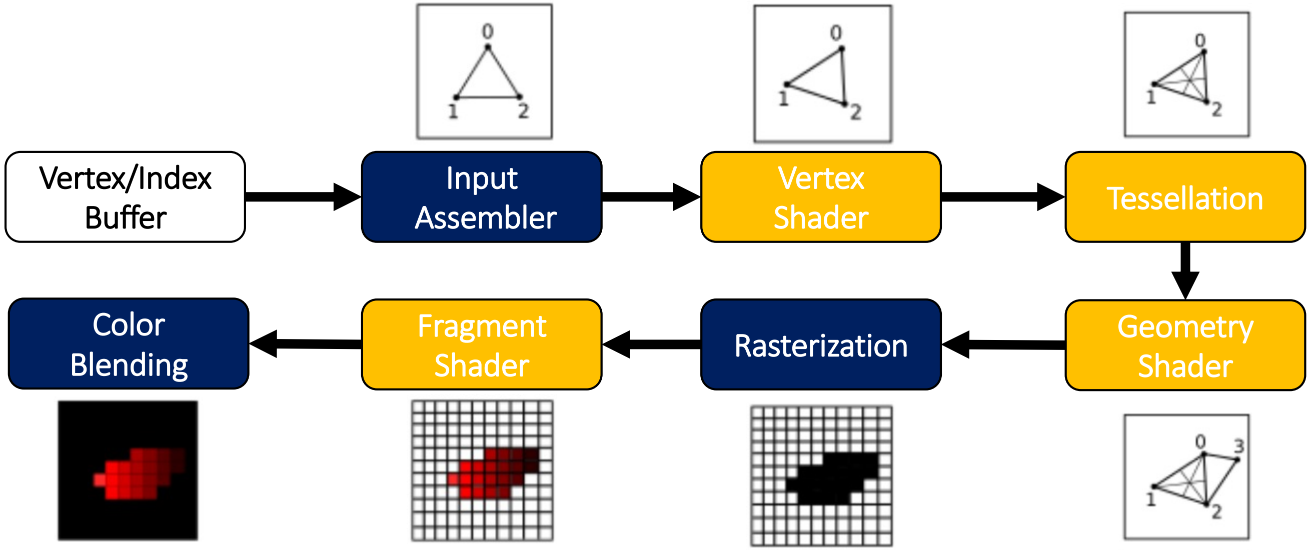
TOGETHER.
TOMORROW.
EWHA

Maximizing Thread-Level Parallelism on GPUs

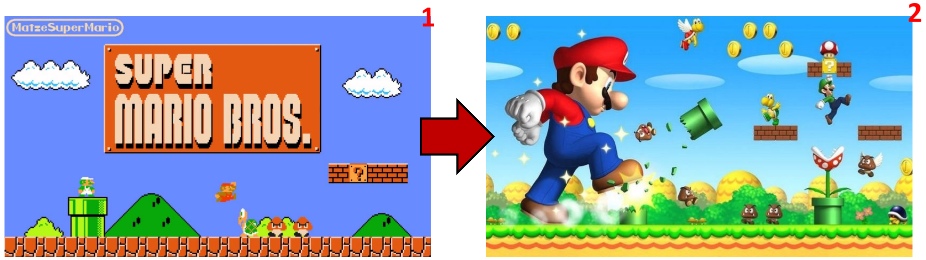
Yoon, Myung Kuk (윤명국)
Department of Computer Science and Engineering

이화여자대학교
EWHA WOMANS UNIVERSITY

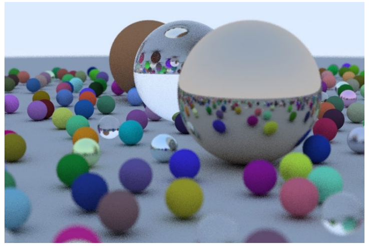
Purpose of GPUs



[Graphics Pipeline]



