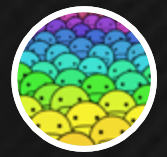
AI 芯片 – GPU 详解

Tensor Core 原理



ZOMI

Talk Overview

1. **硬件基础**
   - GPU 工作原理
   - GPU AI编程本质
2. **英伟达 GPU 架构**
   - GPU基础概念
   - 从 Fermi 到 Volta 架构
   - Turing 到 Hopper 架构
   - Tensor Code 和 NVLink 详解
3. **GPU 图形处理**
   - GPU 逻辑模块划分
   - 算法到 GPU 硬件
   - GPU 的软件栈
   - 图形流水线基础
   - 流水线不可编译单元
   - 光线跟踪流水线

Talk Overview

I. 基本工作原理

- Conv and TC Relationship - 卷积与Tensor Core关系
- TC Basic Principles - Tensor Core的基本原理
- TC Architecture Evolution - Tensor Core架构演进

Talk Overview

1. 基本工作原理

- Conv and TC Relationship - 卷积与Tensor Core关系
- TC Basic Principles - Tensor Core的基本原理
- TC Architecture Evolution - Tensor Core架构演进

2. 深入Tensor Core

- XXX
- XXX

NVIDIA GPU架构发展

架构名称	Fermi	Kepler	Maxwell	Pascal	Volta	Turing	Ampere	Hopper
中文名字	费米	开普勒	麦克斯韦	帕斯卡	伏特	图灵	安培	赫柏
发布时间	2010	2012	2014	2016	2017	2018	2020	2022
核心参数	16个SM，每个SM包含32个CUDA Cores，一共512 CUDA Cores	15个SMX，每个SMX包括192个FP32+64个FP64 CUDA Cores	16个SM，每个SM包括4个处理块，每个处理块包括32个CUDA Cores +8个LD/ST Unit + 8 SFU	GP100有60个SM，每个SM包括64个CUDA Cores，32个DP Cores	80个SM，每个SM包括32个FP64 +64 Int32+64 FP32+8个Tensor Cores	102核心92个SM，SM重新设计，每个SM包含64个Int32+64个FP32+8个Tensor Cores	108个SM，每个SM包含64个FP32+64个INT32+32个FP64+4个Tensor Cores	132个SM，每个SM包含128个FP32+64个INT32+64个FP64+4个Tensor Cores
特点&优势	首个完整GPU计算架构，支持与共享存储结合的Cache层次GPU架构，支持ECC GPU架构	游戏性能大幅提升，首次支持GPU Direct技术	每组SM单元从192个减少到每组128个，每个SMM单元拥有更多逻辑控制电路	NVLink第一代，双向互联带宽160 GB/s，P100拥有56个SM HBM	NVLink2.0，Tensor Cores第一代，支持AI运算	Tensor Core2.0，RT Core第一代	Tensor Core3.0，RT Core2.0，NVLink3.0，结构稀疏性矩阵MIG1.0	Tensor Core4.0，NVlink4.0，结构稀疏性矩阵MIG2.0
纳米制程	40/28nm 30亿晶体管	28nm 71亿晶体管	28nm 80亿晶体管	16nm 153亿晶体管	12nm 211亿晶体管	12nm 186亿晶体管	7nm 283亿晶体管	4nm 800亿晶体管
代表型号	Quadro 7000	K80 K40M	M5000 M4000 GTX 9XX系列	P100 P6000 TTX1080	V100 TiTan V	T4，2080TI RTX 5000	A100 A30系列	H100

NVIDIA GPU架构发展

架构名称	Volta	Turing	Ampere	Hopper
中文名字	伏特	图灵	安培	赫柏
发布时间	2017	2018	2020	2022
核心参数	80个SM，每个SM包括32个FP64+64 Int32+64 FP32+8个Tensor Cores	102核心92个SM，SM重新设计，每个SM包含64个Int32+64个FP32+8个Tensor Cores	108个SM，每个SM包含64个FP32+64个INT32+32个FP64+4个Tensor Cores	132个SM，每个SM包含128个FP32+64个INT32+64个FP64+4个Tensor Cores
特点&优势	NVLink2.0，Tensor Cores第一代，支持AI运算	Tensor Core2.0，RT Core第一代	Tensor Core3.0，RT Core2.0，NVLink3.0，结构稀疏性矩阵MIG1.0	Tensor Core4.0，NVlink4.0，结构稀疏性矩阵MIG2.0
纳米制程	12nm 211亿晶体管	12nm 186亿晶体管	7nm 283亿晶体管	4nm 800亿晶体管
代表型号	V100 TiTan V	T4，2080TI RTX 5000	A100 A30系列	H100

