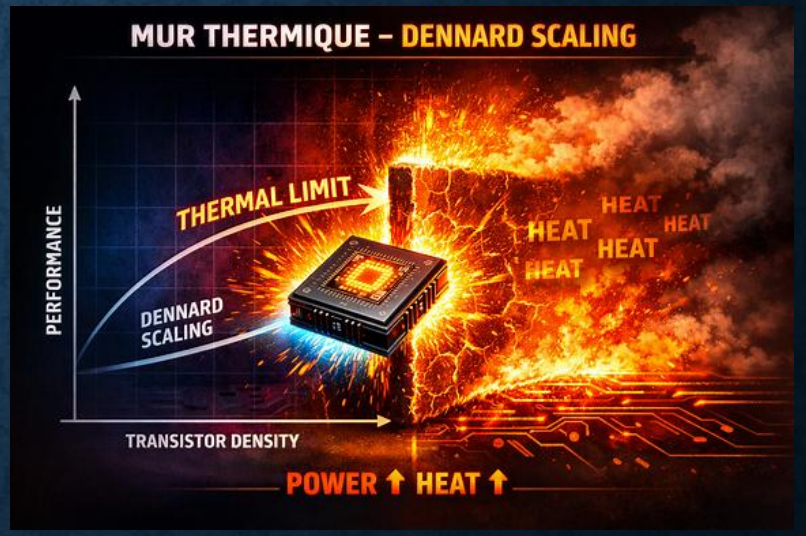
○Effondrement Dennard scaling (2005)

✓ 1 - Fuites électriques augmentent

✓2 - La tension ne peut plus être réduite

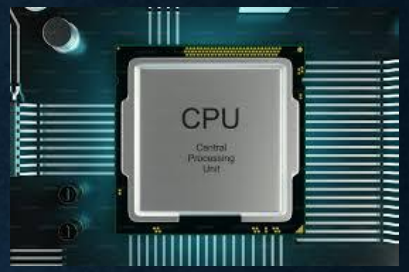
✓3 - la consommation par transistor ne baisse plus

Plus de transistors = plus de chaleur



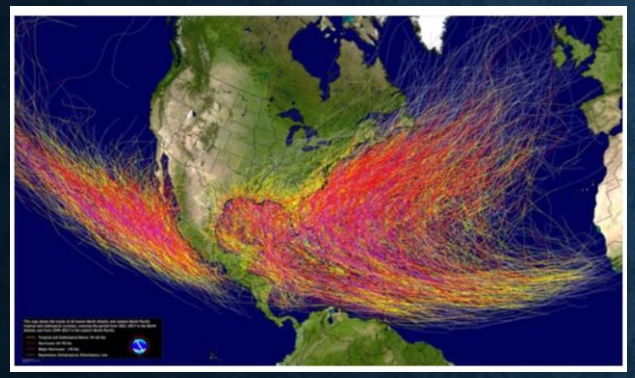
❖**Conséquence :** impossible de continuer à augmenter fréquence comme avant.

CPU VS GPU



➢2 - Les besoins explosent :

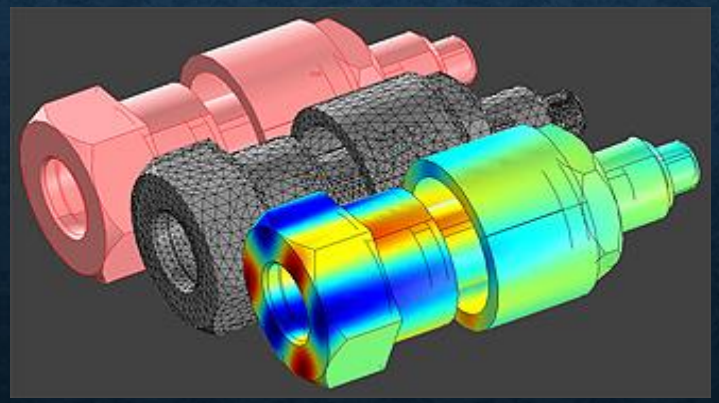
Climat



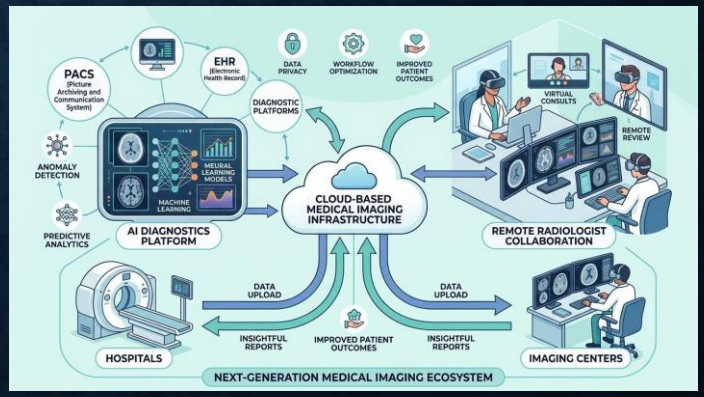
IA

Physique

Simulation scientifique



Imagerie médicale



CPU VS GPU

✓Excellent pour exécuter des tâches séquentielles.

✓Excellent gérer des instructions complexes.

✓Excellent pour minimiser la latence (répondre vite à une tâche unique).

✓Faire du multitâche (OS, applications, interruptions…).

➢ **Excellent dans les tâches où chaque étape dépend de la précédente, faible latence, tâches complexes, peu de cœurs.**

CPU VS GPU

- **Références des images :**
- How Dennard Scaling Allowed Transistors to Shrink
- https://www.postdicom.com/fr/blog/ai-in-radiology-medical-imaging
- https://www.simtecsolution.fr/fr/offre-de-service/sous-traitance-de-calculs-developpement-de-modeles
- https://www.climato-realistes.fr/les-oscillations-oceaniques-ont-une-influence-majeure-sur-le-developpement-du-climat-et-la-formation-des-cyclones-tropicaux/
- https://www.radiofrance.fr/franceinter/podcasts/la-terre-au-carre/la-terre-au-carre-du-mardi-30-septembre-2025-5372538
- https://liora.io/ia-generative-tout-savoir

CPU VS GPU

L’arrêt du Dennard Scaling et le mur thermique ont mis fin à la croissance des CPU. La solution a été d’exploiter une architecture différente : les GPU, capables de multiplier les unités de calcul plutôt que la fréquence.

CPU VS GPU

- Les GPU ont été initialement conçus pour le rendu graphique : calcul des pixels, textures, images et scènes 3D. On en trouve chez Intel, AMD, Apple et **NVIDIA**.

- En 2006, **NVIDIA** révolutionne le calcul parallèle avec le concept de GPGPU (General-Purpose GPU). Cette approche permet d'utiliser la carte graphique pour des calculs généraux grâce à l'API CUDA, qui offre la possibilité de programmer directement en langage C sur le GPU.

- L'architecture **NVIDIA** repose sur un parallélisme massif : chaque GPU regroupe des centaines à des milliers de cœurs simples, organisés en SM, capables d'exécuter des milliers de threads en parallèle.

CPU VS GPU

- ✓Excellent pour exécuter la même opération sur beaucoup de données (SIMD).
- ✓Excellent pour maximiser le débit (Floating-Point Operations Per Second: FLOPS).
- ✓Excellent pour paralléliser massivement avec des milliers de cœurs ou de threads.
- ✓Excellent pout traiter des matrices, vecteurs, pixels…

- ➢Excellent dans les tâches parallèles, indépendantes, répétitives, massives. Haut débit, milliers de cœurs.

CLUSTER DE CALCUL GPU

- Plusieurs nœuds de calcul (serveurs physiques)
- Chaque nœud contient plusieurs GPU (4, 8, 16…)
- Tous les nœuds sont reliés par un réseau très rapide (InfiniBand, NVLink, Ethernet 200 400 Gb/s)

POURQUOI UN CLUSTER DE CALCUL GPU

❑ Pour traiter des modèles dépassant la capacité d'une seule machine, ex:

✓Grands modèles d'IA (LLM, vision, multimodal)

✓Simulations climatiques

✓Dynamique des fluides (CFD)

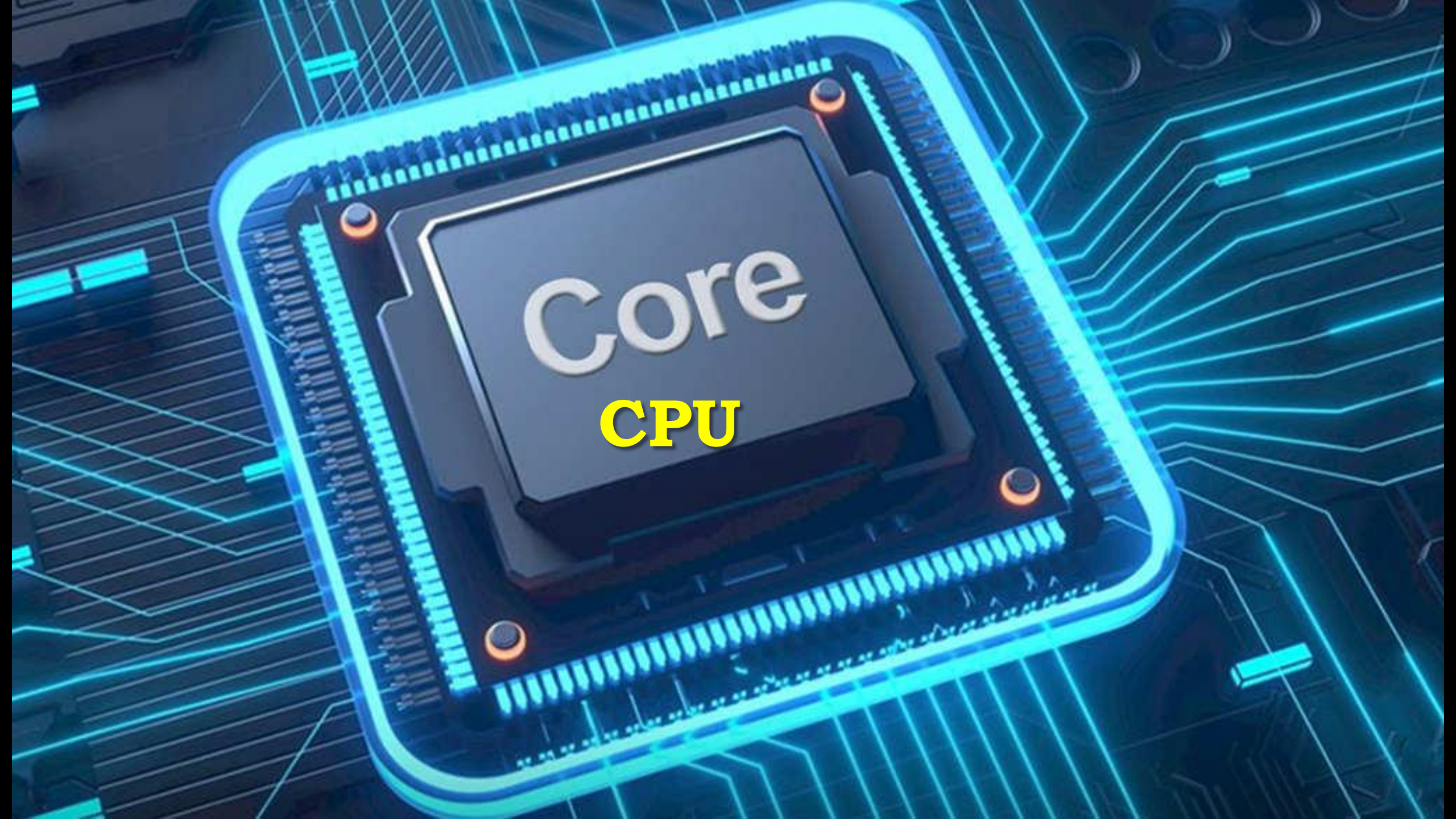
✓Chimie quantique

Un entraînement IA qui prendrait 3 mois sur 1 GPU peut prendre : 3 jours sur 100 GPU et quelques heures sur 1000 GPU,

(Image Source: NVIDIA)

Présentation d’un cluster GPU Blackwell–Grace combine les GPU NVIDIA Blackwell (B200/B100) avec les CPU Grace via NVLink-C2C. Cette architecture fournit une cohérence mémoire unifiée, une bande passante de plusieurs To/s et une puissance de calcul adapté aux modèles IA de plusieurs centaines de milliards de paramètres.