[Super Mario]

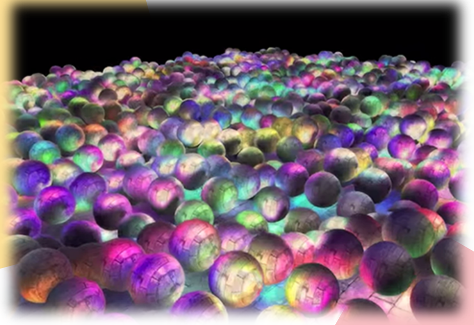


[Ray Tracing]

History of GPUs

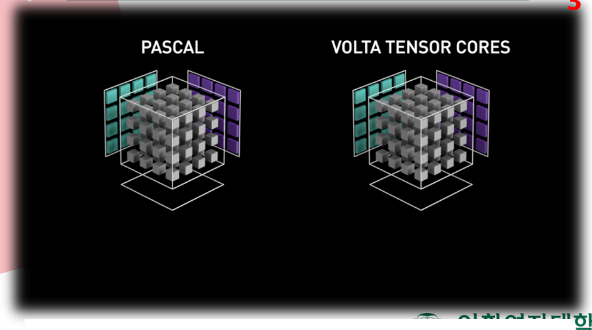
[Fixed Functions]

Improving Image Quality



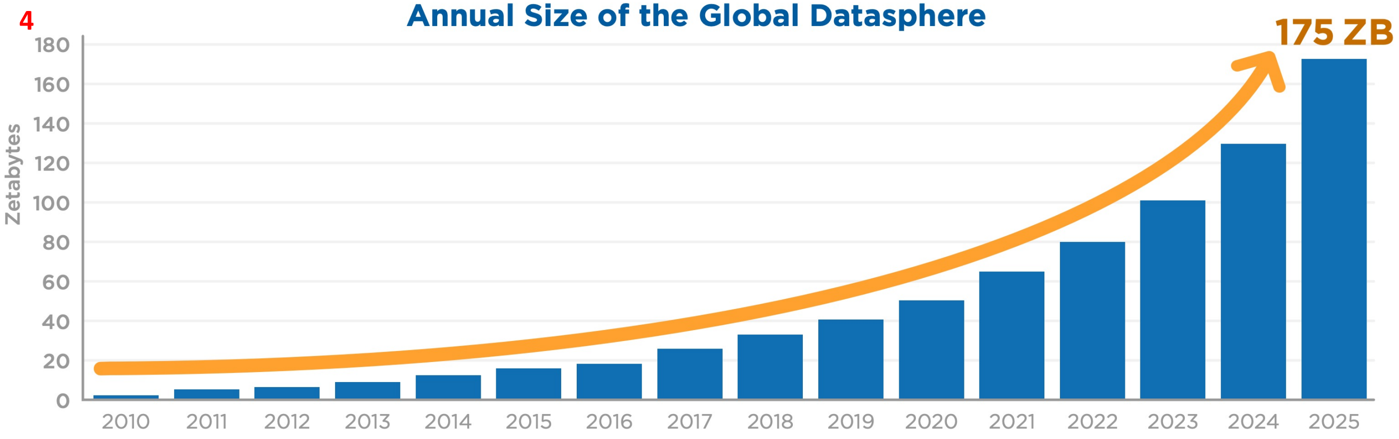
[Programmable Stages]

High Throughput Computing

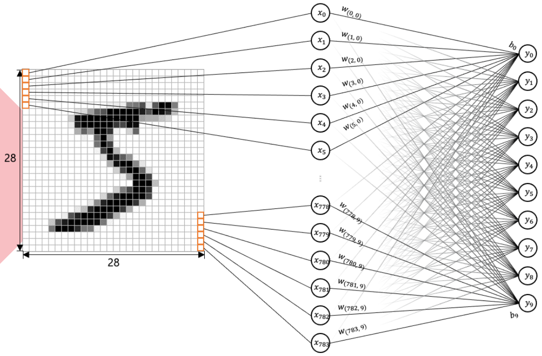


[General Purpose Applications]