卷积计算

主页 动态 投稿 122 合集和列表 19 收藏 4 订阅 设置 搜索视频、动态

【推理引擎】Kernel优化

播放全部

AI 网络模型中，看到的是算子；但是在推理引擎实际执行的是具体的 Kernel，而推理引擎中 CNN 尤其 Kernel 优化尤为重要。

默认排序 升序排序 编辑

Kernel优化架构介绍！【推理引擎】Kernel优化第01篇

2362 2-12

卷积优化：卷积操作基础原理！【推理引擎】Kernel优化

2234 2-26

卷积优化：Im2Col算法和组合优化算法【推理引擎】Kernel

1542 3-1

卷积优化：Winograd算法【推理引擎】Kernel优化第04

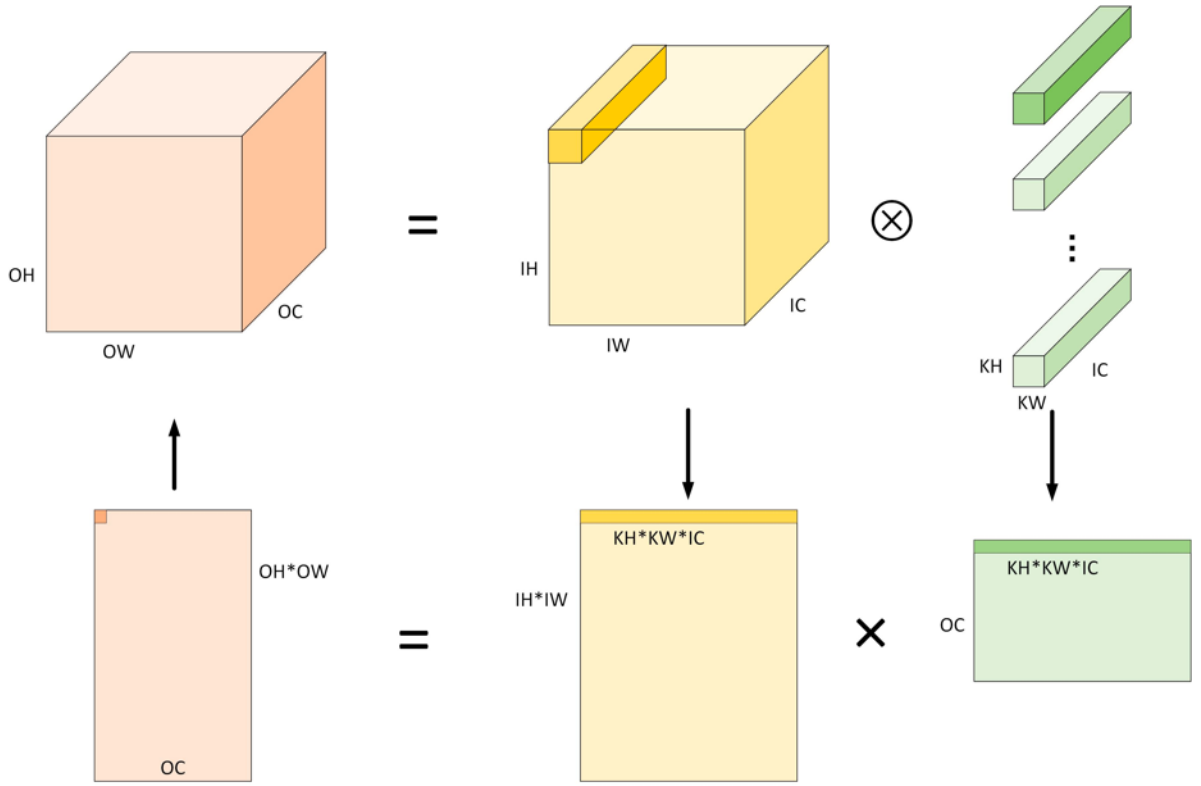
1657 3-2

QNNPack之间接优化算法【推理引擎】Kernel优化第05

1104 3-3

CNN vs GEMM

- 卷积神经网络 CNN 包含许多卷积层。算法上通过数据重排，完成 Im2col 的操作之后会得到一个输入矩阵，卷积 Weights 可以转换为一个矩阵，因此卷积的计算就可以转换为两个矩阵相乘的求解，得到最终的卷积计算结果。GEMM 计算可以被成批地放在一起，作为单个大型矩阵乘法运算运行。