(Image Source: NVIDIA)


Core
CPU

CPU (INTEGRATED ELECTRONICS, INTEL)

CPU-Z: https://www.cpuid.com/softwares/cpu-z.html

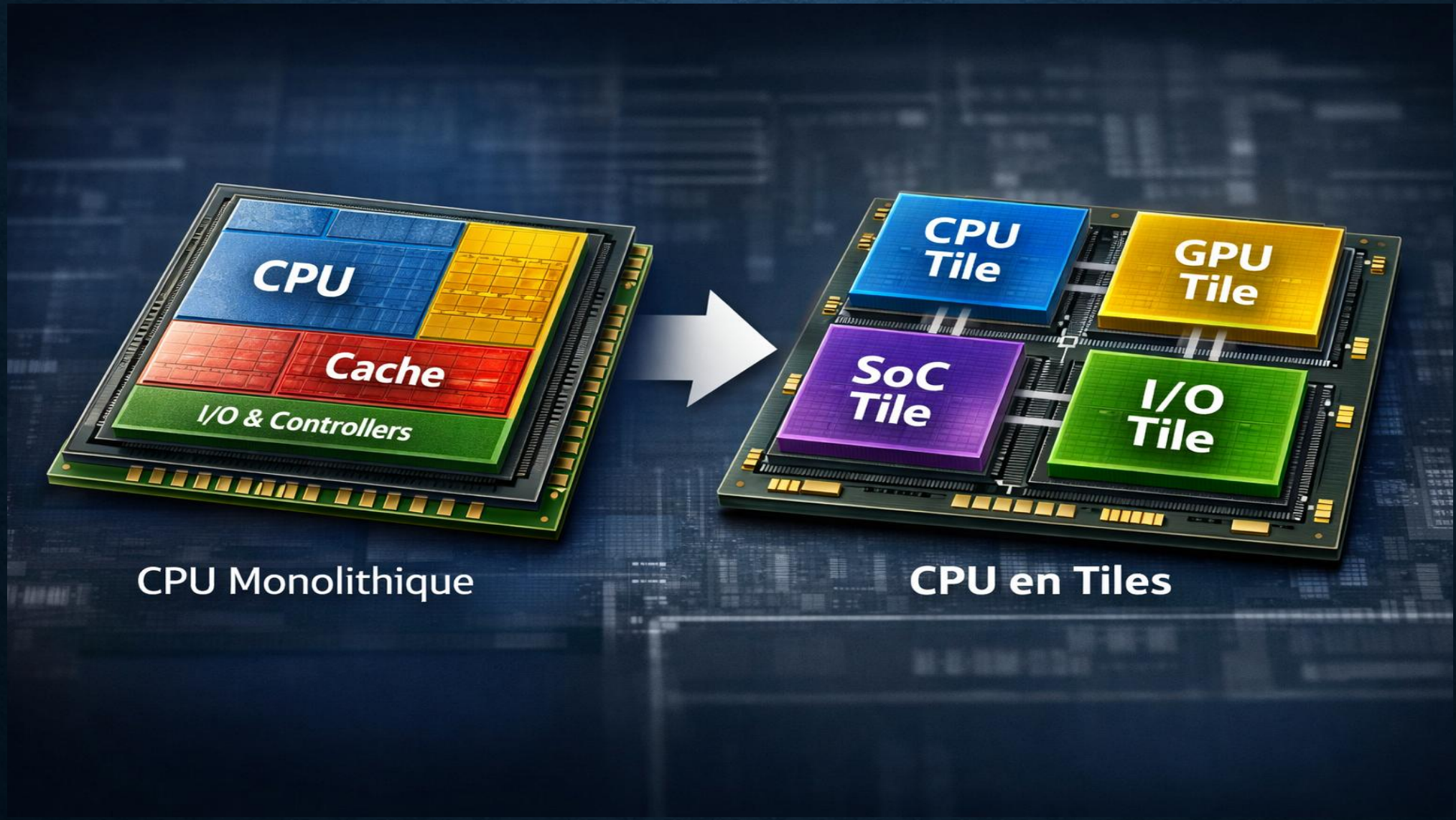
CPU-Z permet d'identifier précisément le processeur : modèle, architecture, nombre de cœurs, fréquences et caractéristiques techniques.

Ici, on va uniquement parler des cœurs du processeur, sans entrer dans les détails des caches, GPU (Intel, AMD, Apple). L'idée est simplement d'expliquer rapidement ce qu'est un cœur CPU.

INTEL : DU MONOLITHIQUE AUX TILES

- ✓ Avant 2021, les CPU Intel étaient monolithiques : un seul die regroupait cœurs, cache, iGPU et contrôleurs. Cette approche offrait simplicité et faible latence, mais atteignait ses limites thermiques.
- ✓ Alder Lake (2021) introduit l'hybridation P-cores + E-cores, première étape vers la modularité.
- ✓ Depuis Meteor Lake (2023), Intel adopte une architecture en tiles : CPU, GPU, SoC et IO séparés.
- ✓ Les tiles améliorent l'efficacité, la modularité et la fabrication (multi-foundries).
- ✓ Les générations Arrow Lake et Panther Lake poursuivent cette transition chiplet.

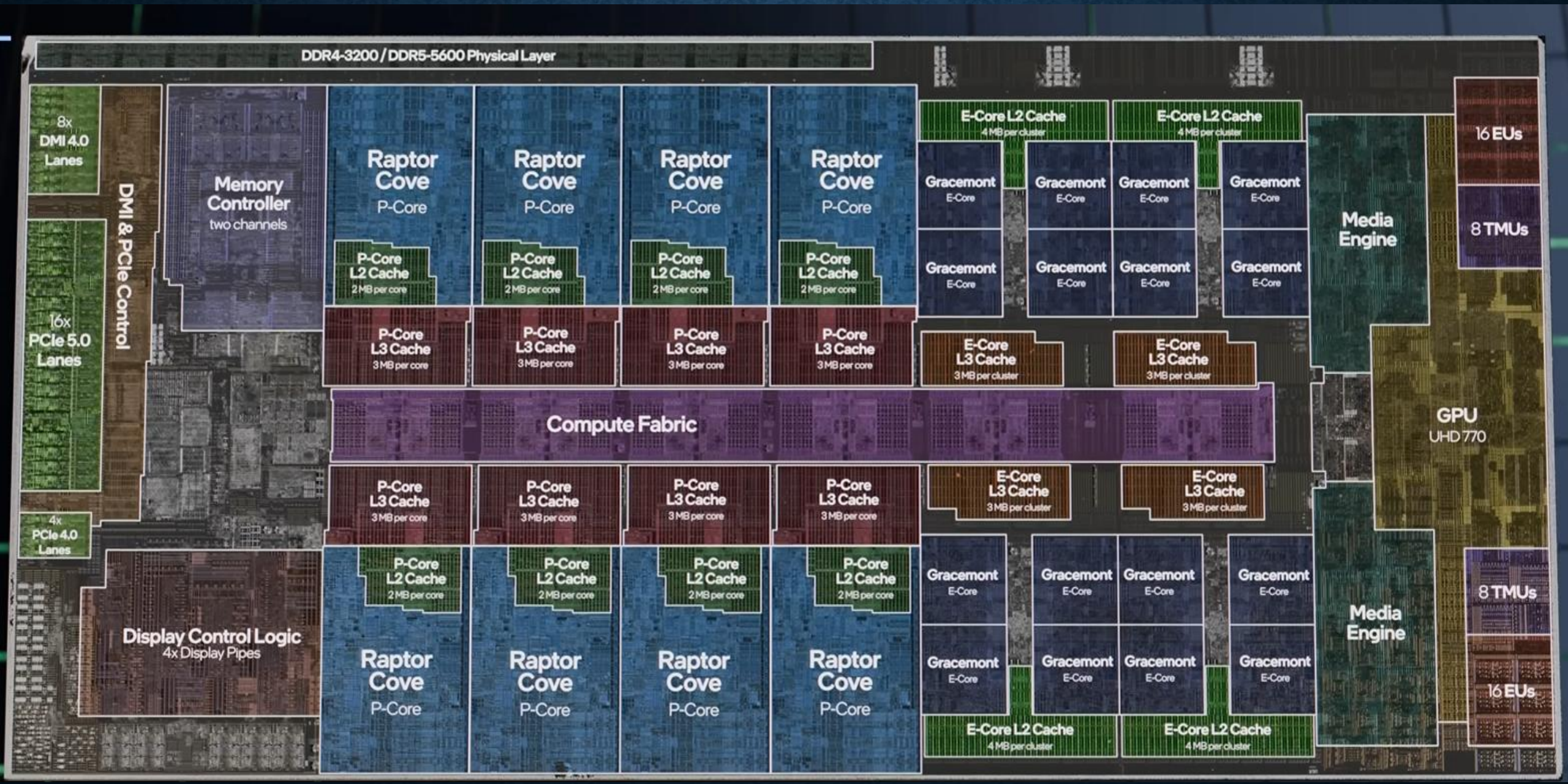
CPU MONOLITHIQUE VS CPU EN TILES



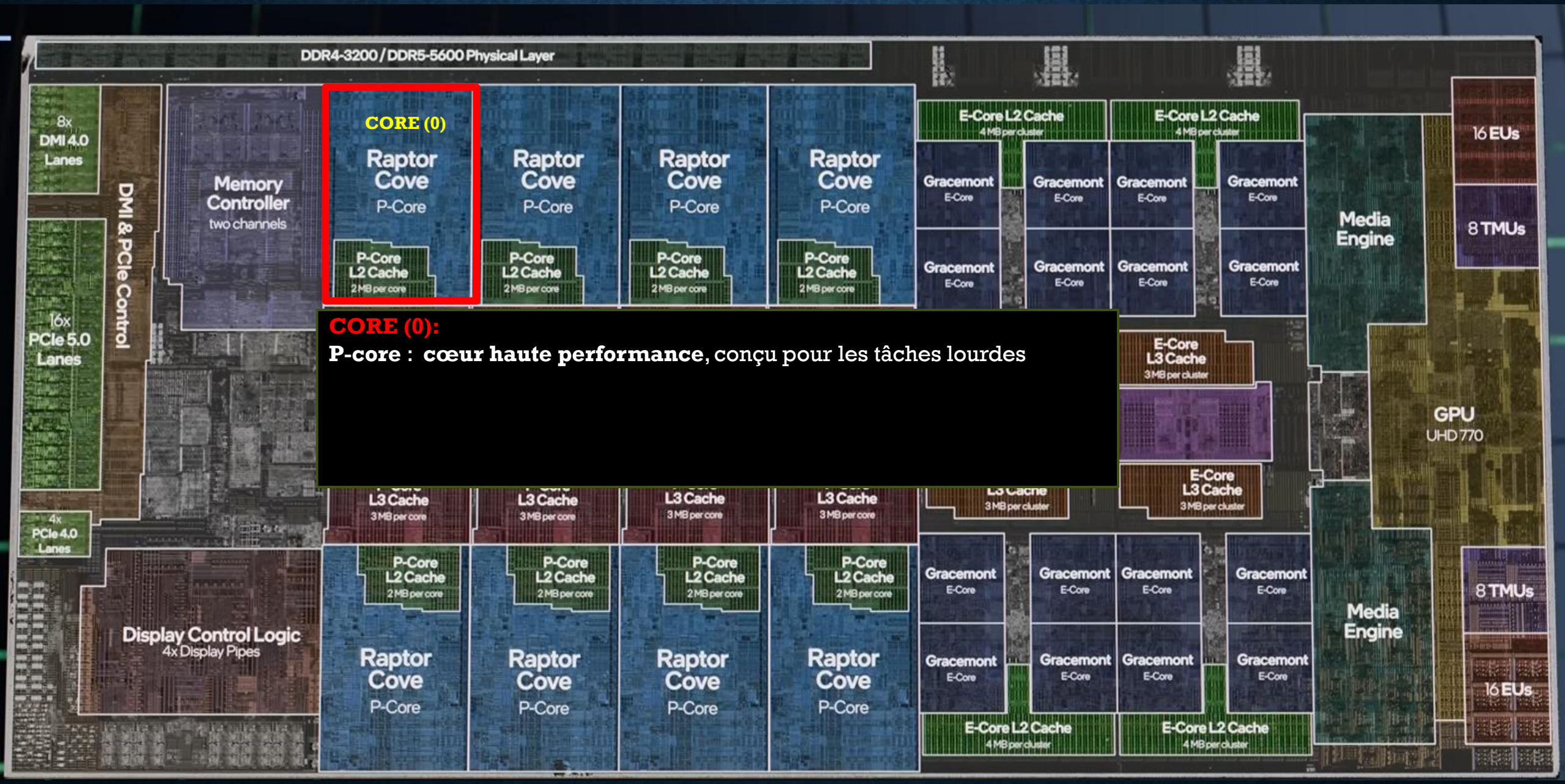
INTEL ARCHITECTURE

Année	Génération	Nom commercial	Micro-architecture P-core	Type d'architecture
2021	12e gen	Core i (Alder Lake)	Golden Cove	Hybride monolithique
2022–2023	13e gen	Core i (Raptor Lake)	Raptor Cove	Hybride monolithique
2023	14e gen (refresh)	Core i (Raptor Lake Refresh)	Raptor Cove	Hybride monolithique
2023–2024	1re gen Ultra	Core Ultra (Meteor Lake)	Redwood Cove	Tiles (chiplets)
2024–2025	2e gen Ultra	Core Ultra (Arrow Lake)	Lion Cove	Tiles
2025–2026	3e gen Ultra	Core Ultra (Panther Lake)	Cougar Cove	Tiles

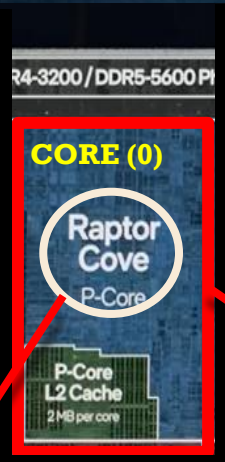
MICRO-ARCHITECTURES DE CŒURS CPU (RAPTOR COVE)



MICRO-ARCHITECTURES DE CŒURS CPU (RAPTOR COVE)



MICRO-ARCHITECTURES DE CŒURS CPU



Architecture

- **Cœur/Cores = unité physique** qui exécute des instructions.

  (Des milliards de transistors)

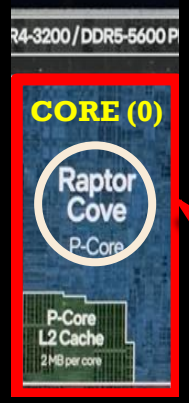
- Thread = unité logique qui représente une tâche, ou un flux d'instructions que le processeur peut exécuter.

- Plus de threads = meilleur multitâche, mais pas autant de puissance qu'un vrai cœur.

- Un cœur = un ouvrier réel.

