RTX ON – THE NVIDIA TURING GPU

John Burgess, NVIDIA

INTRODUCING TURING

Greatest Leap Since 2006 CUDA GPU

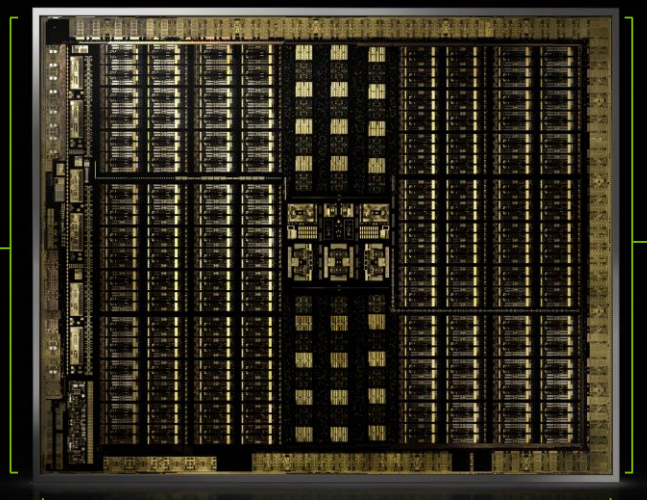
Turing SM

14 TFLOPS + 14 TIPS

Concurrent FP & INT

Enhanced L1 cache

Uniform datapath & RF

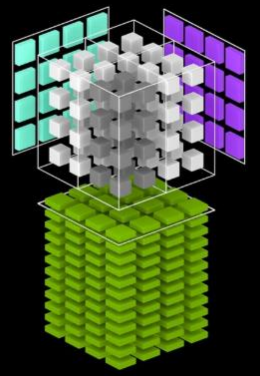


Tensor Core

114 TFLOPS FP16

228 TOPS INT8

455 TOPS INT4



RT Core

First Ray Tracing GPU

10 Giga Rays/sec

Ray Triangle Intersection

BVH Traversal

INTRODUCING TURING

TU102 – TITAN RTX
18.6 BILLION TRANSISTORS

SM	72
CUDA CORES	4608
TENSOR CORES	576
RT CORES	72
GEOMETRY UNITS	36
TEXTURE UNITS	288
ROP UNITS	96
MEMORY	384-bit 7 GHz GDDR6
NVLINK CHANNELS	2