New Era of Big Data and AI



[Growing Data Size]

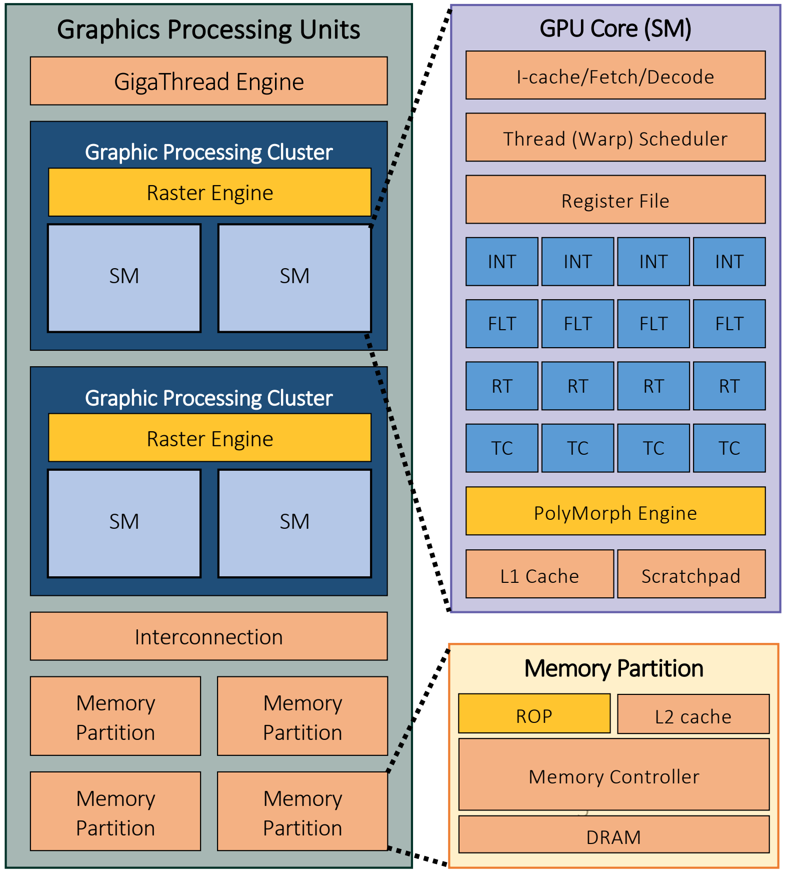


[Applications] **[Machine Learning Algorithm]**

TOGETHER.
TOMORROW.
EWHA

이화여자대학교
EWHA WOMANS UNIVERSITY

GPU Architecture

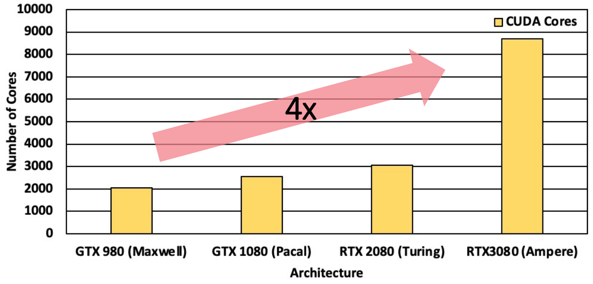


[GPU Architecture]