MICRO-ARCHITECTURES DE CŒURS CPU

- Fréquence (GHz)
- Débit de calcul (FLOPS)



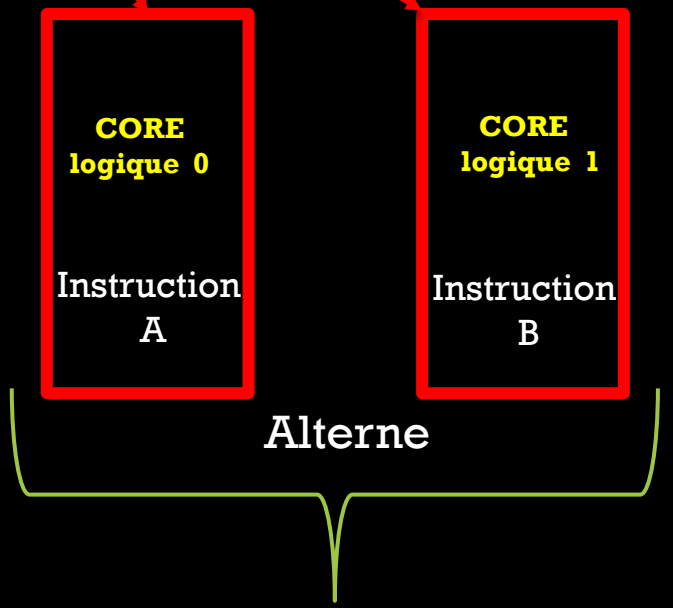
Cœur physique

Un cœur peut gérer **1 ou 2 threads** selon la technologie:

- ✓ Intel: (Hyper-Threading , ancien PC)
- ✓ AMD: (Simultaneous Multithreading, SMT).

Avantage (HT):

- Augmente l'efficacité d'un cœur,
- Améliore le multitâche
- Reduction des temps morts internes du processeur.

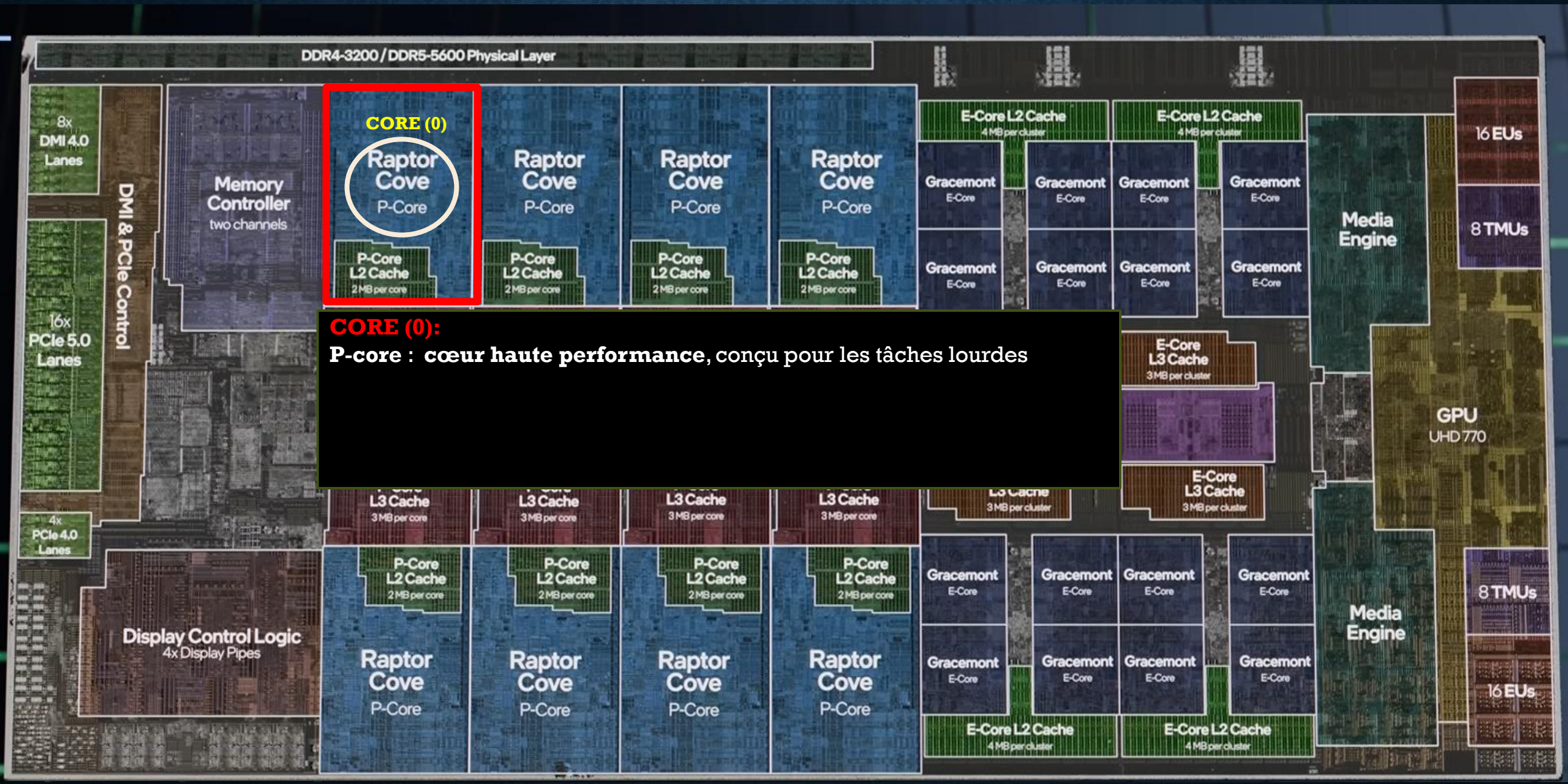


Désavantage (HT):

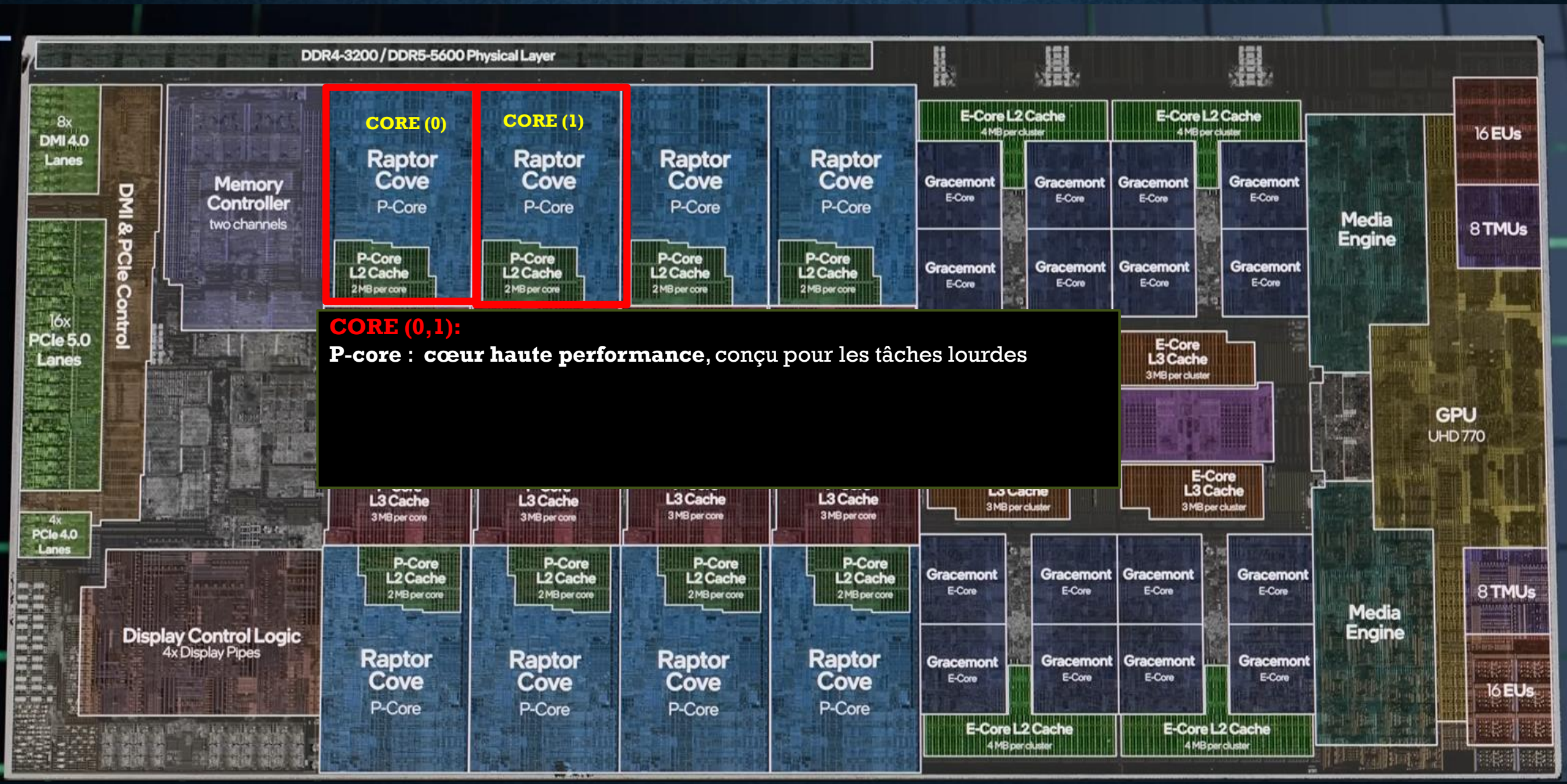
- Peut ralentir dans certains cas
- Gain variable (souvent faible)
- Consomme un peu plus
- Ajoute des risques de sécurité
- Complexifie la gestion des tâches