Tensor Core
基本原理

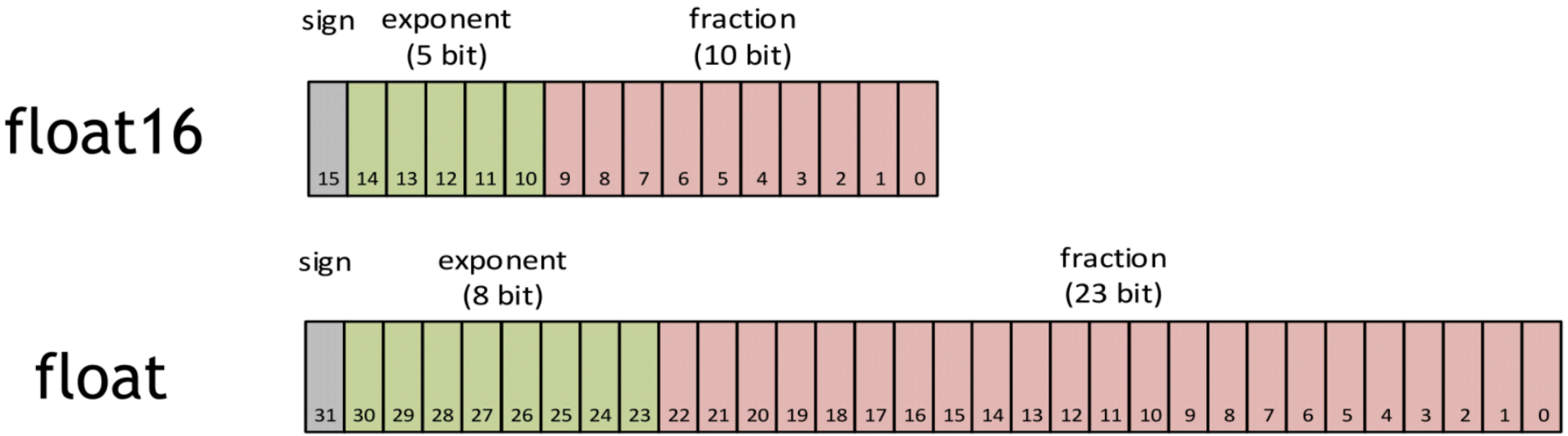
Question?

- 什么是混合精度？

- 网络层面既有 FP16 又有 FP32。

混合精度

- 混合精度是指在底层硬件算子层面，使用半精度（FP16）作为输入和输出，使用全精度（FP32）进行中间结果计算从而不损失过多精度的技术。这个底层硬件层面其实指的就是 Tensor Core，所以 GPU 上有 Tensor Core 是使用混合精度训练加速的必要条件。



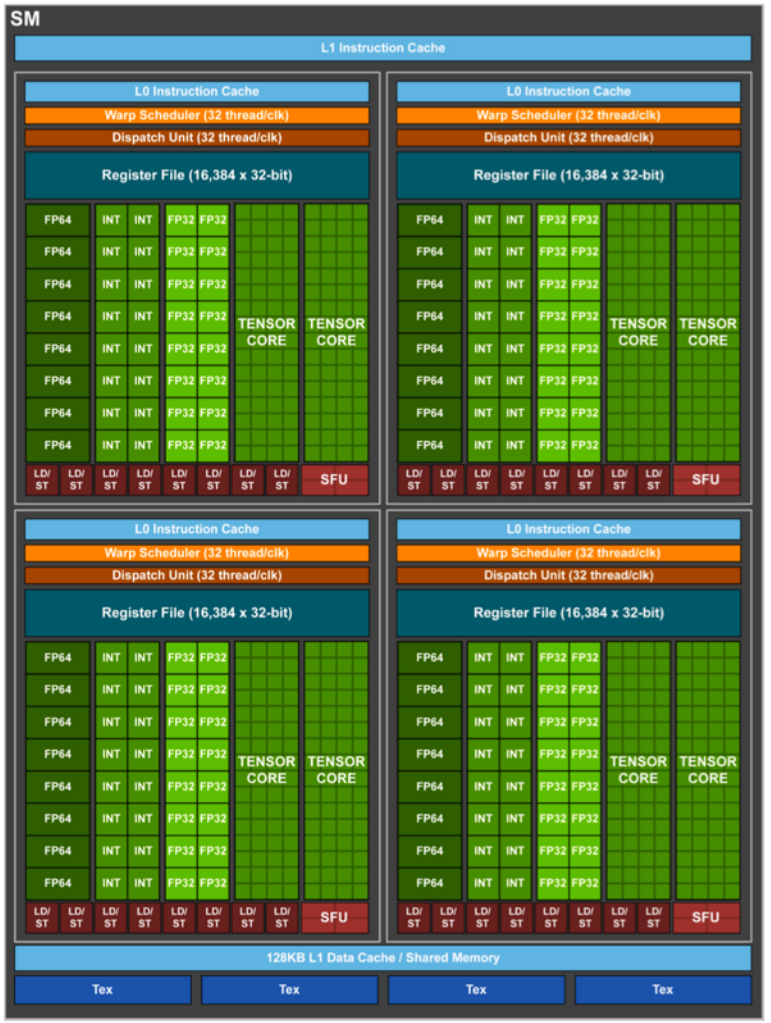
Volta 架构第一代 Tensor Core

CUDA Core:

- GPU 并行模式实现深度学习功能过于通用，最常见 Conv/GEMM 操作，依旧要被编码成 FMA，硬件层面还是需要把数据按：~~寄存器-ALU-寄存器-ALU-寄存器~~，方式来回搬运。

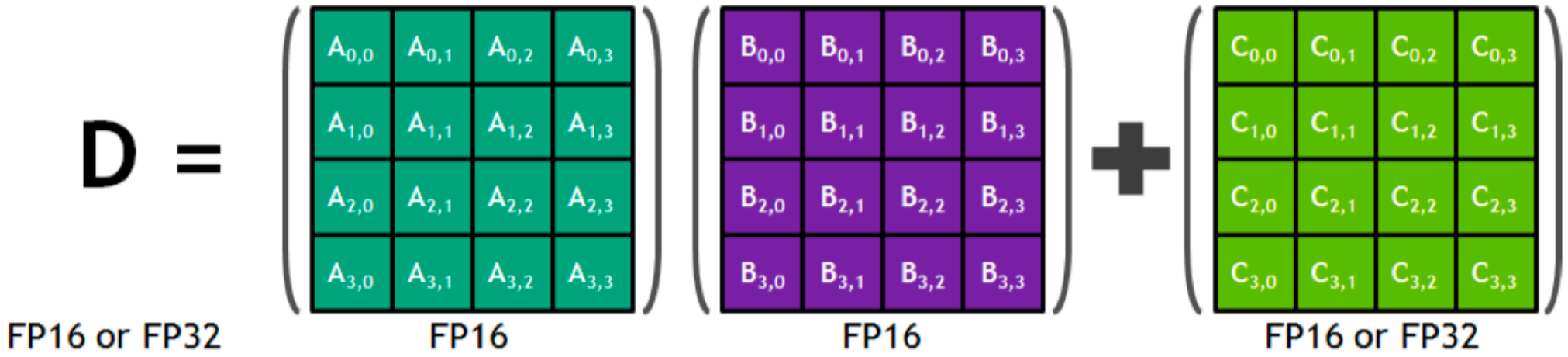
Tensor Core:

- V100 Tensor Core 提供可编程矩阵乘法和累加单元（matrix-multiply-and-accumulate units），可为 AI 训练和推理提供125 Tensor TFLOPS 算力。V100 包含640个Tensor内核：每个SM 8个。



Volta 架构第一代 Tensor Core

- 每个 Tensor Core 每周期能执行 **4x4x4 GEMM**，64 个 FMA。执行运算 **D=A*B+C**，其中A、B、C 和 D 是 4×4 矩阵。**矩阵乘法**输入 A 和 B 是 FP16 矩阵，而**累加矩阵** C 和 D 可以是 FP16或 FP32 矩阵。
- Tensor Core执行融合乘法加法，其中两个4*4 FP16矩阵相乘，然后将结果添加到4*4 FP16或FP32矩阵中，最终输出新的4*4 FP16或FP32矩阵。



Volta 架构第一代 Tensor Core：混合精度

- 混合精度是指在底层硬件算子层面，使用半精度（FP16）作为输入和输出，使用全精度（FP32）进行中间结果计算从而不损失过多精度的技术。这个底层硬件层面其实指的就是 Tensor Core，所以 GPU 上有 Tensor Core 是使用混合精度训练加速的必要条件。