NVIDIA TURING GPU – NEW EFFICIENT SM

Turing SM >1.5x Pascal SM Performance

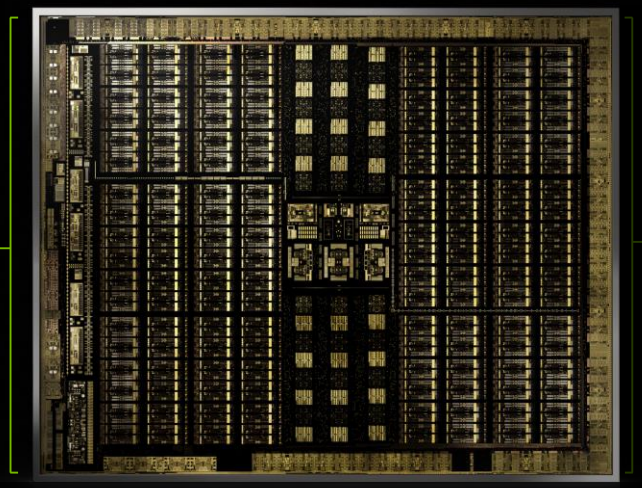
Turing SM

14 TFLOPS + 14 TIPS

Concurrent FP & INT

Enhanced L1 cache

Uniform datapath & RF

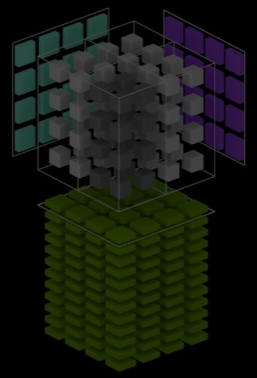


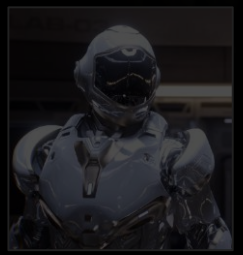
Tensor Core

114 TFLOPS FP16

228 TOPS INT8

455 TOPS INT4





RT Core

First Ray Tracing GPU

10 Giga Rays/sec

Ray Triangle Intersection

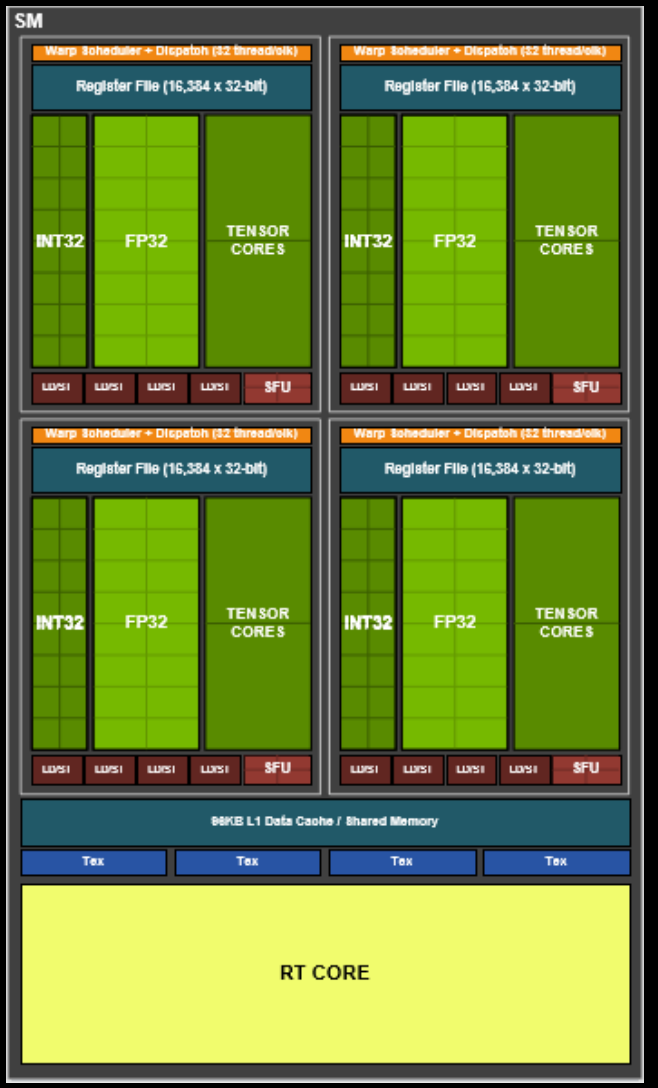
BVH Traversal

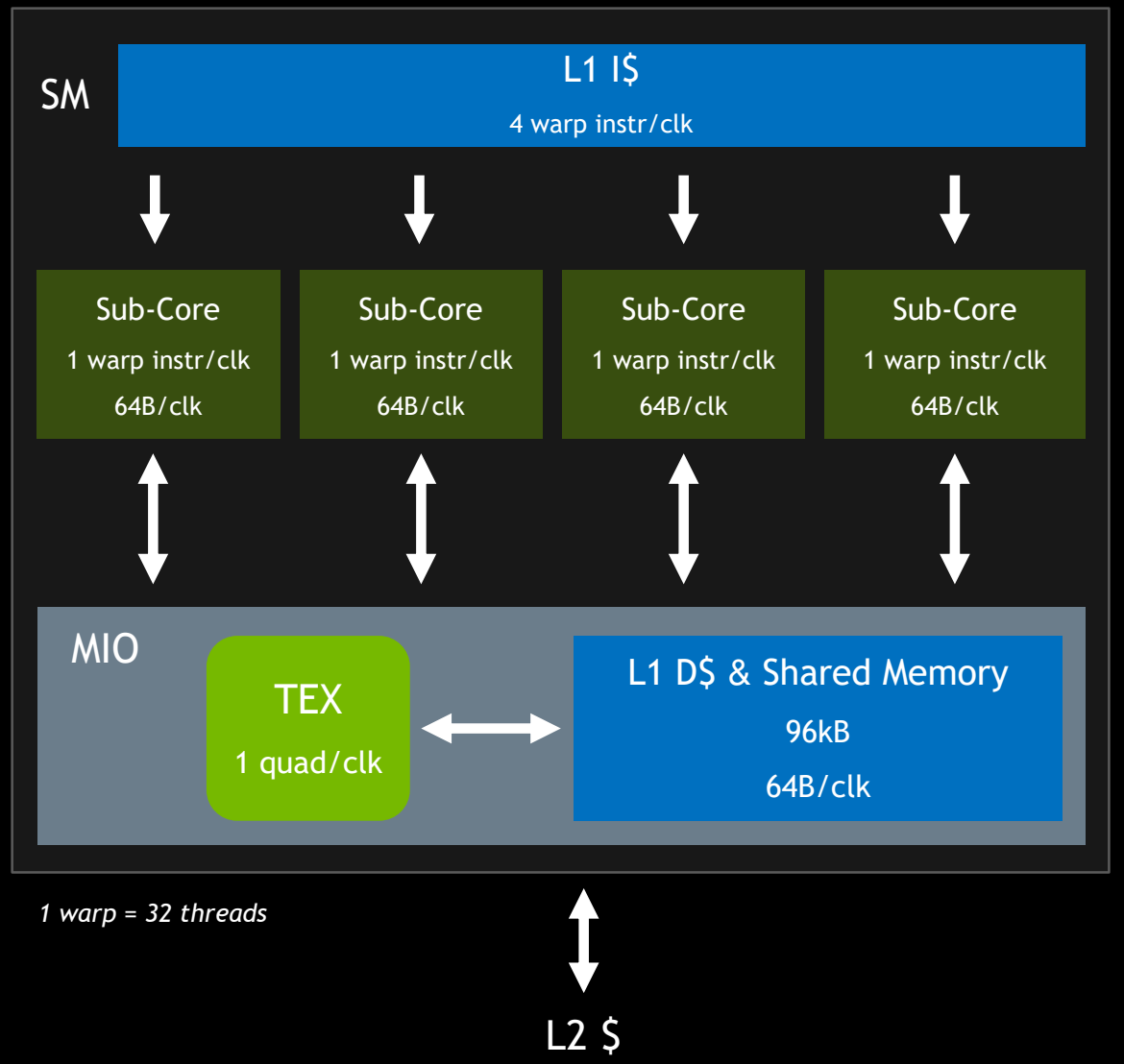
TURING SM

Concurrent FP & INT Execution Datapaths

Enhanced L1 cache

Uniform Datapath & RF





TURING SM MICROARCHITECTURE

Evolved for Efficiency

Built on foundation of Volta SM

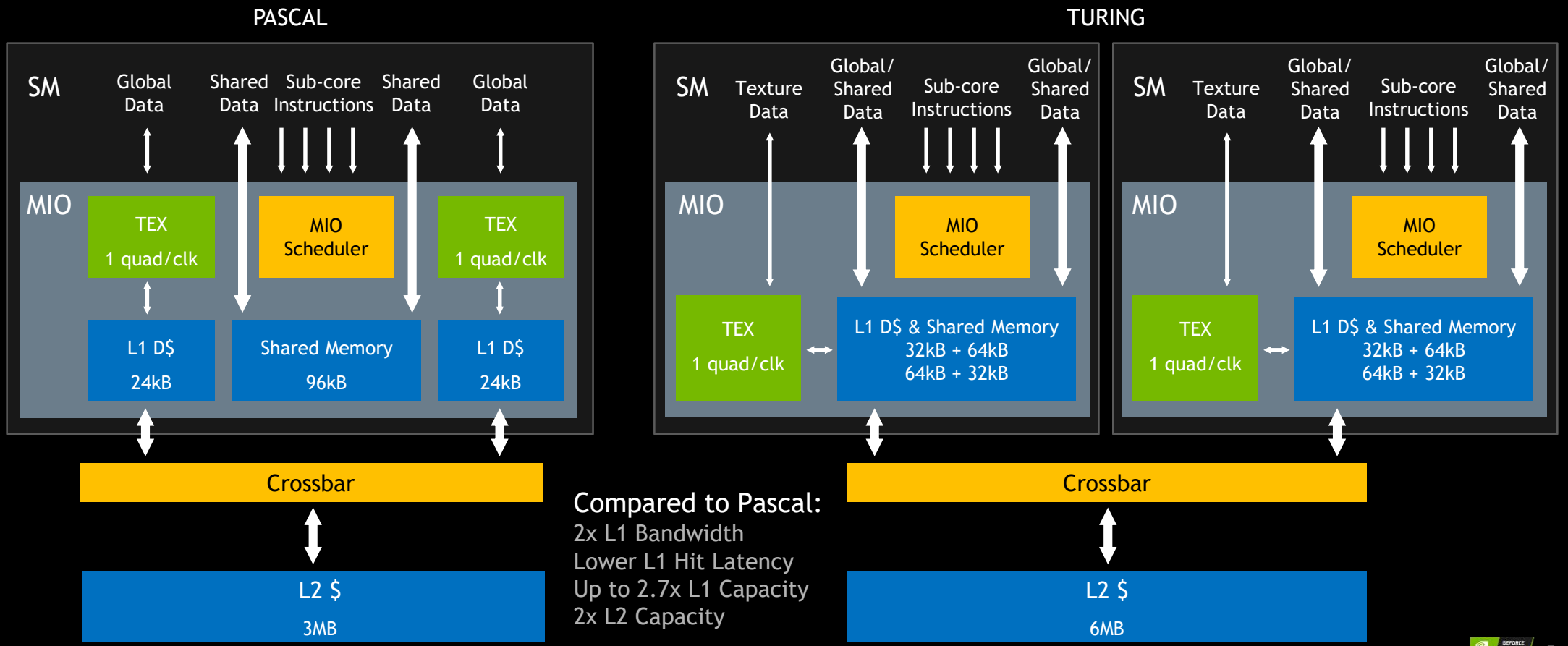
(V100: HPC/Datacenter solution between Pascal and Turing Architectures: see HotChips2017 talk)

Compared to Pascal, Turing provides:

- Twice the schedulers
- Simplified issue logic
- Large, fast L1 cache unified with TEX $ and Shared Memory