Abandon Hyper-Threading à partir de la 15e génération

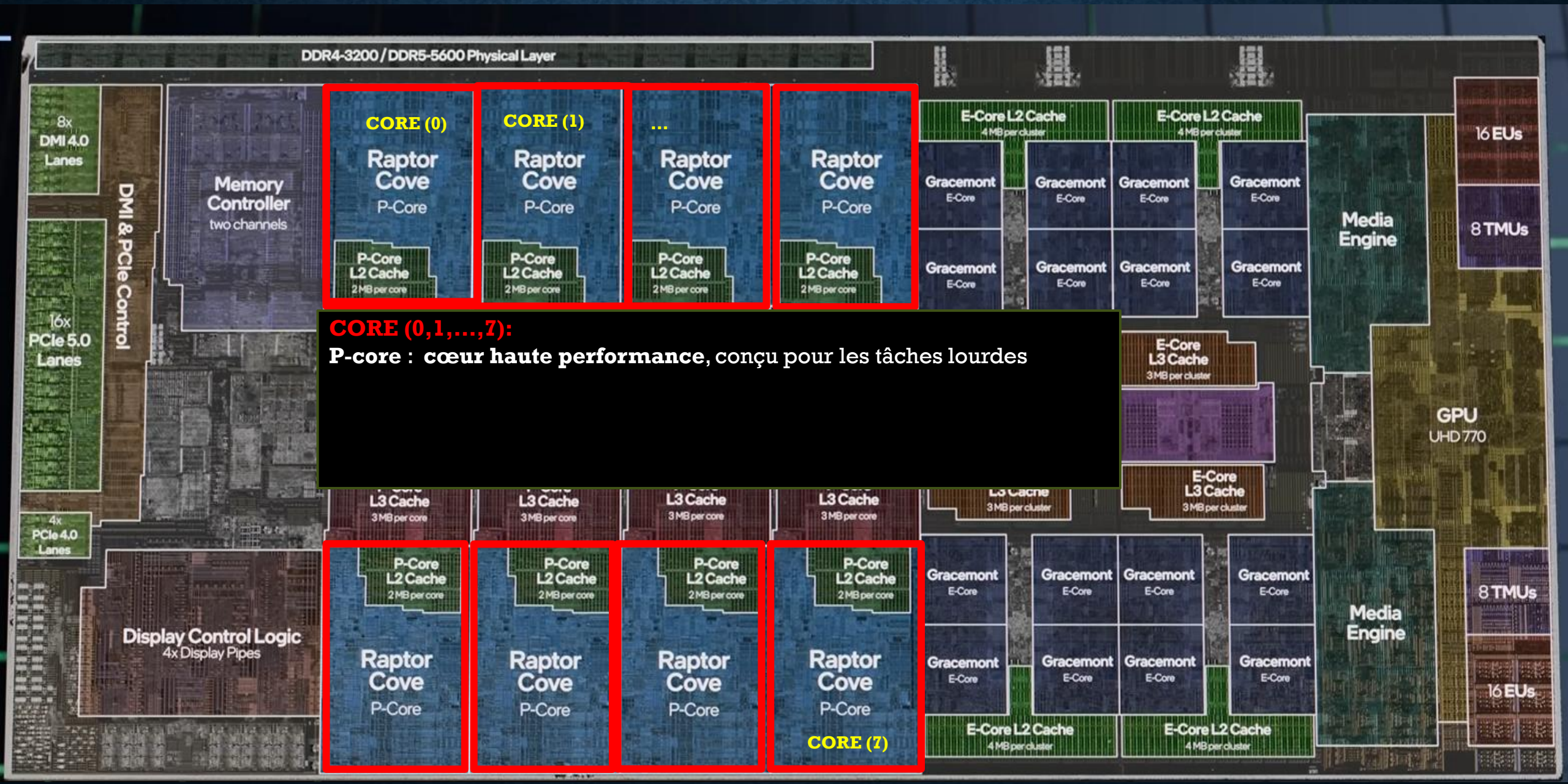
MICRO-ARCHITECTURES DE CŒURS CPU



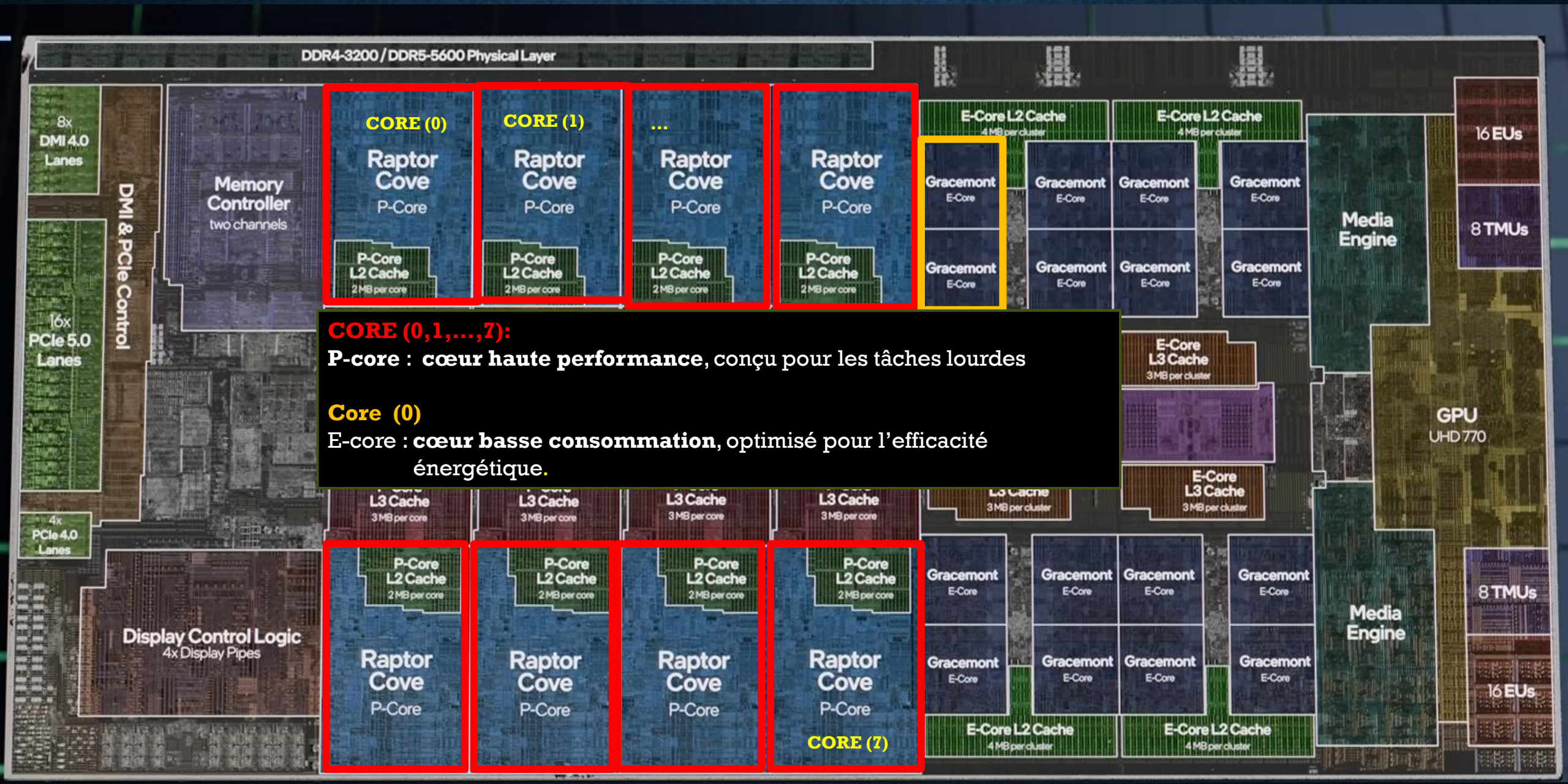
MICRO-ARCHITECTURES DE CŒURS CPU



MICRO-ARCHITECTURES DE CŒURS CPU



MICRO-ARCHITECTURES DE CŒURS CPU



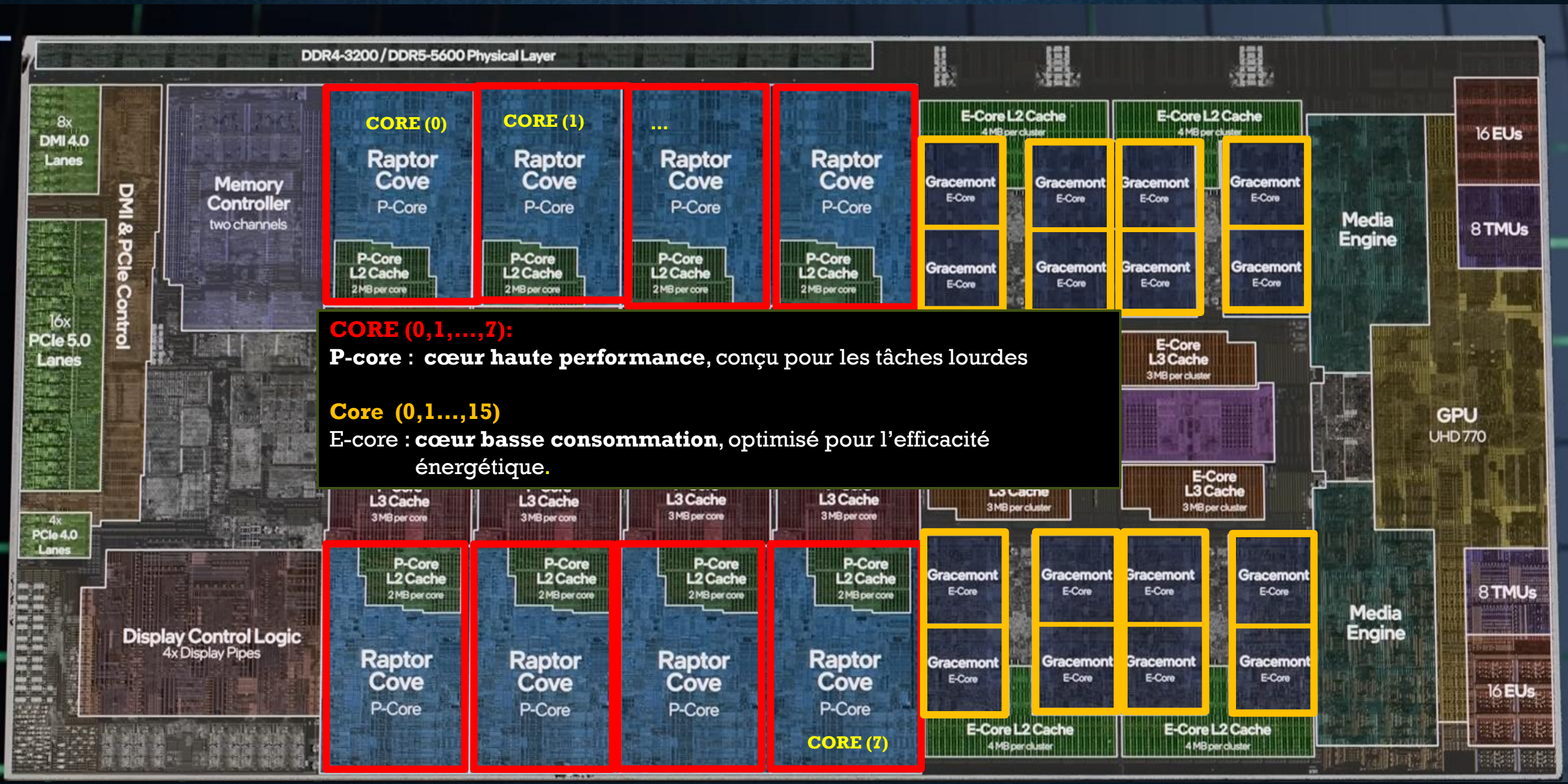
CORE (0,1,…,7):

P-core : **cœur haute performance**, conçu pour les tâches lourdes

Core (0)

E-core : **cœur basse consommation**, optimisé pour l'efficacité énergétique.

MICRO-ARCHITECTURES DE CŒURS CPU



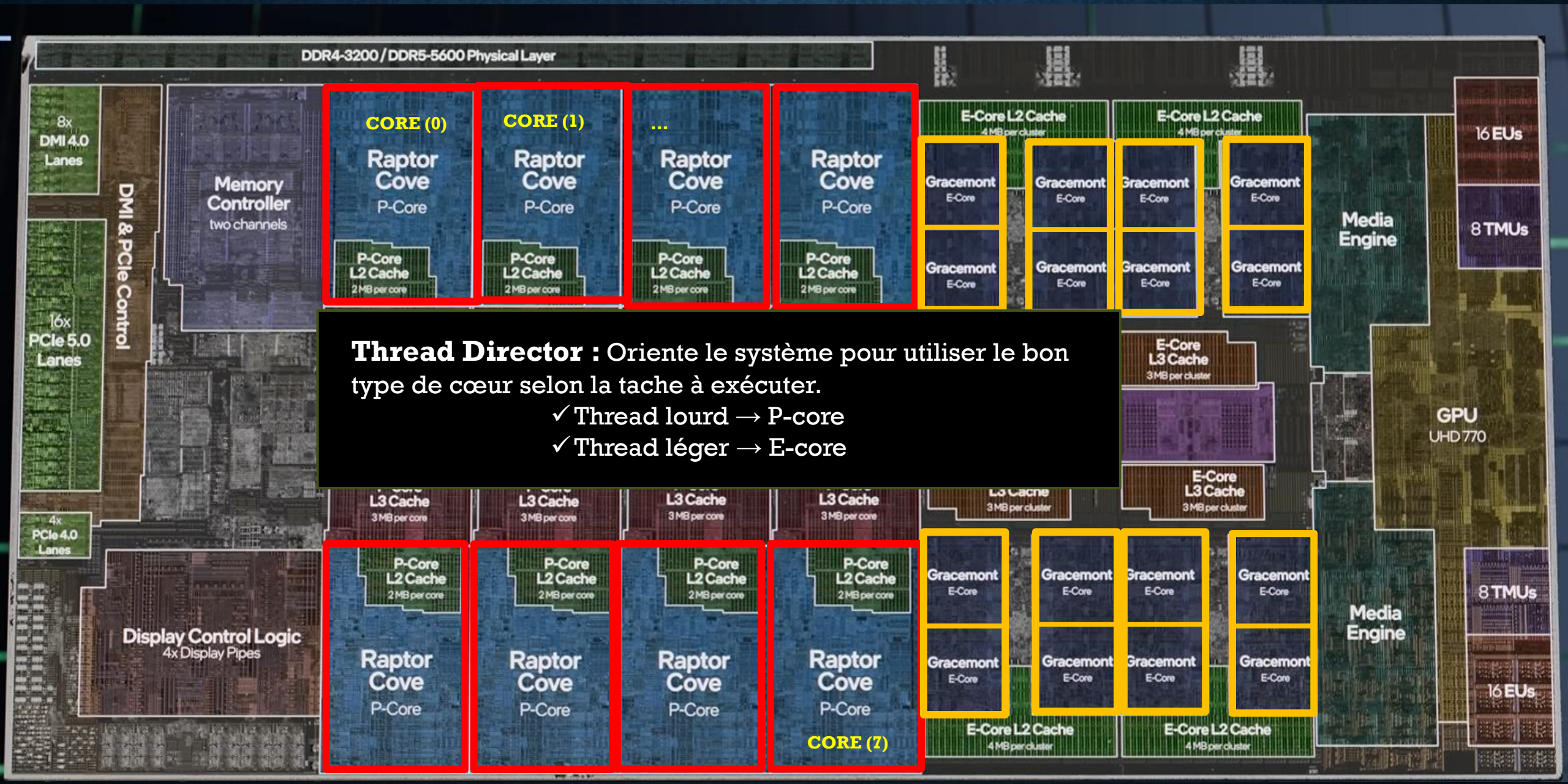
CORE (0,1,…,7):

P-core : **cœur haute performance**, conçu pour les tâches lourdes

Core (0,1…,15)

E-core : **cœur basse consommation**, optimisé pour l'efficacité énergétique.

MICRO-ARCHITECTURES DE CŒURS CPU



Thread Director : Oriente le système pour utiliser le bon type de cœur selon la tache à exécuter.

- ✓ Thread lourd → P-core
- ✓ Thread léger → E-core

CPU (ADVANCED MICRO DEVICES, AMD)

CPU (AMD, MONOLITHIQUE)

Année	Modèle	Type	Cœurs	Threads	Multithreading
2022	Ryzen 9 7950X	Desktop	16	32	SMT
2022	Ryzen 9 7900X	Desktop	12	24	SMT
2023	Ryzen 7 7800X3D	Desktop	8	16	SMT
2023	Threadripper 7980X	Workstation	64	128	SMT
2024	Threadripper PRO 9995WX	Workstation	96	192	SMT

Thread Director (non) , P-core (non), E-core (non) SMT (oui) : Simultaneous Multithreading

CPU (MONOLITHIQUE, NON MONOLITHIQUE)

- **M3 Max**

  Une seule puce → 16 cœurs (12P + 4E)

- **M3 Ultra**

  Deux M3 Max fusionnés via UltraFusion → Existe en **28 cœurs** et **32 cœurs**

Modèle	Total cœurs	P-cores	E-cores
M3 Max	16	12	4
M3 Ultra (28 cœurs)	28	**20**	**8**
M3 Ultra (32 cœurs)	32	**24**	**8**

Thread Director (non) , P-core (oui), E-core (oui), SMT (non), MT(oui)

API POUR EXPLOITER LES CŒURS/CORE EN PARALLÈLE

❖**Intel / AMD**

- ✓ Intel Thread Director (automatique, intégré au CPU)
- ✓oneTBB (le plus optimisé)
- ✓OpenMP, MPI, std::thread = natif C++
- ✓MKL (Intel Math Kernel Library)

✓Extensions au jeu d'instructions : Intel® SSE4.1, Intel® SSE4.2, Intel® AVX2 (voir CPU-Z)

API POUR EXPLOITER LES CŒURS/CORE EN PARALLÈLE

- **Apple Silicon:** CPU + GPU + IA unifiés (M1/M2/M3)

  - ✓Grand Central Dispatch (GCD)

  - ✓Metal Performance Shaders (MPS)

  - ✓OpenMP

Dans cette section, nous allons parler que des GPU NVIDIA, choisis pour leur puissance très élevée, et leur écosystème logiciel mature (CUDA, Tensor Cores, cuDNN). Ils sont aujourd'hui la référence pour l'entraînement des gros modèles IA : LLM, modèles de vision, modèles multimodaux, simulations complexes, etc..

La majorité des PC fixes et portables haut de gamme intègrent un GPU NVIDIA dédié, permettant d'exploiter la puissance de calcul parallèle (CUDA Cores, Tensor Cores) lorsque les applications le nécessitent ou lorsque le développeur choisit d'accélérer ses calculs.