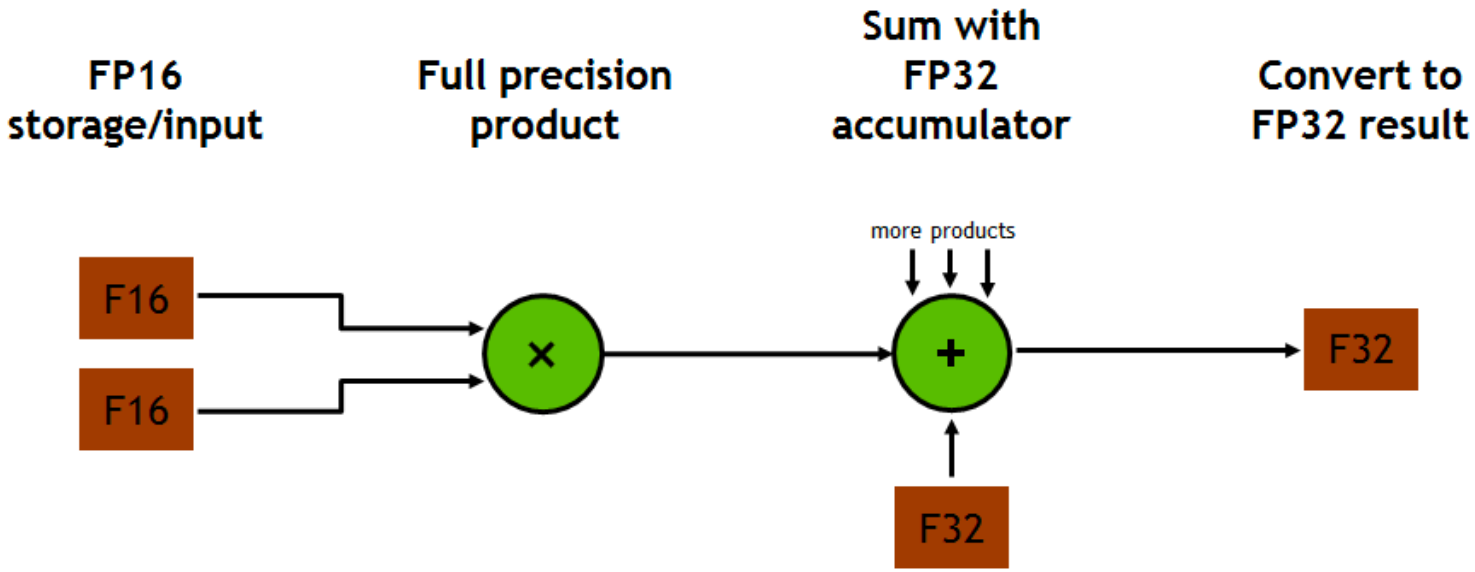
Training values storage	Matrix-Mult Accumulator	Name
FP32	FP32	FP32 training
FP16	FP32	Mixed precision training
FP16	FP16	FP16 training

With mixed or FP16 training, master weights can be FP16 or FP32.

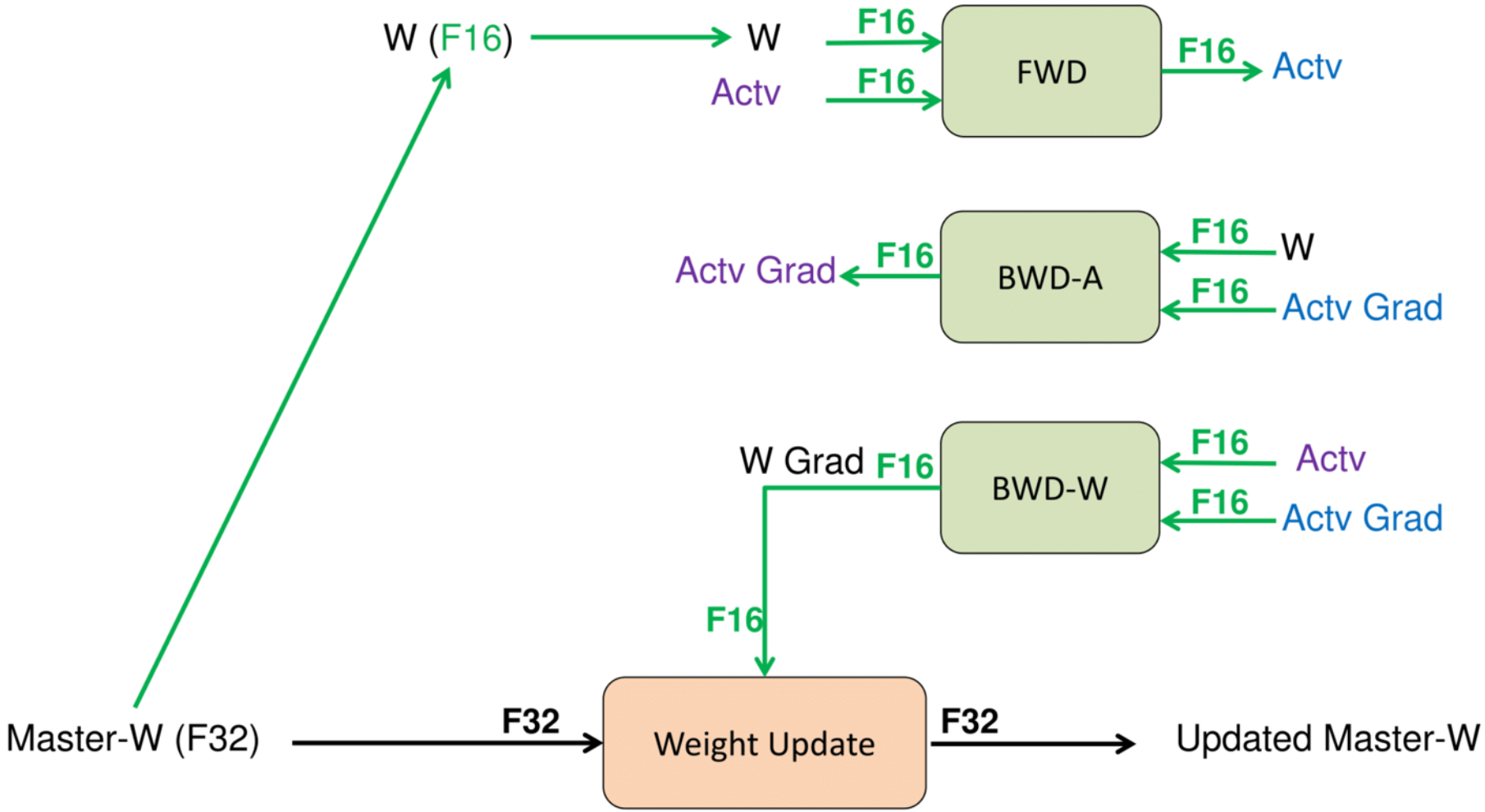
Volta: **Mixed precision training** with **FP32 master weight storage.**