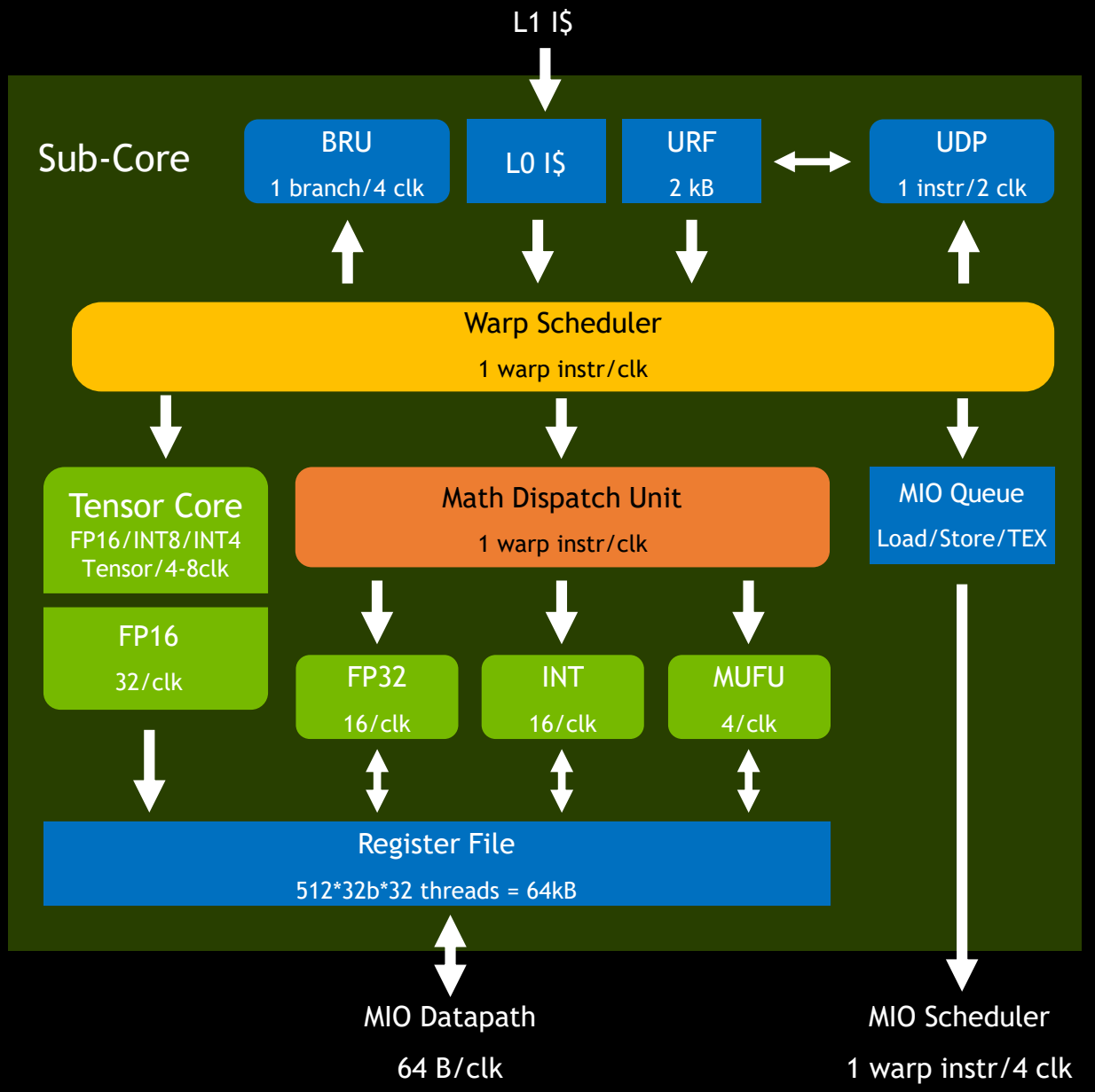
NEW CACHE & SHARED MEM ARCHITECTURE

Evolved for Efficiency



Compared to Pascal:

- 2x L1 Bandwidth
- Lower L1 Hit Latency
- Up to 2.7x L1 Capacity
- 2x L2 Capacity



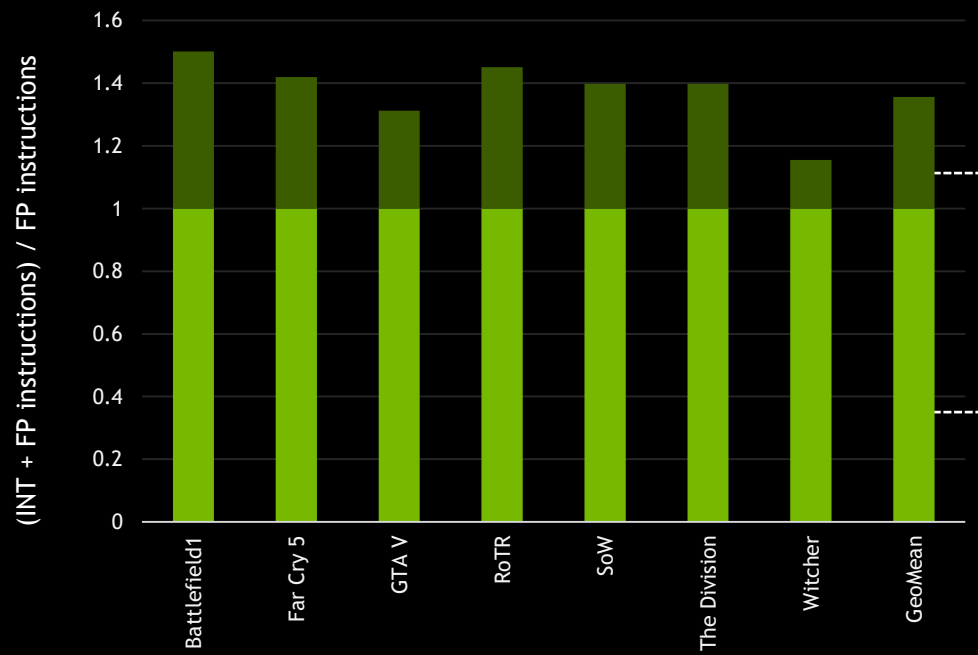
TURING SM MICROARCHITECTURE

Evolved for Efficiency

Compared to Pascal:

- Twice the register file capacity
- Improved SIMT model & branch unit
- Concurrent FP and INT execution
- New Uniform registers and datapath
- New Tensor Core
  - 16x8x8 FP16 tensor/8 clk
  - 8x8x16 INT8 tensor/4 clk
  - 8x8x32 INT4 tensor/4 clk
- Fast FP16 math

CONCURRENT EXECUTION



Per 100 FP instructions,
average 36 INT PIPE instructions
(ie iadd, select, fp min/max, compare etc)

UNIFORM DATAPATH & REGISTER FILE

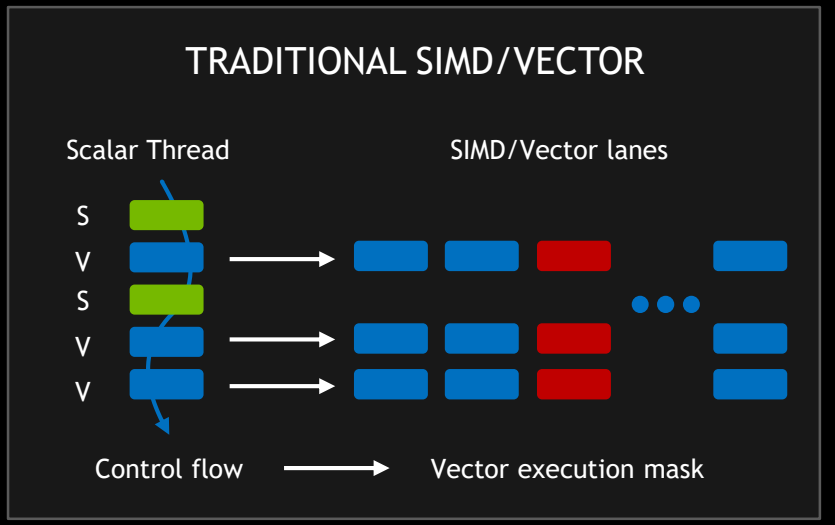
Goal: Exploit redundant computation & data across multiple threads while preserving our Independent Thread Scheduling model

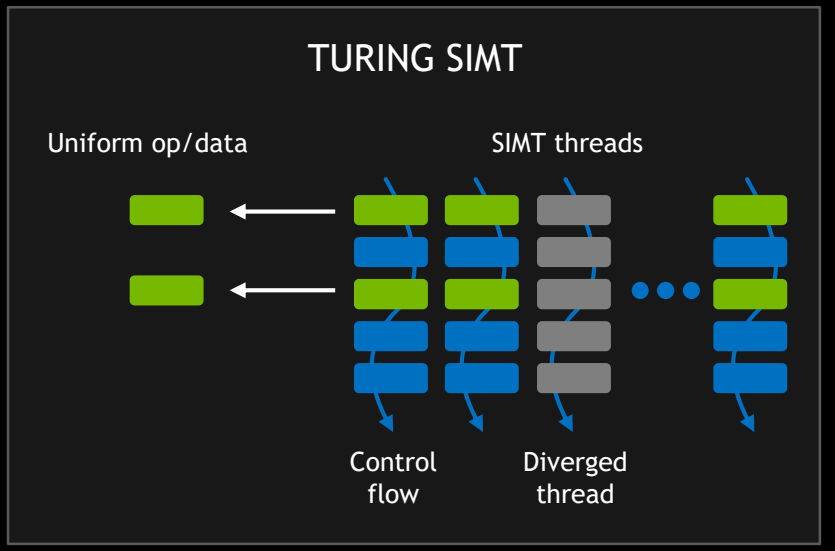
Automatically promote ops/data when warp-uniform data is detected