GPU

GPU-Z: [Download TechPowerUp GPU-Z v2.69.0 | TechPowerUp]

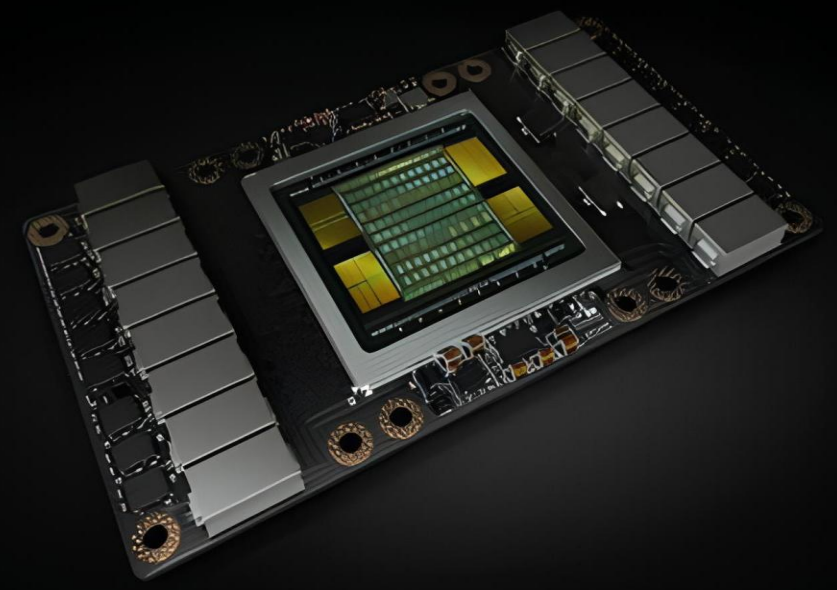
GPU-Z permet d'identifier précisément la carte graphique : modèle, architecture, mémoire, fréquences, bus, pilotes et capteurs.

ARCHITECTURE VOLTA (2017, DATA CENTER)

ANNOUNCING TESLA V100

GIANT LEAP FOR AI & HPC
VOLTA WITH NEW TENSOR CORE

21B xtors | TSMC 12nm FFN | 815mm²
5,120 CUDA cores
7.5 FP64 TFLOPS | 15 FP32 TFLOPS
NEW 120 Tensor TFLOPS
20MB SM RF | 16MB Cache | 16GB HBM2 @ 900 GB/s
300 GB/s NVLink



Premiers Tensor Cores, les fondations du deep learning moderne.

ARCHITECTURE VOLTA (2017, DATA CENTER)

- ✓La compagnie NVIDIA, spécialiste des GPU, de l'IA et du HPC, introduit Volta en 2017 : le premier GPU conçu pour l'IA moderne. Volta était réservé aux data centers. Les Tensor Cores n'arrivent au grand public qu'avec Turing (2018).

- ✓Tesla V100 (Volta) : 5 120 CUDA cores, 80 SM, Tensor Cores 1re génération → accélération massive des opérations matricielles (GEMM).

- ✓Volta marque le début des GPU optimisés pour le deep learning et la parallélisation extrême.

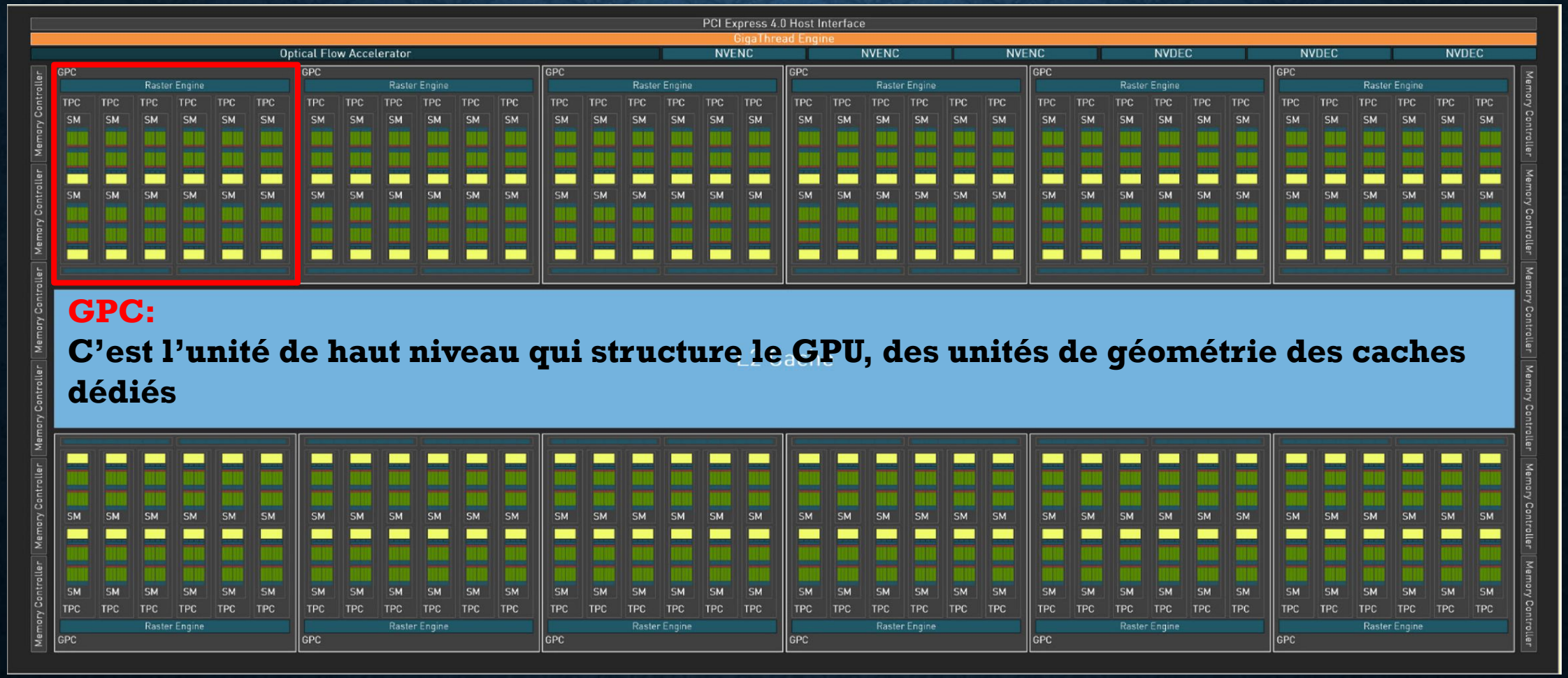
- ✓Volta est la base de toutes les architectures GPU modernes.

Dans cette section, nous nous intéressons à l'architecture Ada 6000,
la génération de GPU NVIDIA conçue pour
le calcul parallèle et l'IA moderne.

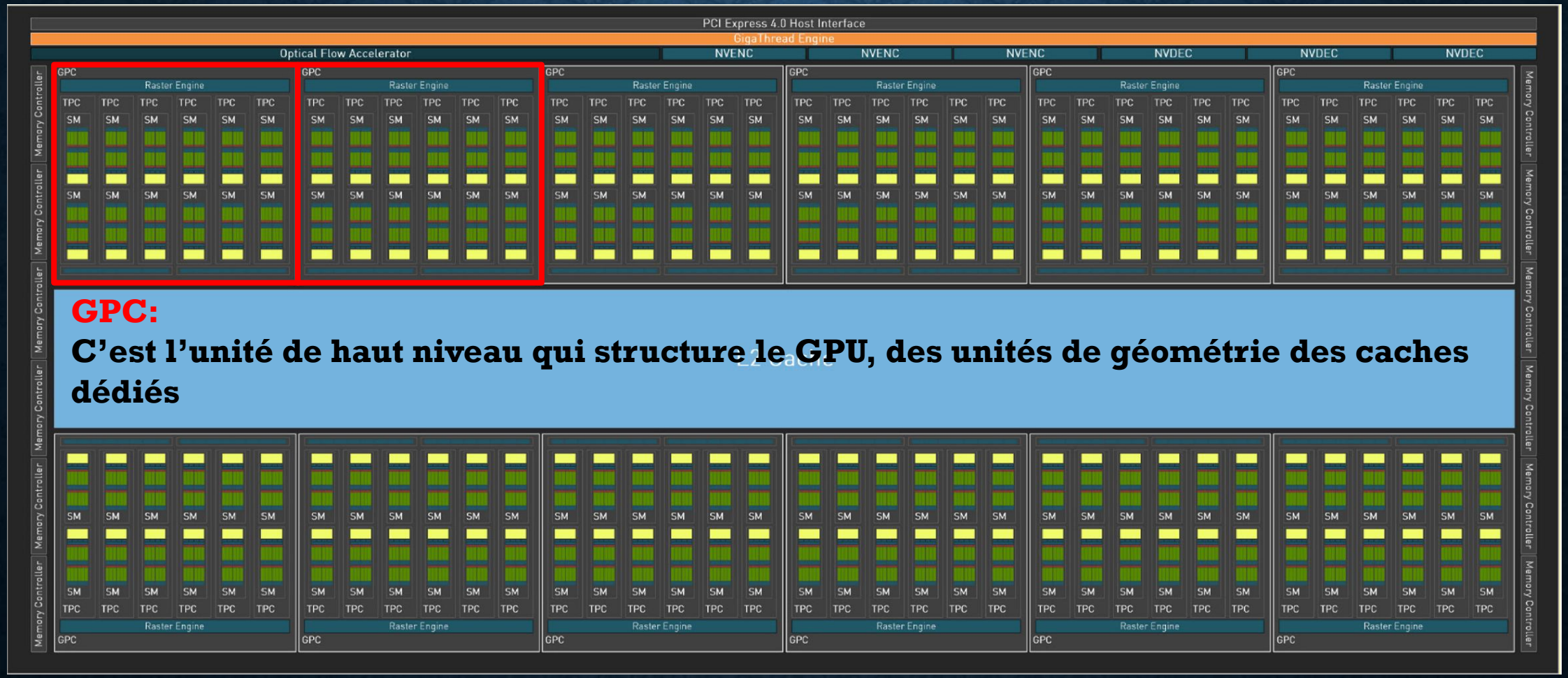
AD102 = die haut de gamme de l'architecture Ada.
RTX 6000 Ada = carte professionnelle exploitant
cette architecture pour l'IA, le rendu et le HPC.

GPU (AD102, RTX ADA6000)

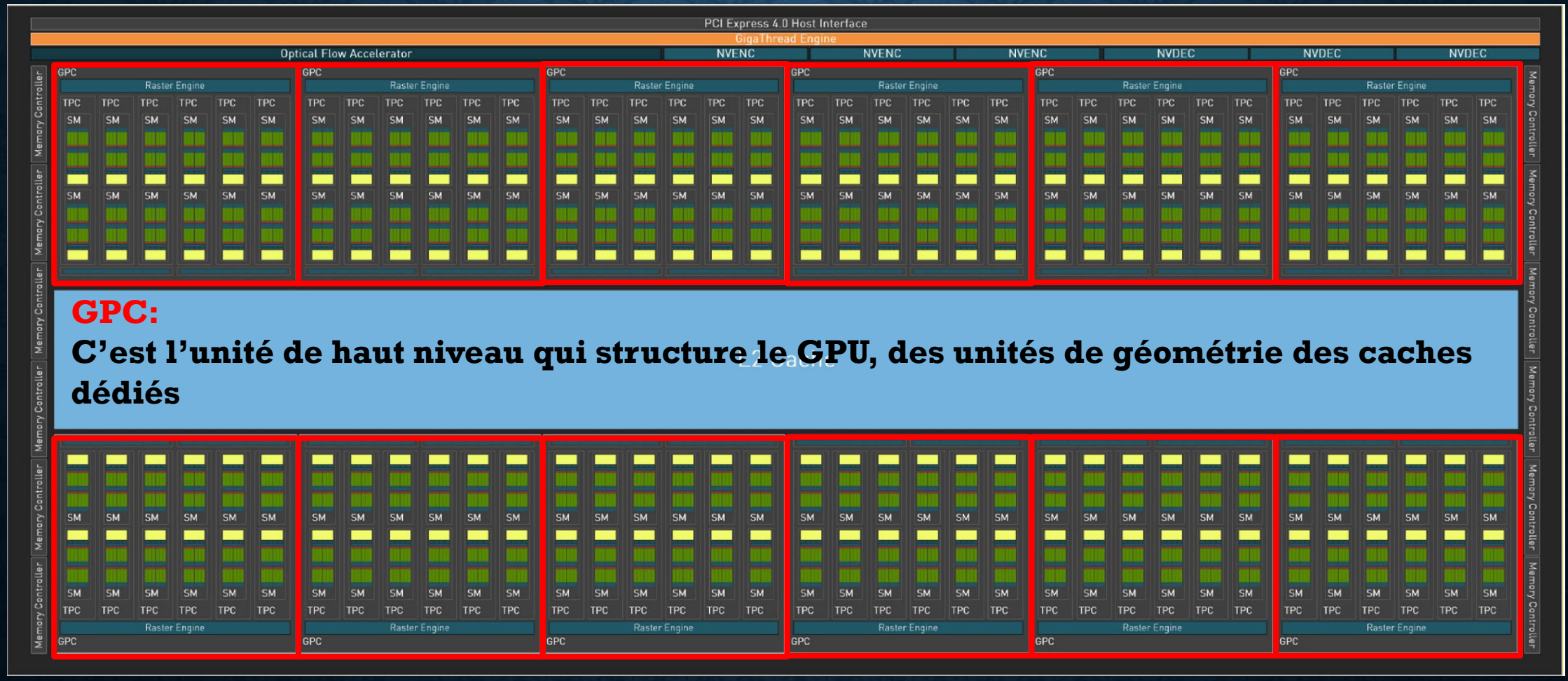
GPU (AD102, RTX ADA6000)