Volta 架构第一代 Tensor Core

- 每个 Tensor Core 每个时钟执行 64 个 FP32 FMA 混合精度运算，SM中8个 Tensor Core，每个时钟周期内总共执行 512 个浮点运算。
- 因此在 AI 应用中， Volta V100 GPU的吞吐量与Pascal P100 GPU相比，每个 SM 的 AI 吞吐量增加了 8 倍，总共增加了12倍。

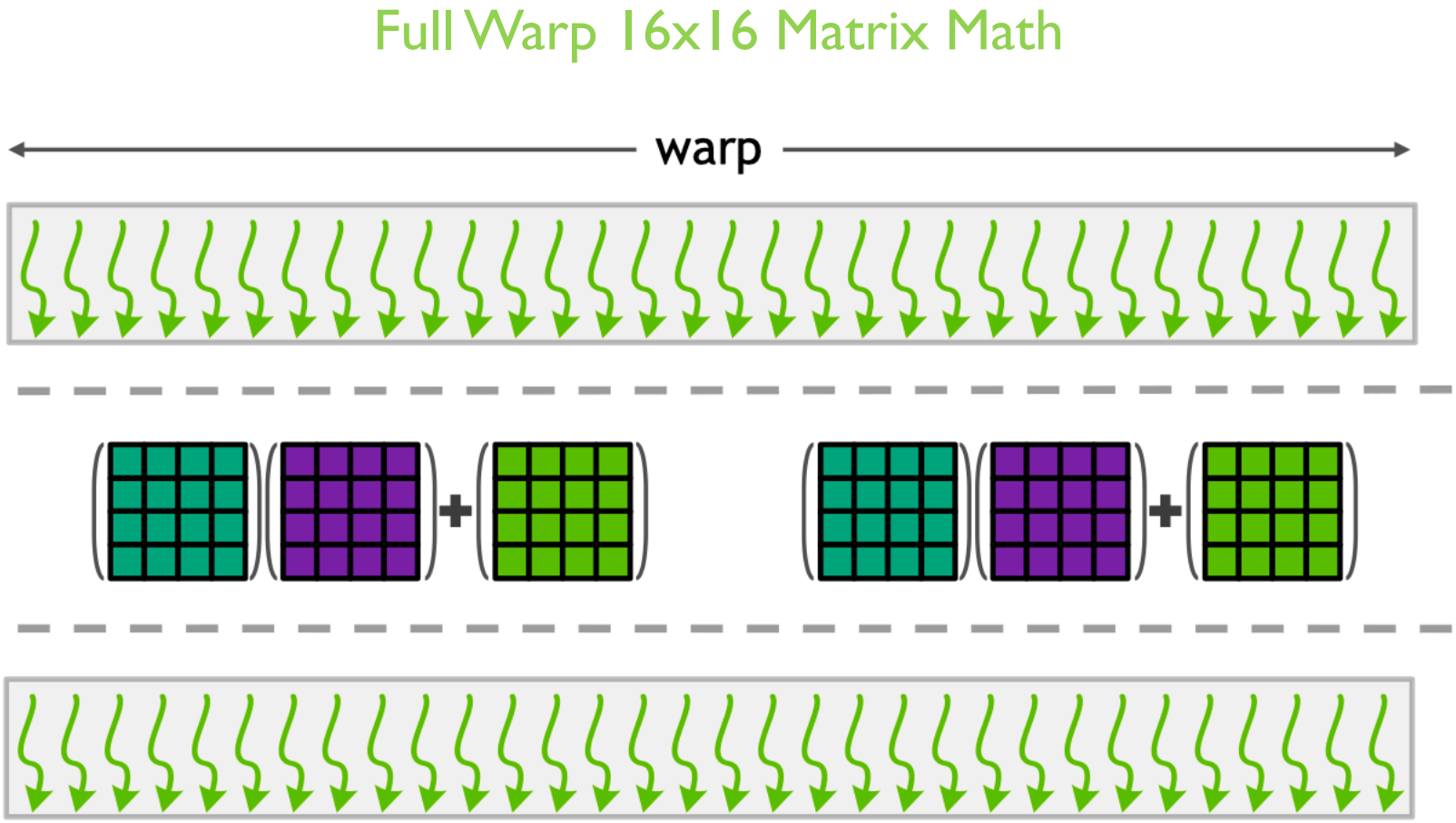


Volta 架构第一代 Tensor Core



Tensor Core & CUDA Programing

张量同步的方式



- Warp 同步操作
- 16x16 的矩阵计算
- 结果存储在不同的 Warp

Volta 架构第一代 Tensor Core

- 程序执行过程中，多个 Tensor Core 可以同时通过线程 wrap 来执行，在一个 Wrap 内的线程可以通过 Tensor Core 来提供16x16x16 的矩阵运算。
- CUDA将 Tensor Core 在 Wrap level 的计算操作通过 **CUDA C++WMMA API** 对外提供，这些C++接口提供了专门的矩阵加载、矩阵乘法和累加以及矩阵存储操作。