- Compiler + hardware assist
- Executed by an independent datapath
- ‘Reverse vectorization’

Example: Enabling DX12 bindless constants with URF/UDP on Forza MS7 yielded +12.7% performance

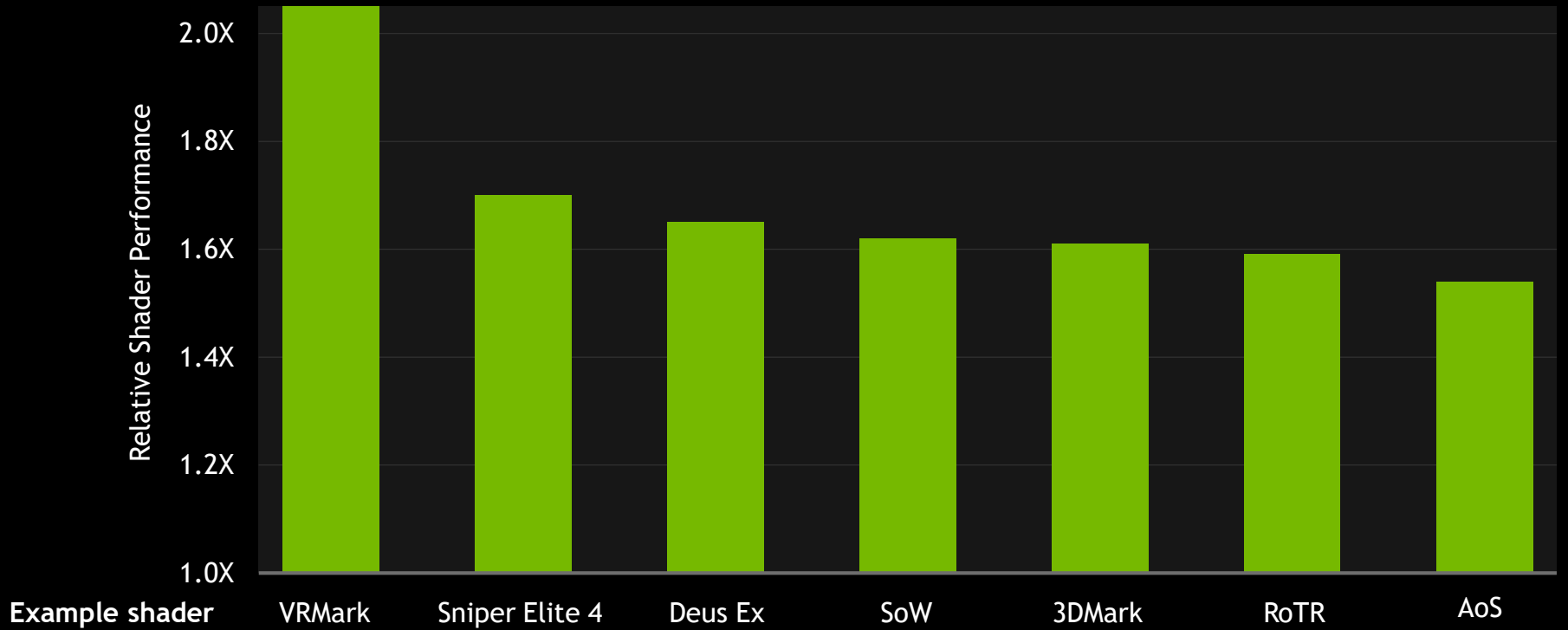
```
…
UIADD3          UR13, UR9, 0x300001, URZ
ULDC.64         UR20, [UR6 + 0x18], !UP7
UIADD3          UR6, UR8, UR10, URZ
UIADD3          UR8, UR9, 0x300002, URZ
FSETP.NEU.FTZ.AND P1, PT, R15, cx[UR20][0x64], PT
ULOP3.LUT       UR12, UR13, 0xfffff, URZ, 0xc0, !UP7
…
```





TURING SHADING PERFORMANCE VS PASCAL

>50% Improved Performance per Core



NVIDIA TURING GPU – NEW TENSOR CORE

Turing Tensor Core for Real-time Inference

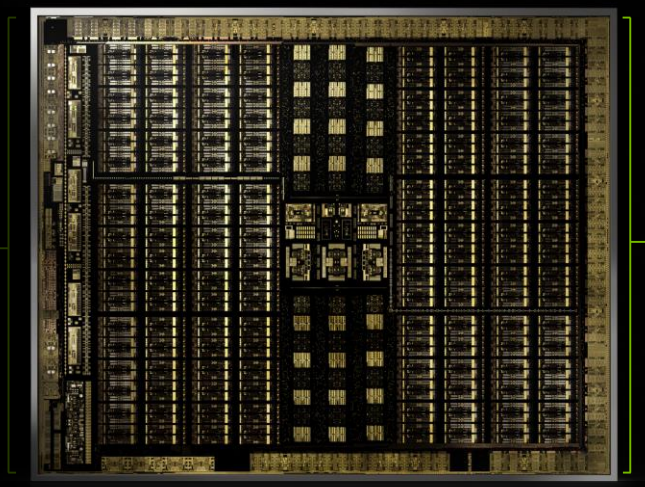
Turing SM

14 TFLOPS + 14 TIPS

Concurrent FP & INT

Enhanced L1 cache

Uniform datapath & RF

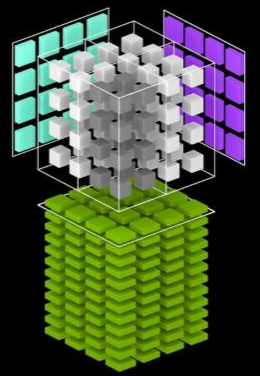


Tensor Core

114 TFLOPS FP16

228 TOPS INT8

455 TOPS INT4



RT Core

First Ray Tracing GPU

10 Giga Rays/sec

Ray Triangle Intersection

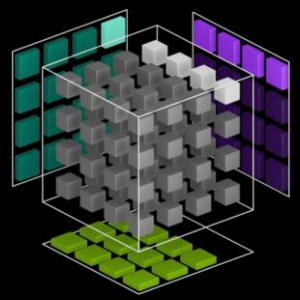
BVH Traversal

TENSOR CORE

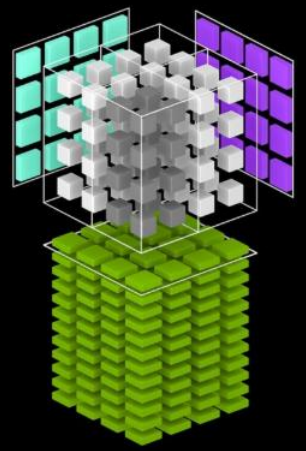
Breakthrough Acceleration for Computation of Matrix Multiplies

$$\begin{pmatrix} A_{0,0} & A_{0,1} & A_{0,2} & A_{0,3} \\ A_{1,0} & A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} \\ A_{3,0} & A_{3,1} & A_{3,2} & A_{3,3} \end{pmatrix} \begin{pmatrix} B_{0,0} & B_{0,1} & B_{0,2} & B_{0,3} \\ B_{1,0} & B_{1,1} & B_{1,2} & B_{1,3} \\ B_{2,0} & B_{2,1} & B_{2,2} & B_{2,3} \\ B_{3,0} & B_{3,1} & B_{3,2} & B_{3,3} \end{pmatrix} + \begin{pmatrix} C_{0,0} & C_{0,1} & C_{0,2} & C_{0,3} \\ C_{1,0} & C_{1,1} & C_{1,2} & C_{1,3} \\ C_{2,0} & C_{2,1} & C_{2,2} & C_{2,3} \\ C_{3,0} & C_{3,1} & C_{3,2} & C_{3,3} \end{pmatrix}$$