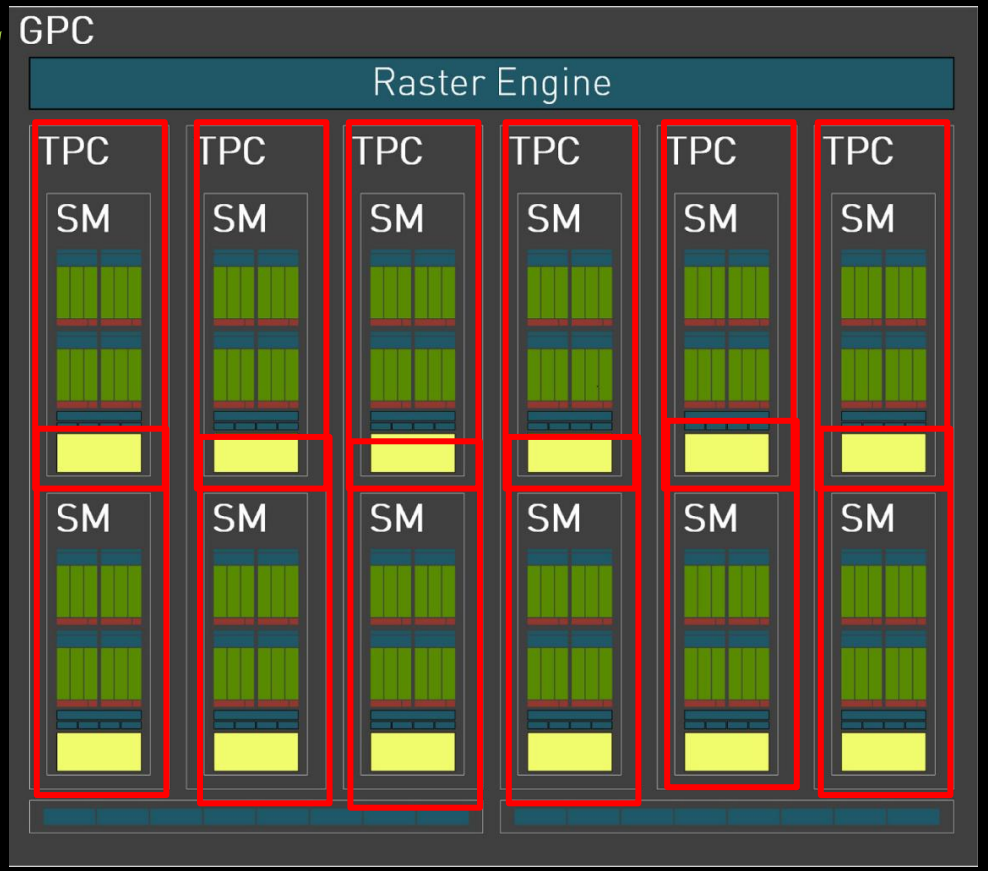
GPU (AD102, RTX ADA6000)



GPU (AD102, RTX ADA6000)



GPU (AD102, RTX ADA6000)



GPC: Graphics Processing Cluster

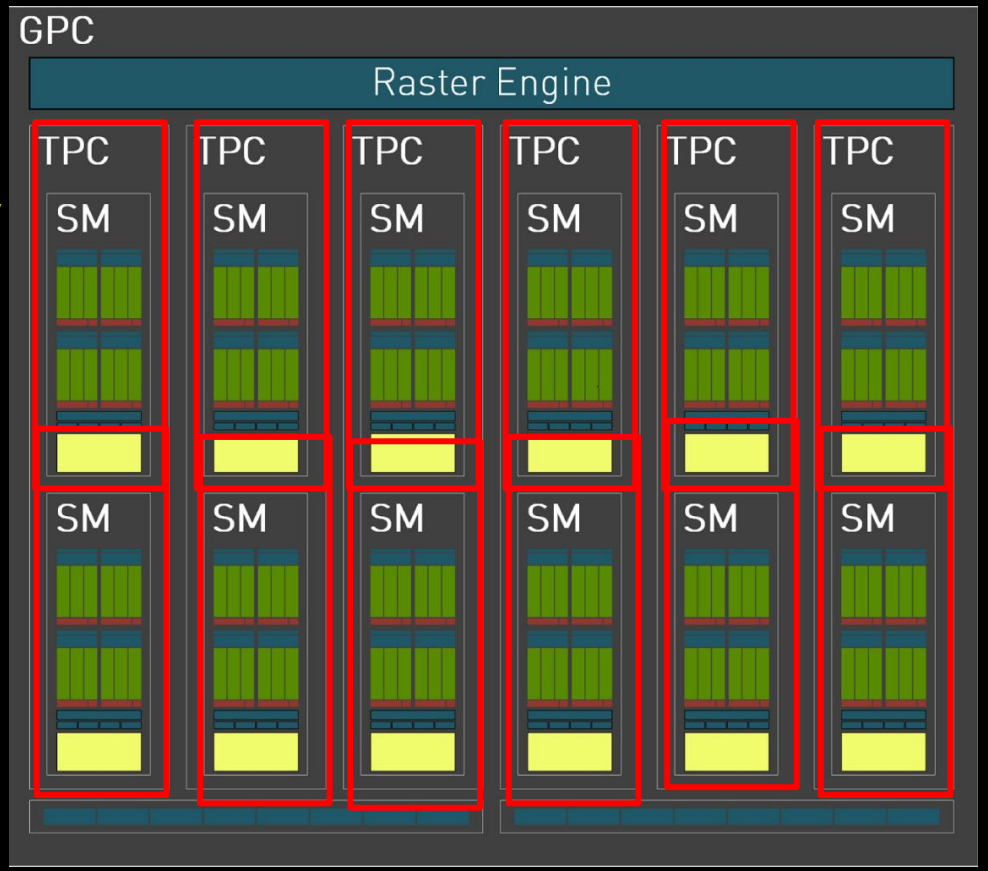
- ✓ C'est l'unité de haut niveau qui structure le GPU.
- ✓ Unités de géométrie des caches dédiés

GPU (AD102, RTX ADA6000)

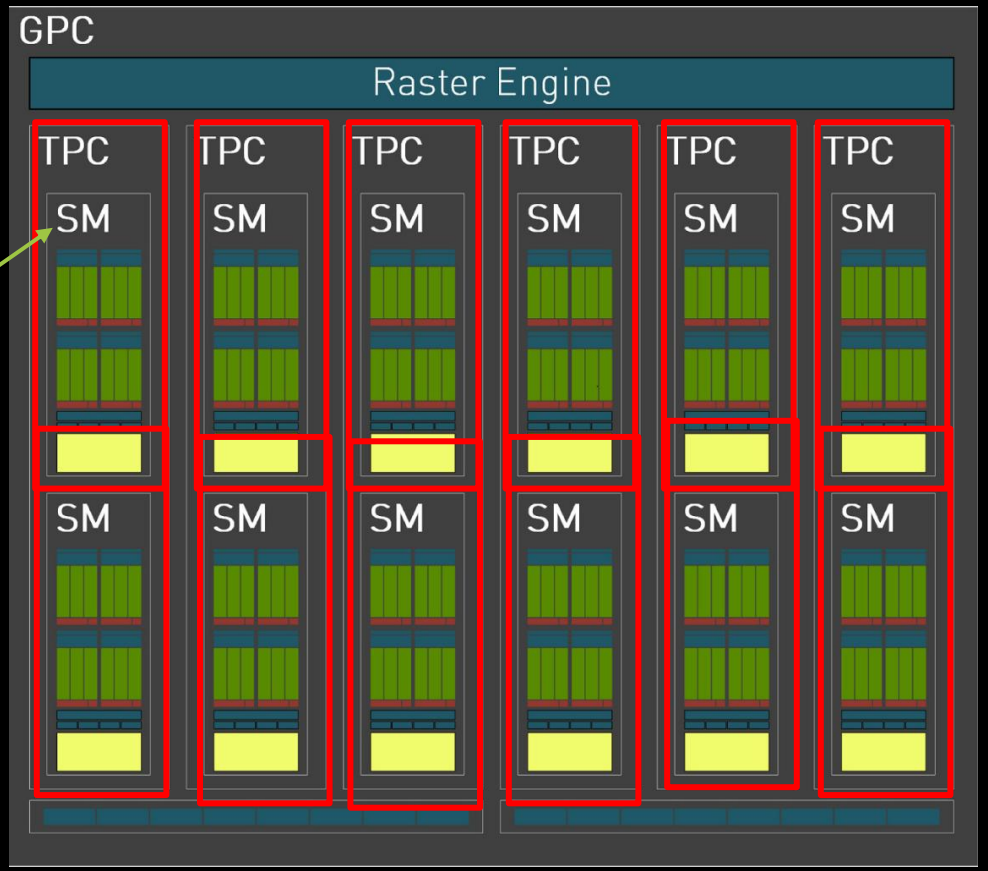
TPC: Texture/Processor Cluster

2 SM (dans la plupart des architectures)

- ✓ unités de texture (TMU)
- ✓ caches L1/TEX



GPU (AD102, RTX ADA6000)

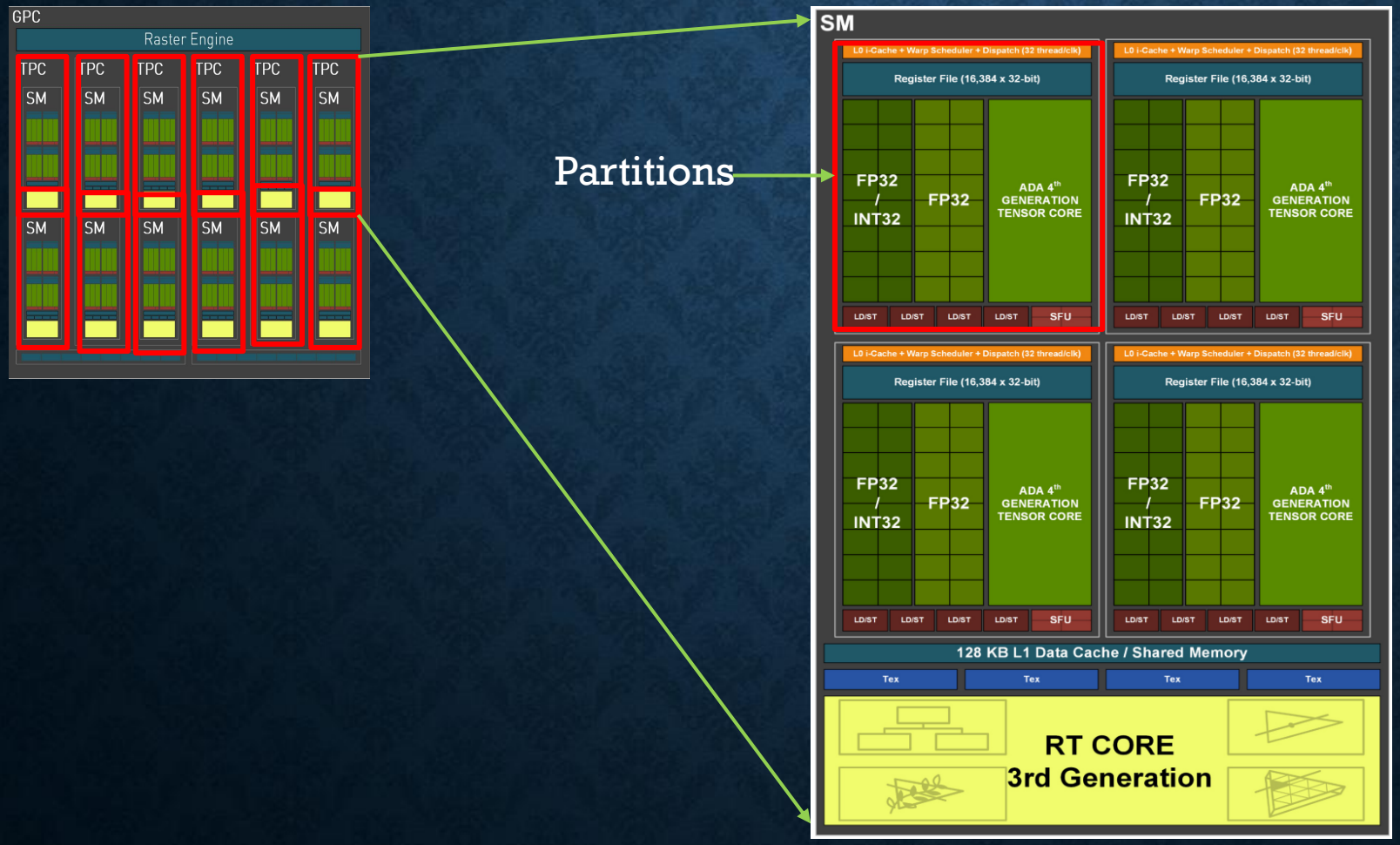


SM : *Streaming Multiprocessor*

- ✓ CUDA cores
- ✓ Tensor Cores
- ✓ RT Cores (selon architecture)
  des registres
- ✓ des unités de chargement
- ✓ un scheduler de threads