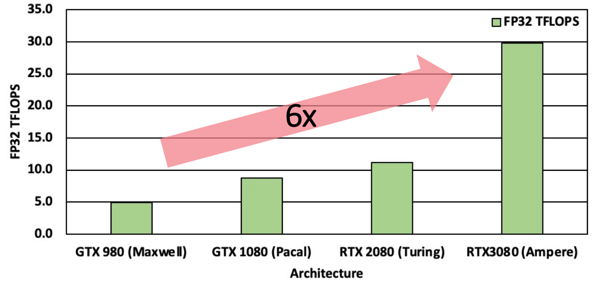
🟧 Controller + Memory

🟨 Fixed Function for Graphics

🟦 Processing Units for Graphics and General Applications



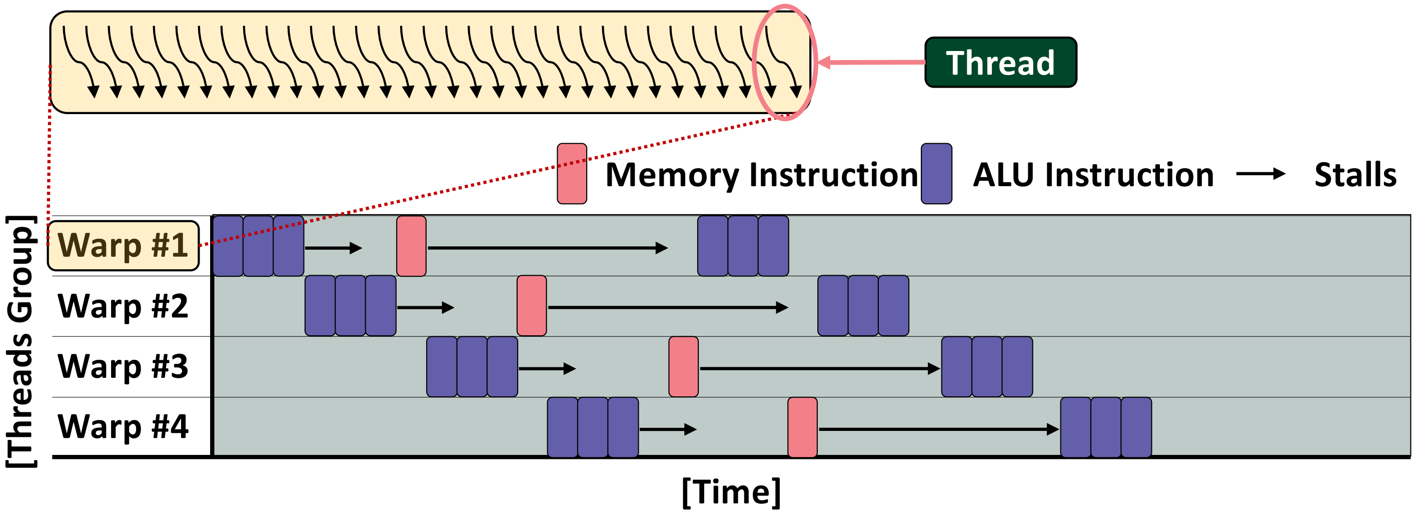
[GPU Cores]



[GPU Throughput]

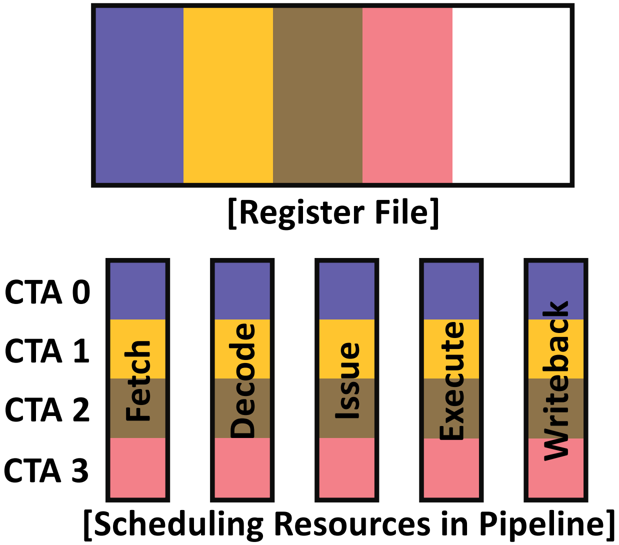
High Thread Level Parallelism (TLP)