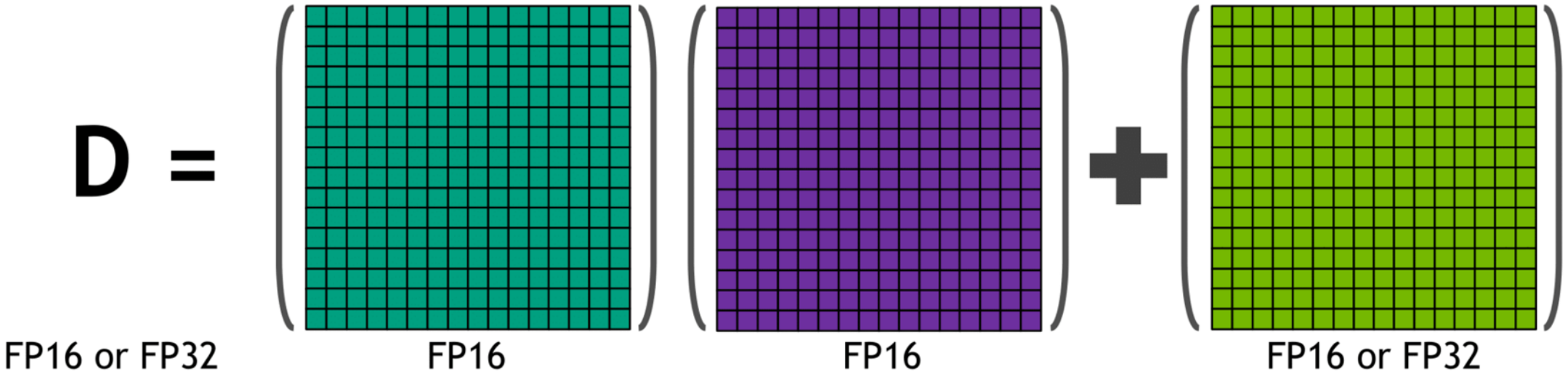
```
1
2   template<typename Use, int m, int n, int k, typename T, typename Layout=void> class fragment;
3
4   void load_matrix_sync(fragment<...> &a, const T* mptr, unsigned ldm);
5   void load_matrix_sync(fragment<...> &a, const T* mptr, unsigned ldm, layout_t layout);
6   void store_matrix_sync(T* mptr, const fragment<...> &a, unsigned ldm, layout_t layout);
7   void fill_fragment(fragment<...> &a, const T& v);
8   void mma_sync(fragment<...> &d, const fragment<...> &a, const fragment<...> &b, const fragment<...> &c, bool satf=false);
9
```