PASCAL



TURING TENSOR CORES

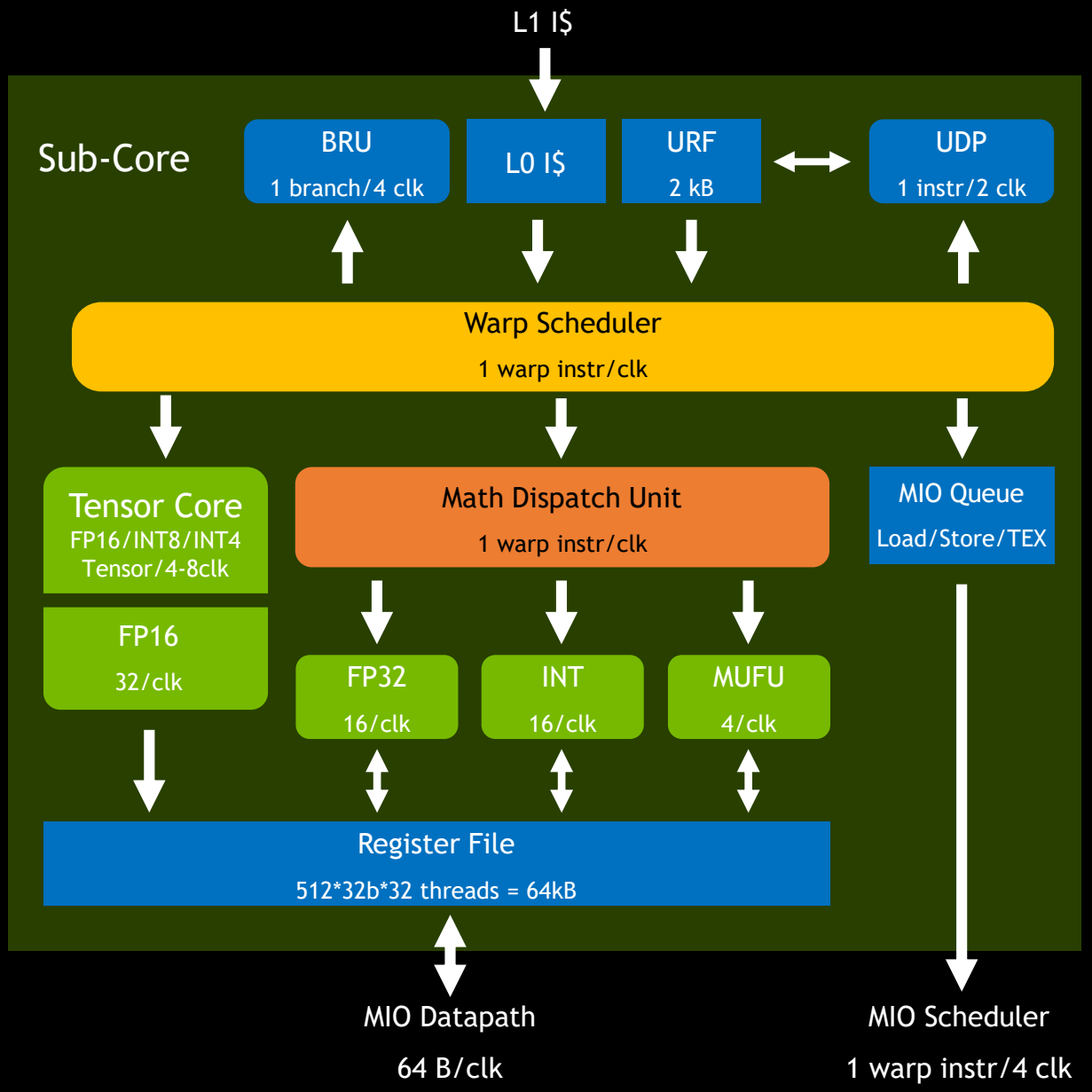


114 TFLOPS FP16

228 TOPS INT8

455 TOPS INT4

**GTX 2080 Ti*



TENSOR CORE

Breakthrough Acceleration for Computation of Matrix Multiplies

Multi-thread collaborative matrix math operation

- Sharing operands across threads saves RF and shared memory BW

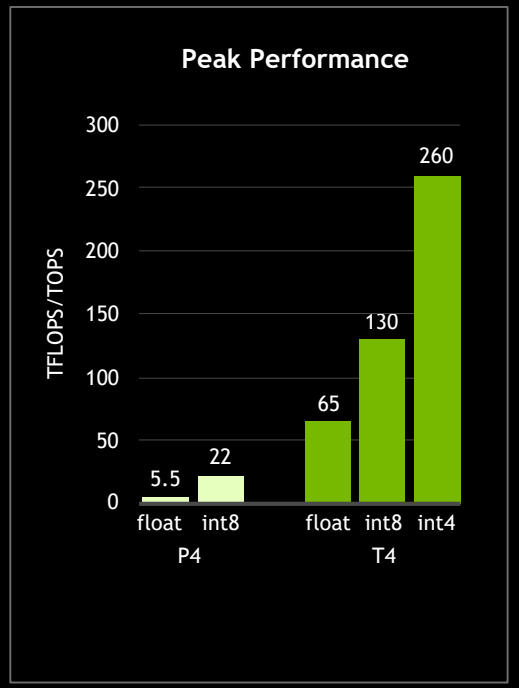
Fine-grained integration inside SM

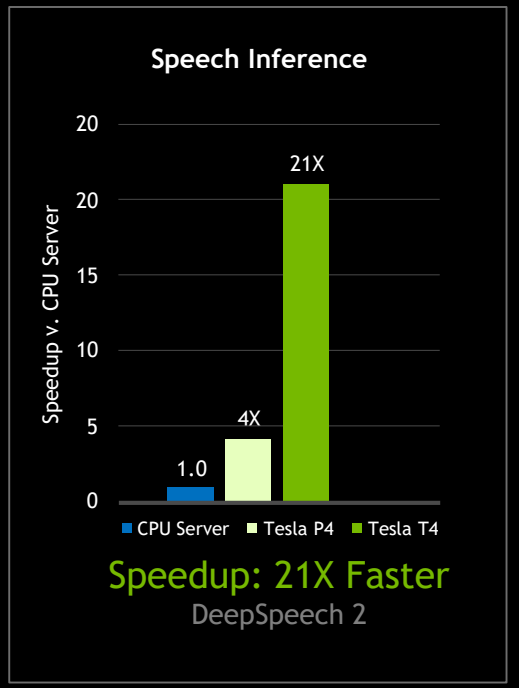
- Provides maximum algorithmic flexibility
  - Different activation functions, Batch norm variants, etc.
- Leverages huge storage capacity and BW provided by RF and shared mem/L1$

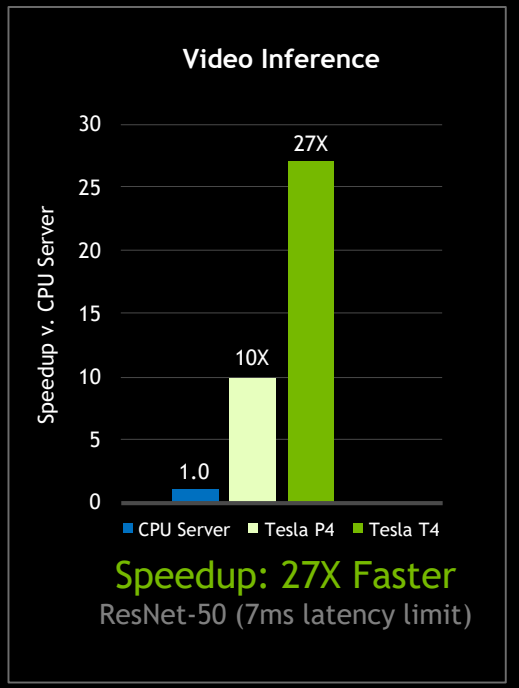
8b & 4b integer support with 32b accumulation for maximum inference performance