- Warp/Wavefront: A set of 32/64 threads within a thread block
- Advantage of high TLP: hiding stalls from a warp by other warps' executions

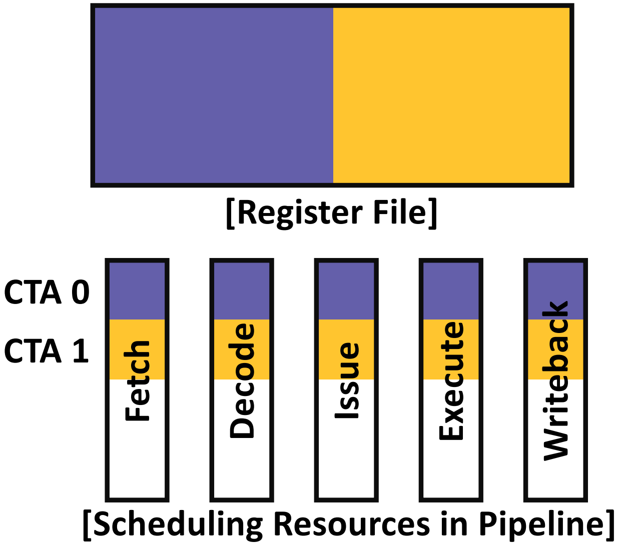


TLP Limiting Factors

- How to decide the number of threads (CTAs) assigned to SM?
  - **Scheduling limit:** thread counts
  - **Capacity limit:** register file size and shared memory size



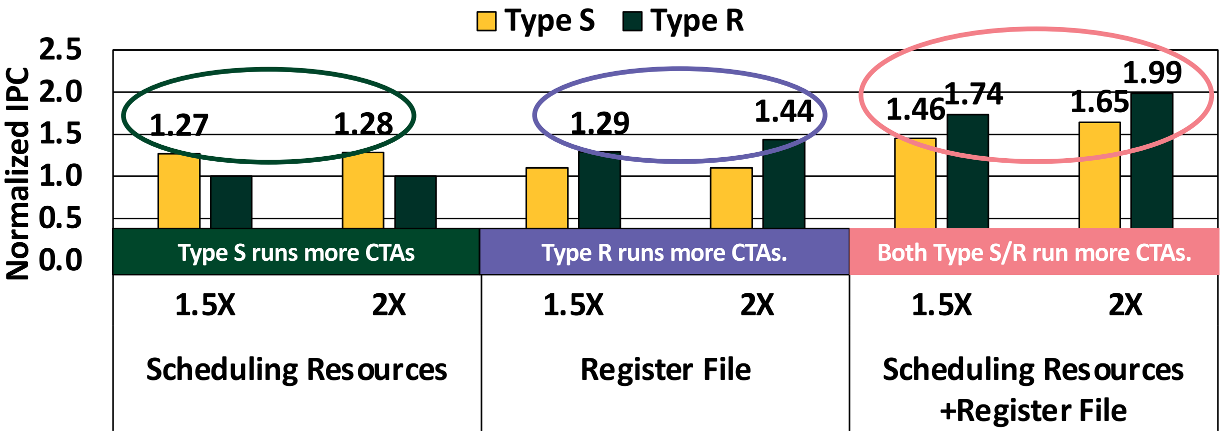
Scheduling Limit



Capacity Limit

No Scheduling & Capacity Limits

- **What if no scheduling and capacity limits on GPUs?**
  - Hiding stalls from a warp by other warps' executions



[Performance Improvement]