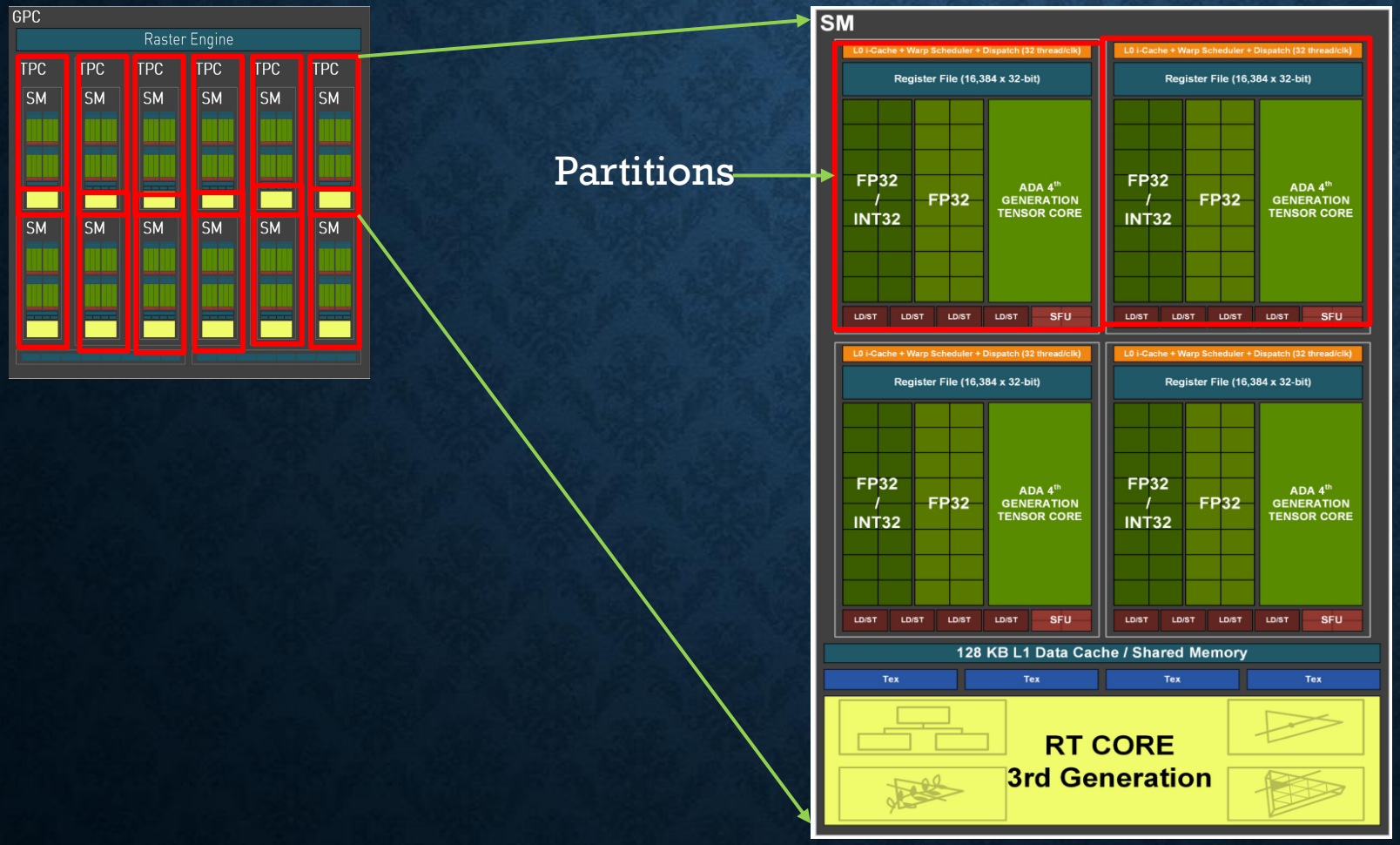
SM est l'équivalent du cœur dans un CPU

GPU (AD102, RTX ADA6000)



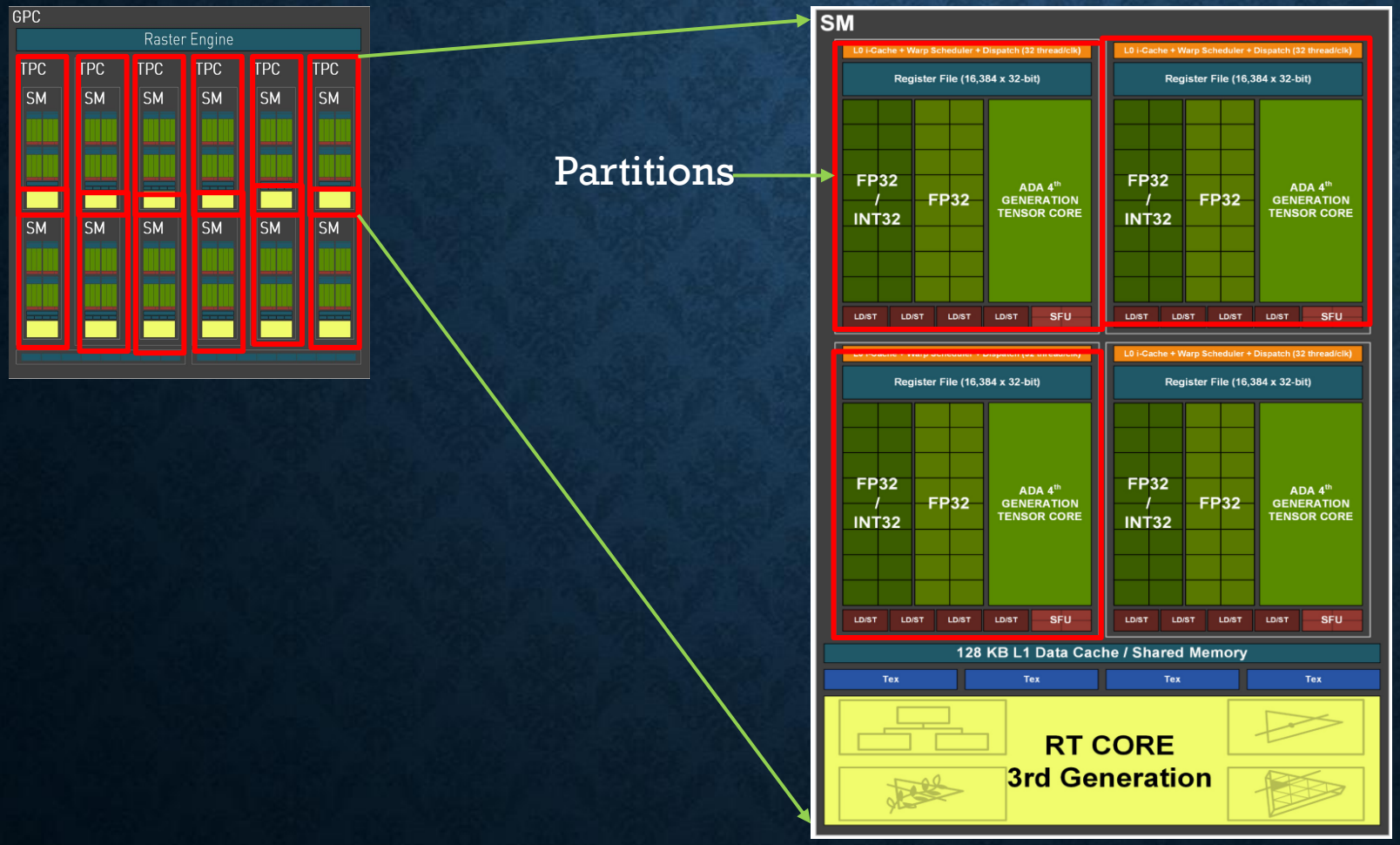
Partitions : 1

GPU (AD102, RTX ADA6000)



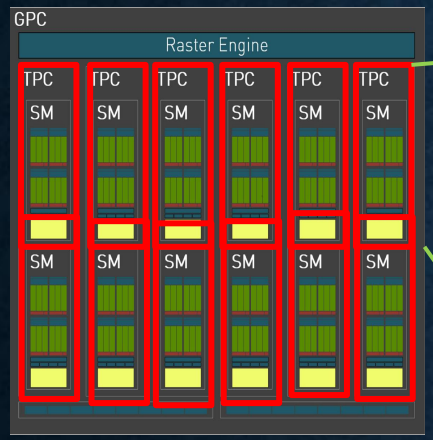
Partitions : 2

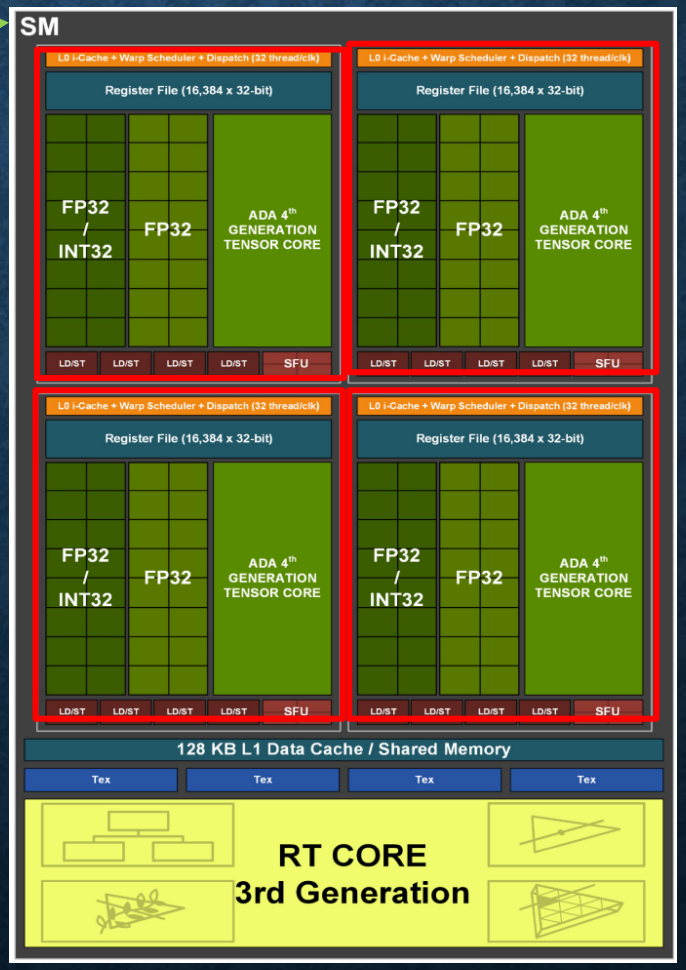
GPU (AD102, RTX ADA6000)



Partitions : 3

GPU (AD102, RTX ADA6000)





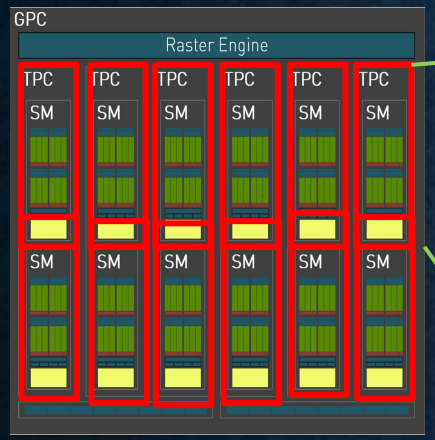
Nombre de Partitions : 4

Chaque partition : 32 cuda cores

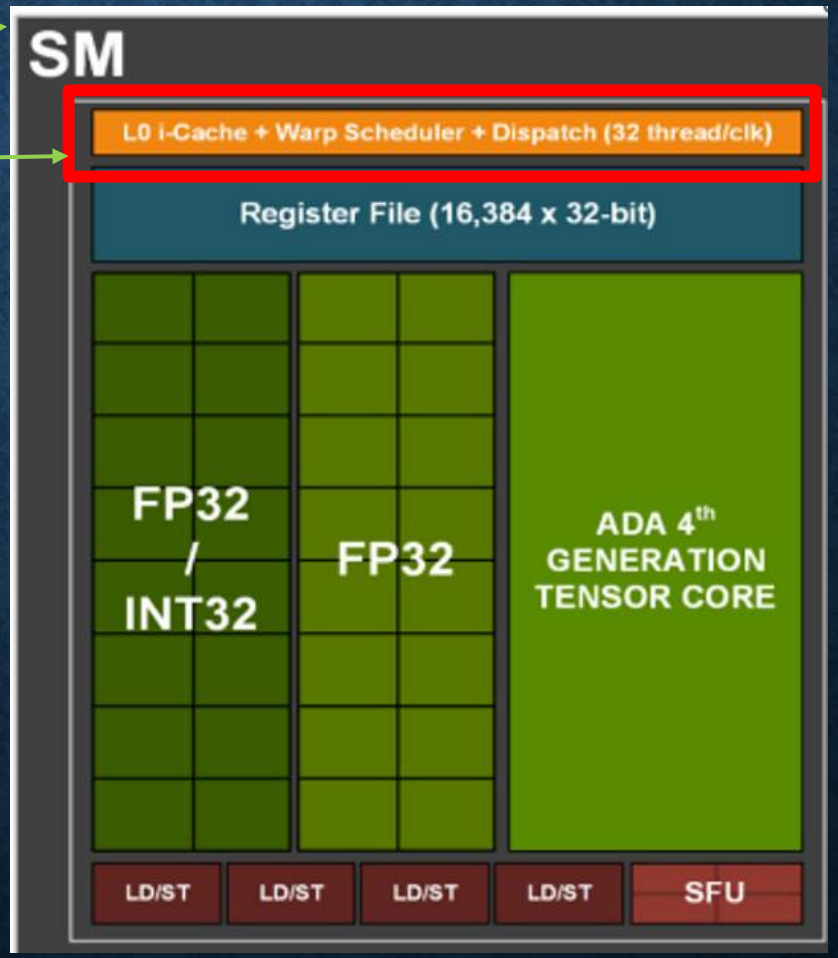
1 SM = 128 cuda core (physique)

1 SM peut gérer 48 warps

GPU (AD102, RTX ADA6000)