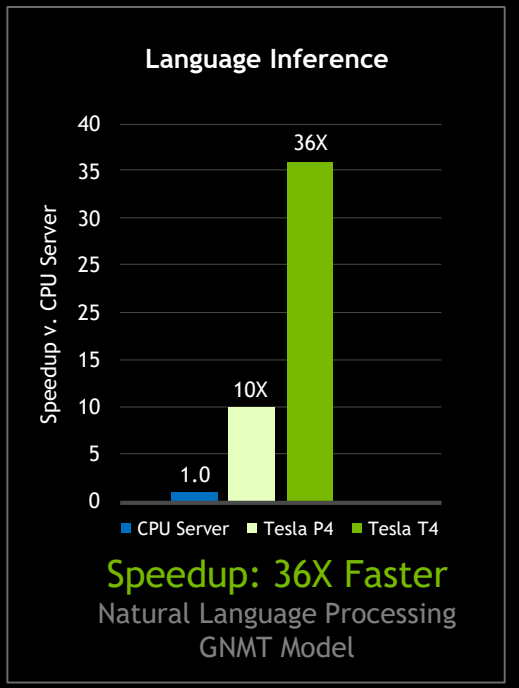
DEEP LEARNING INFERENCE ON TESLA T4

Up to 36X Faster Than CPUs | Accelerates All AI Workloads









ENDLESS POSSIBILITIES OF DEEP LEARNING

Deep Learning Disruption in Gaming and Professional Graphics

DYNAMIC NEURAL GRAPHICS: DLSS

VOICE COMMANDS

STYLE TRANSFER & CONTENT CREATION:
GauGAN

MATERIAL & ART ENHANCEMENT

AI SLOW MOTION VIDEO

FACIAL & CHARACTER ANIMATION

NVIDIA TURING GPU – NEW RT CORE

Turing RTX is 7x Pascal Ray Tracing Performance

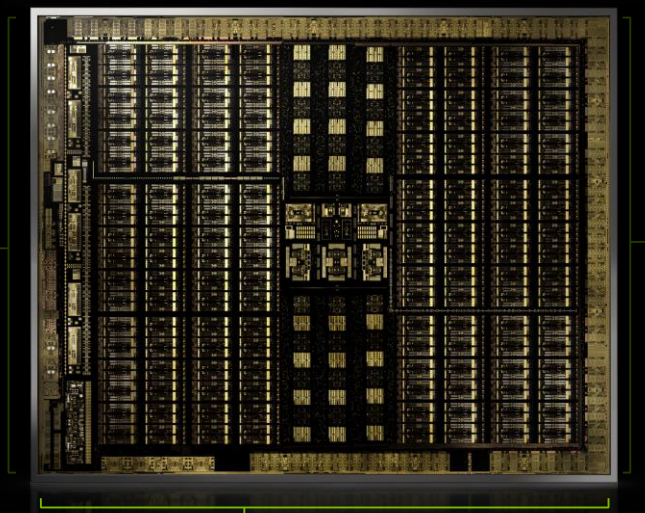
Turing SM

14 TFLOPS + 14 TIPS

Concurrent FP & INT

Enhanced L1 cache

Uniform datapath & RF

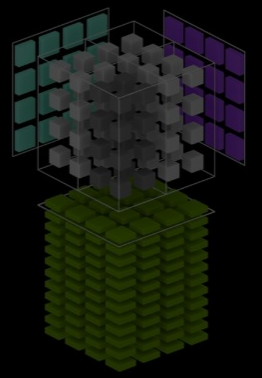


Tensor Core

114 TFLOPS FP16

228 TOPS INT8

455 TOPS INT4



RT Core

First Ray Tracing GPU

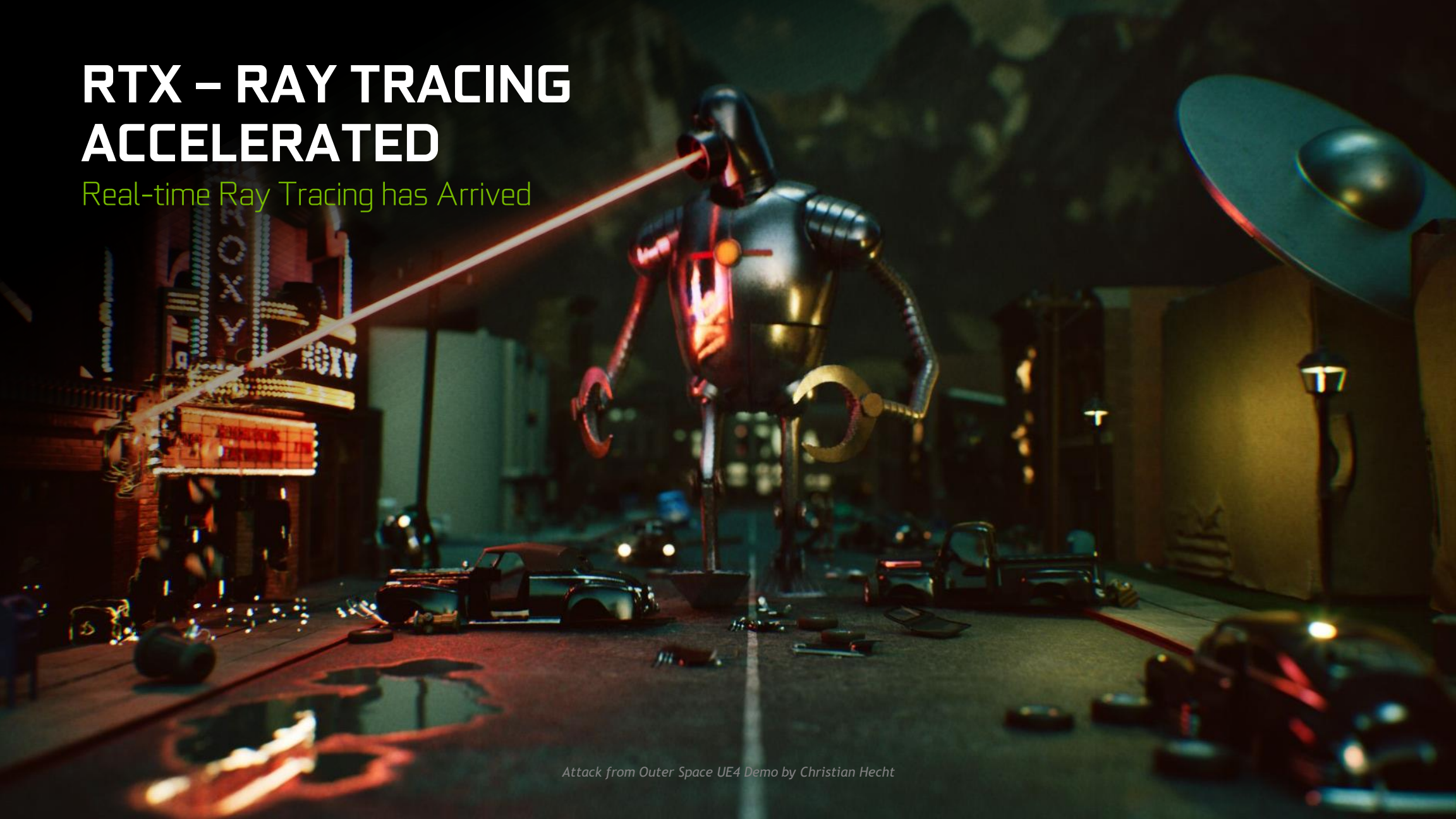
10 Giga Rays/sec

Ray Triangle Intersection

BVH Traversal