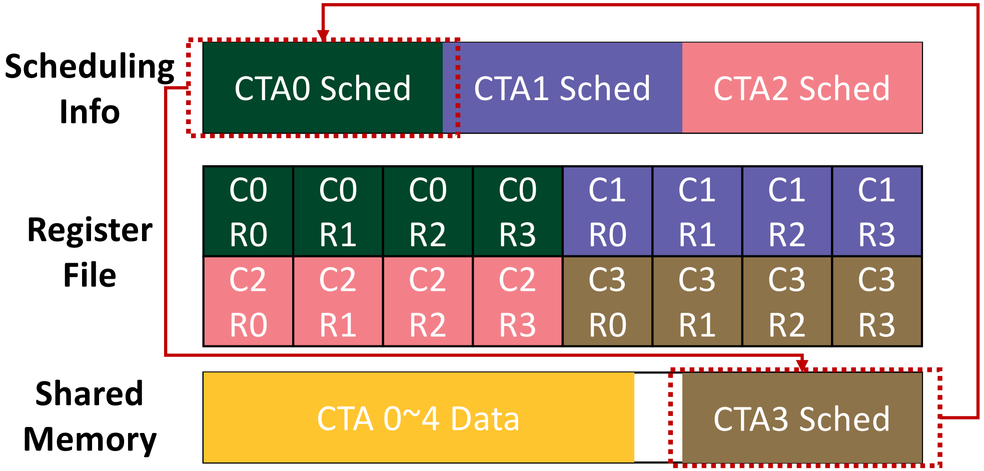
Introducing additional scheduling resources and register file
<u>increases hardware complexity and costs</u>

Virtual Thread Architecture[6]

- **Goal: Dispatching more threads onto GPUs to fill up the register file and shared memory without increasing the scheduling limits**

- **Active CTAs: issuable CTAs**
  - Up to the scheduling limit
- **Inactive CTAs: not issuable CTAs**
  - Up to the capacity limit

- **CTA scheduling (context) information**
  - Warp ID (Virtual Warp ID)
  - SIMT stack (PC, RPC, and active mask)
  - CTA identifiers



Scheduling Info

CTA0 Sched | CTA1 Sched | CTA2 Sched

Register File

C0 R0	C0 R1	C0 R2	C0 R3	C1 R0	C1 R1	C1 R2	C1 R3
C2 R0	C2 R1	C2 R2	C2 R3	C3 R0	C3 R1	C3 R2	C3 R3

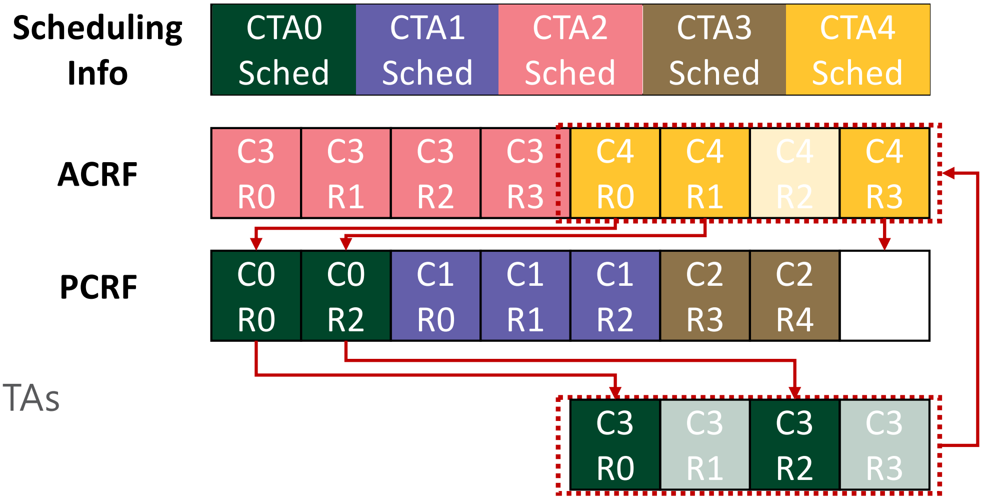
Shared Memory

CTA 0~4 Data | CTA3 Sched

FineReg Management[7]