Warps

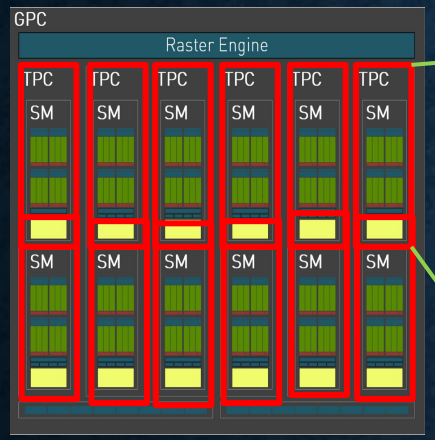


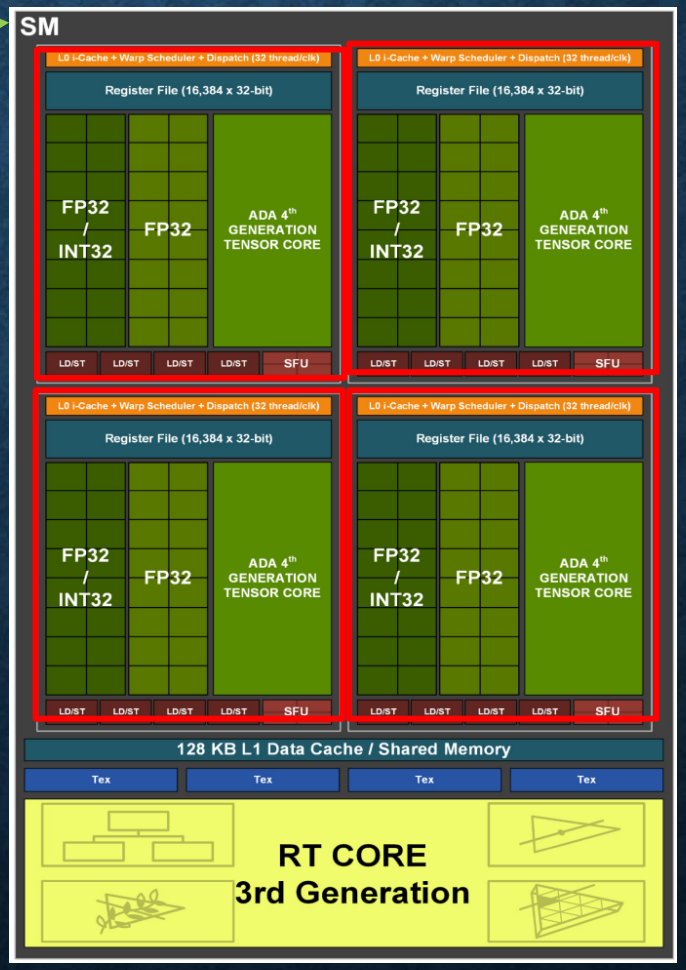
Warps schedule = 32 Threads

Un SM = 4 x 32 = 128 threads/cycle (débit)

Tous les threads d'un warp exécutent la même instruction, mais sur des données différentes (SIMT)

GPU (AD102, RTX ADA6000)





RTX ADA 6000 = 18176 threads / par cycle

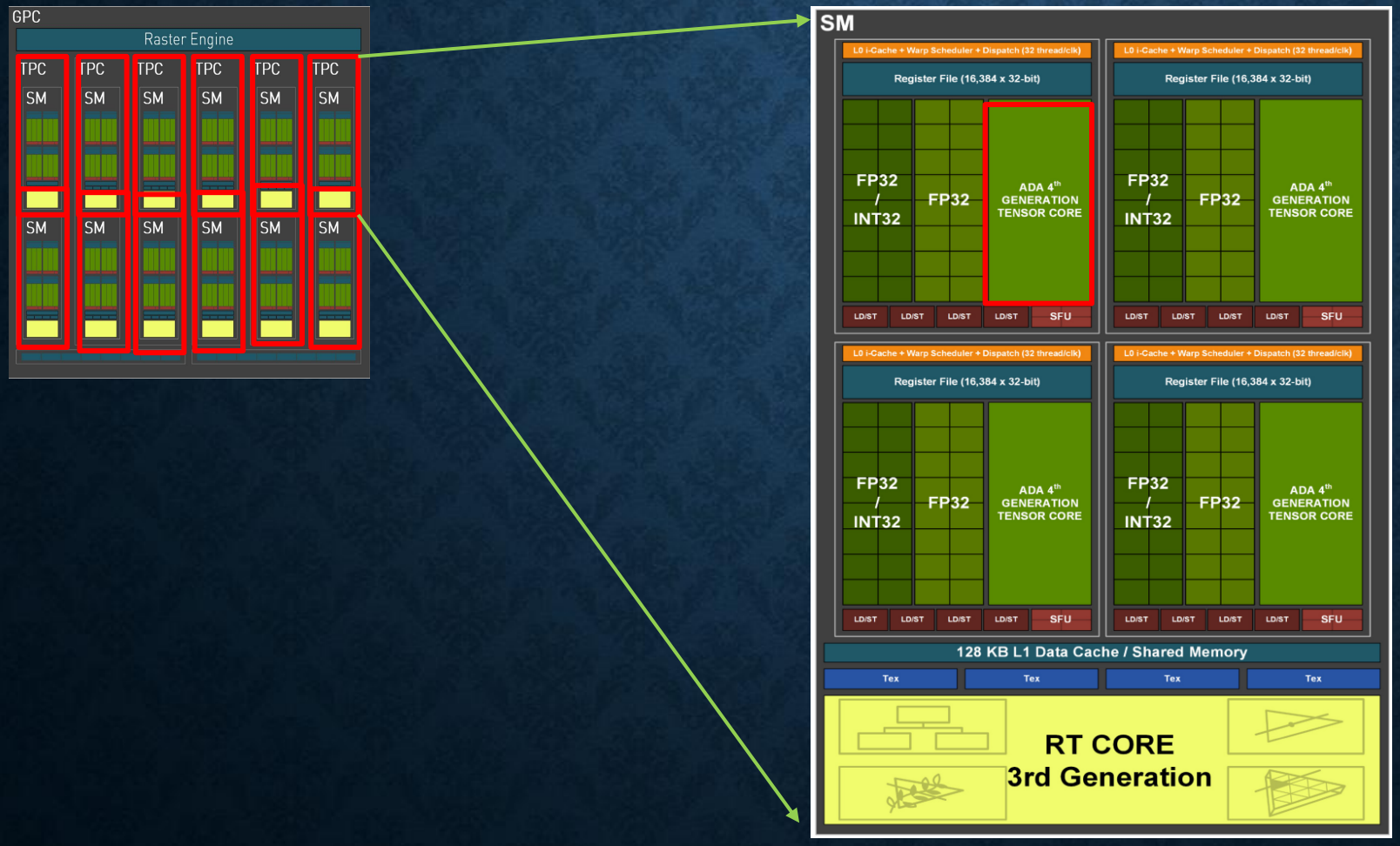
SM (Streaming Multiprocessor)

1 SM = 128 threads/cycle

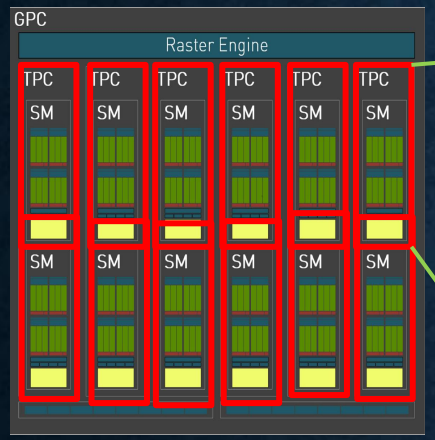
Threads maximum dans un SM (occupancy) : 1536/SM, sauf Turing 1024.

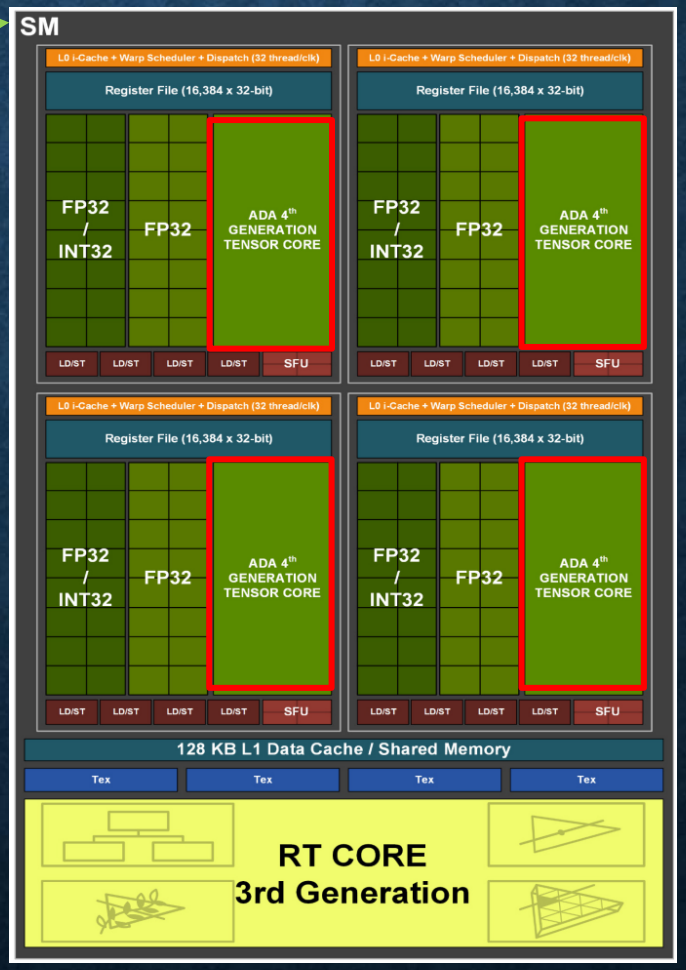
142 SM * 1536 = 221184 threads totaux.

GPU (AD102, RTX ADA6000)



GPU (AD102, RTX ADA6000)





Tensor core:

Unité spécialisée pour les multiplications de matrices (D=A×B+C, FMA)

IA, deep learning, DLSS, traitement vidéo

FORMULE GÉNÉRALE FLOPS D'UN GPU

- ✓ **Nombre de cœurs** = total des CUDA cores (FP32)
- ✓ **Opérations par cycle** = 2 (FMA = multiply + add)
- ✓ **Fréquence** = fréquence boost en hertz (Hz)

❖ **FLOPS = nombre de cœurs * 2 * fréquence**

Ada 6000 = 18176 * 2 * 2505 MHz = 91 061 760 FLOPS

FLOPS = Floating-Point Operations Per Second

nombre d'opérations flottantes effectuées en une seconde.

https://top500.org/lists/top500/2018/11/ TOP500 List - November 2025 | TOP500

AVANTAGES D'UN GPU NVIDIA

- Puissance de calcul massive en parallèle
- Accélération énorme pour l'IA
- Rendu graphique avancé
- Mémoire dédiée très rapide
- Bonne efficacité énergétique pour le calcul parallèle

DESAVANTAGES D'UN GPU NVIDIA

- Mauvais pour les tâches séquentielles
- Consommation électrique élevée
- Prix très élevé
- Dépendance à la VRAM
- Taille physique importante

Dépend de l'API:
CUDA
(Compute Unified Device Architecture)

MODÈLE ENTRAÎNÉ SUR GPU

Modèle	Paramètres
GPT-4	~1,76–1,8T
Gemini Ultra	centaines de milliards (non publiés)
Gemini Pro	dizaines à ~100B
Gemini (autre source)	~256B
Llama	7B → 405B → jusqu'à 2T selon versions
DeepSeek V4-Pro	1,6T total / 49B actifs
DeepSeek V4-Flash	284B total / 13B actifs
Grok-4.1	~3T
Grok-5	6T

MODÈLE ENTRAÎNÉ SUR CPU

Ce sont des modèles < 100M paramètres

- ✓ Logistic Regression / SVM / Random Forest (scikit-learn)
- ✓ Petits CNN (MNIST, CIFAR-10)
- ✓ Petits Transformers (TinyBERT, DistilBERT sur petits datasets)
- ✓ LLaMA-3 8B → fine-tuning possible sur CPU, mais très lent
- ✓ GPT-2 small (124M) → entraînable sur CPU (mais lent)
- ✓ MobileNet / EfficientNet-Lite (petits datasets)
- ✓ RNN / LSTM de petite taille

Ces modèles peuvent être entraînés sur CPU mais avec un temps d'entraînement très long.

MODÈLE ENTRAÎNÉ SUR CPU

Modèles explicitement conçus pour CPU:

- ✓ Intel OpenVINO → optimisé pour inference, pas entraînement
- ✓ Apple Accelerate / MPS → CPU + GPU Apple
- ✓ ONNX Runtime CPU → pour petits modèles
- ✓ FastText (Facebook) → entraîné sur CPU
- ✓ Word2Vec / GloVe → CPU friendly

MODÈLES IMPOSSIBLES À ENTRAÎNER SUR CPU

- ✓ GPT-3 / GPT-4
- ✓ LLaMA-2 / LLaMA-3 (7B, 13B, 70B)
- ✓ Mistral 7B / Mixtral 8x7B
- ✓ DeepSeek V3 / V4
- ✓ Gemini
- ✓ Claude
- ✓ Stable Diffusion
- ✓ ViT-Large / ViT-Huge
- ✓ ResNet-152 / EfficientNet-XL

Ces modèles nécessitent des GPU (A100, H100, B100) ou TPU.

MERCI.

RÉFÉRENCES

R. H. Dennard et al. (1974) — *Design of Ion-Implanted MOSFET's with Very Small Physical*

Mark Bohr (Intel) — *A 30 Year Retrospective on Dennard's MOSFET Scaling Paper* → Analyse historique + impact sur l'industrie.

Zhe Jia et al. (2018) — *Dissecting the NVIDIA Volta GPU Architecture via Microbenchmarking* → Étude technique détaillée (ISA, mémoire, SM).

https://openai.com/

https://www.intel.fr/content/www/fr/fr/products/sku/242293/intel-core-ultra-9-processor-275hx-36m-cache-up-to-5-40-ghz/specifications.html

PassMark CPU Benchmarks - Single threaded - Desktop - Page 1

https://www.cpubenchmark.net/laptop.html

https://semianalysis.com/

https://epoch.ai/

https://www.eleuther.ai/

https://explodingtopics.com/blog/gpt-parameters

https://mljourney.com/gemini-ai-model-parameters-and-performance-benchmarks/

https://en.wikipedia.org/wiki/Llama_%28language_model%29

https://deepinfra.com/deepseek-ai/DeepSeek-V4-Pro/api

https://ybuild.ai/fr/blog/grok-xai-model-parameters-identifier-complete-guide-2026