RTX – RAY TRACING ACCELERATED

Real-time Ray Tracing has Arrived



Attack from Outer Space UE4 Demo by Christian Hecht

PATH TRACED GLOBAL ILLUMINATION

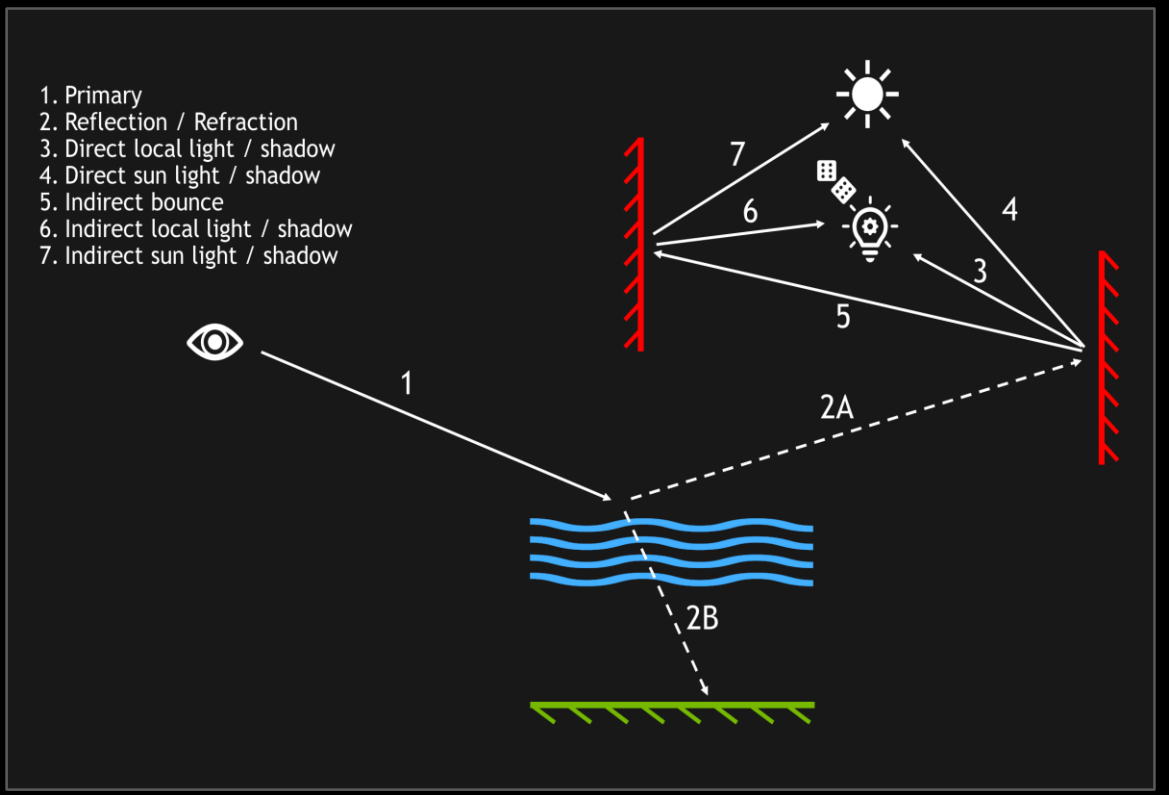
Simulate Physically Based Light Transport by Tracing 'Photons' with Rays

Commonly used for CGI in films

- But many hours to produce final images on CPU

Fundamental building blocks

- Sampling
- Traversal and Intersection
- Material evaluation

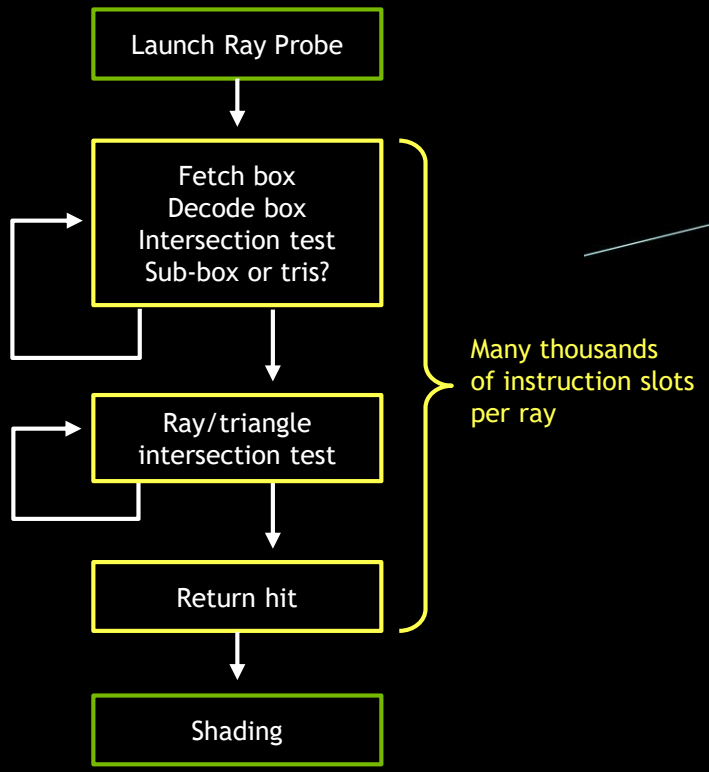


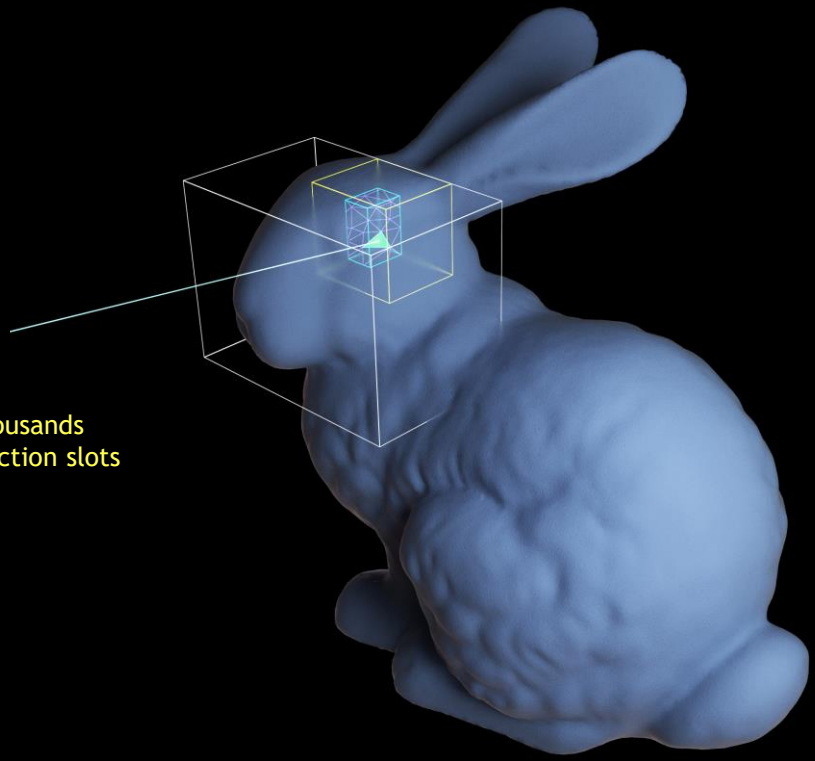
PRE-RTX GPU RAY TRACING

Software Emulation for Ray/Geometry Intersection Search

Pascal SM

Shaders

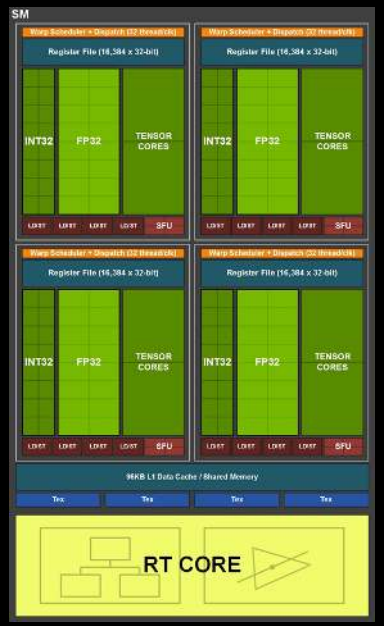




TURING RAY TRACING WITH RT CORES

Hardware Acceleration Replaces Software Emulation

Turing SM



Shaders

Launch Ray Probe

Shading

(E.g. material evaluation, importance sampling, denoising, custom intersection, etc.)