- **Goal: Dispatching more threads onto GPUs by maintaining only the small portion of the live registers**

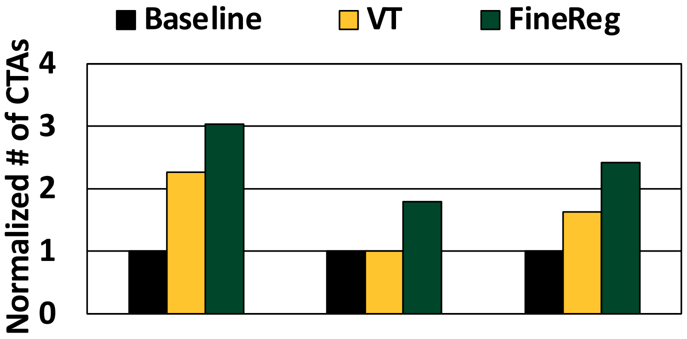
- **ACRF: Active CTA Register File**
  - Same as original GPU register file
  - Keeps all registers of active CTAs
- **PCRF: Pending CTA Register File**
  - Backup CTA register storage
  - Keeps only the live registers of pending CTAs



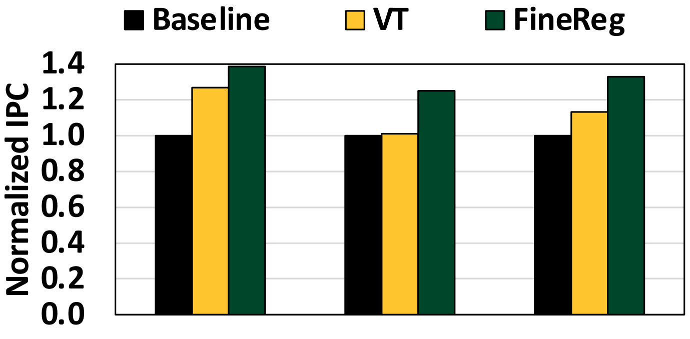
- **Register Liveness**
  - The compiler generates the list of live registers of every instruction

Evaluation

- **Impact on Thread-Level Parallelism**
  - FineReg: 2.42x more threads than the baseline
- **Performance Impact**
  - FineReg: 32.8% IPC (Instructions Per Cycle) improvement



[Number of Concurrent CTAs]



[Normalized IPC]

Conclusion

- **Advantage of high TLP**
  - hiding stalls from a warp by other warps' executions
- **TLP Limiting Factors**
  - **Scheduling limit:** thread counts
  - **Capacity limit:** register file size and shared memory size limits

- **Virtual Thread Architecture**[6]
  - Dispatching more threads onto GPUs to fill up the register file and shared memory without increasing the scheduling limits
- **FineReg Management**[7]
  - Dispatching more threads onto GPUs by maintaining only the small portion of the live registers

Thank You!

- Yoon, Myung Kuk (윤명국)
- Department of Computer Science and Engineering
- E-Mail: myungkuk.yoon@ewha.ac.kr
- Homepage: http://ip-cal.ewha.ac.kr

References

1. Old Super Mario, https://www.youtube.com/watch?v=GlwX5q_Y1-0
2. 3D Super Mario, https://www.bbc.com/news/technology-53402067
3. “NVIDIA TENSOR CORES,” https://www.nvidia.com/en-us/data-center/tensor-cores/
4. The Digitization of the World From Edge to Core, https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf
5. Up to Speed on Deep Learning in Medical Imaging, https://medium.com/the-mission/up-to-speed-on-deep-learning-in-medical-imaging-7ff1e91f6d71
6. Yoon et al., Virtual Thread: Maximizing Thread-Level Parallelism beyond GPU Scheduling Limit, ISCA 2016
7. Oh et al, FineReg: Fine-Grained Register File Management for Augmenting GPU Throughput, MICRO 2018