RT Core

Fetch box
Decode box
Intersection test
Sub-box or tris?

Ray/triangle intersection test

Return hit

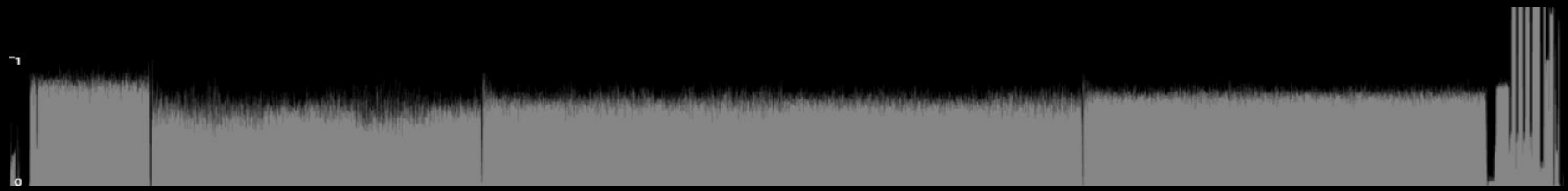
Box Intersection Evaluators

Triangle Intersection Evaluators

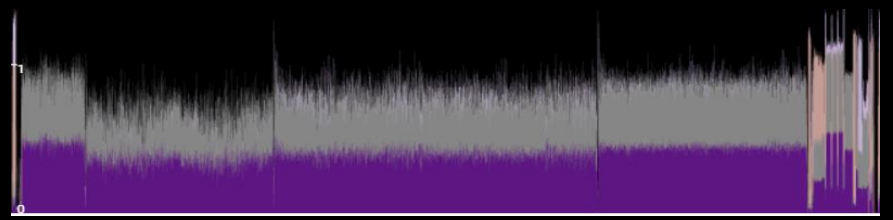
ONE QUAKE II RTX FRAME

Breakthrough Acceleration Enables Real-time Path Tracing

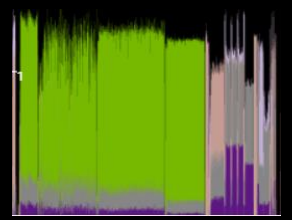
PASCAL
GTX 1080Ti
202 ms
5 fps



TURING
RTX 2080
NO RT CORES
97 ms
10 fps



TURING RTX
RTX 2080
RT CORES ON
29 ms
34 fps



7x speedup

FP32 Cores | INT32 Cores | RT Cores | Other Graphics | Memory

REAL-TIME RAY TRACING IS HERE

GAMES

Most Anticipated Games | Biggest Franchises

ENGINES AND APIs

Support in all Major Game Engines

NVIDIA TURING GPU

Greater Than the Sum of Its Parts

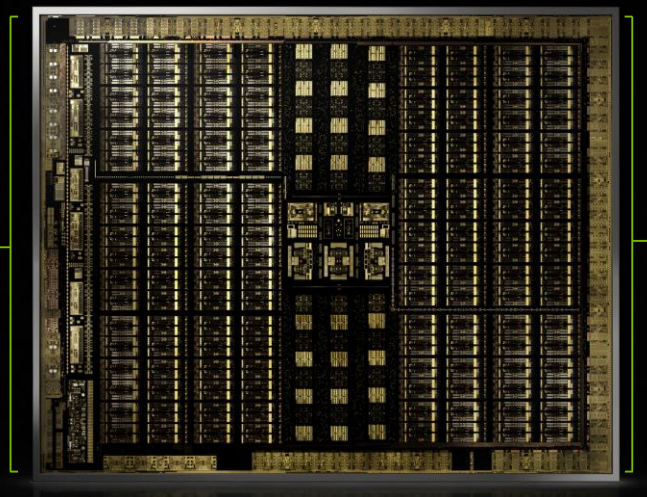
Turing SM

14 TFLOPS + 14 TIPS

Concurrent FP & INT

Enhanced L1 cache

Uniform datapath & RF

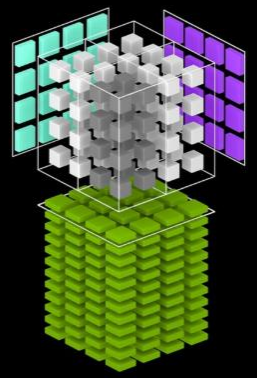


Tensor Core

114 TFLOPS FP16

228 TOPS INT8

455 TOPS INT4



RT Core

First Ray Tracing GPU

10 Giga Rays/sec

Ray Triangle Intersection

BVH Traversal

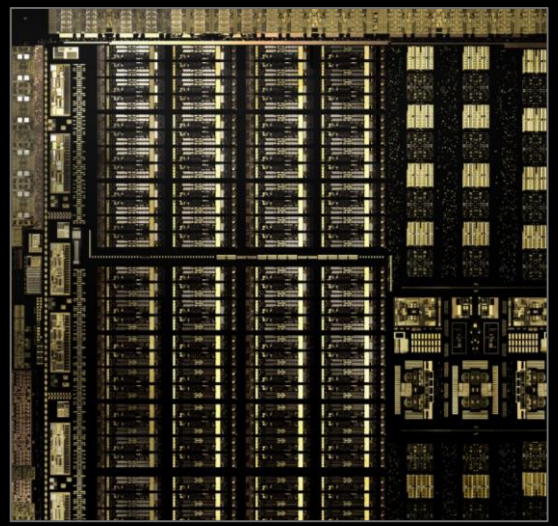
PROFESSIONAL RENDERING ON QUADRO RTX

SM + RTCore + TensorCore = Accelerated Ray Tracing and AI Denoising

NVIDIA TURING GPU

Evolved for Efficiency and Breakthrough Acceleration

TURING GPU

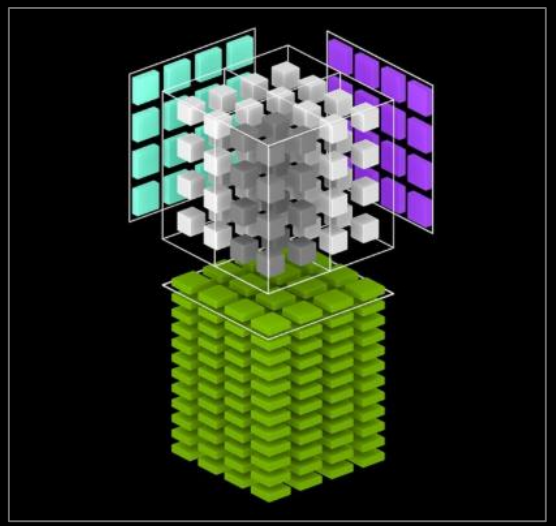


Next Gen Graphics Realized

SM CORE

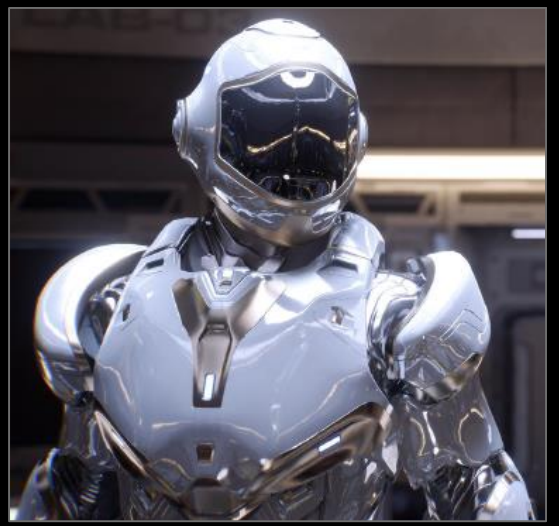
>1.5x Faster SM

TENSOR CORE



Real-time Inference

RT CORE



>7x Faster Ray Tracing

More Turing features: GDDR6, Variable Rate Shading, Mesh Shading, Post-L2 Cache Data Compression, NVLINK Connectivity, USBC, and many more...

THANK YOU - QUESTIONS?


NVIDIA
GEFORCE
RTX