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

CUDA Tensor Core编程

16x16x16 Warp Matrix Multiply and Accumulate(WMMA)

```
wmma::mma_sync(Dmat, Amat, Bmat, Cmat);
```

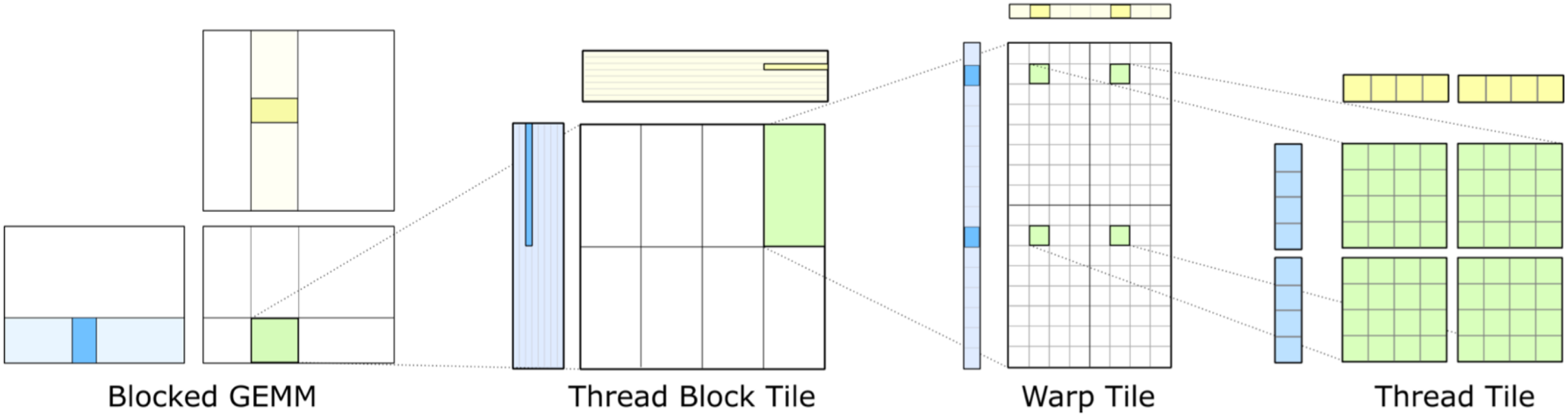


Question?

- 那么 Tensor Core 跟卷积计算或者 GEMM 计算之间怎么映射吗？
  - Tensor Core 一次4x4这么小的kernel，怎么处理 input image 224*224，kernel 7 * 7 的GEMM？
  - 怎么处理 Transformer 结构 input embedding 2048*2048，hidden size 1024*1024 的GEMM？

CNN vs GEMM

- 卷积的计算可以转换为两个矩阵相乘的求解，得到最终的卷积计算结果。
- GEMM 计算可以被成批地放在一起，作为单个大型矩阵乘法运算运行。
- 在 GPU CUDA Core 中，通过线程 Block 提供具体的计算。



Tensor Core 历代发展

	Supported CUDA Core Precisions									Supported Tensor Core Precisions								
	FP8	FP16	FP32	FP64	INT1	INT4	INT8	TF32	BF16	FP8	FP16	FP32	FP64	INT1	INT4	INT8	TF32	BF16
NVIDIA Tesla P4	No	No	Yes	Yes	No	No	Yes	No	No	No	No	No	No	No	No	No	No	No
NVIDIA P100	No	Yes	Yes	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No
NVIDIA Volta	No	Yes	Yes	Yes	No	No	Yes	No	No	No	Yes	No	No	No	No	No	No	No
NVIDIA Turing	No	Yes	Yes	Yes	No	No	Yes	No	No	No	Yes	No	No	Yes	Yes	Yes	No	No
NVIDIA A100	No	Yes	Yes	Yes	No	No	Yes	No	Yes	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
NVIDIA H100	No	Yes	Yes	Yes	No	No	Yes	No	Yes	Yes	Yes	No	Yes	No	No	Yes	Yes	Yes

Reference 引用&参考

1. https://zhuanlan.zhihu.com/p/620257581
2. https://developer.nvidia.com/blog/programming-tensor-cores-cuda-9/
3. https://developer.nvidia.com/blog/accelerating-winml-and-nvidia-tensor-cores/
4. https://developer.nvidia.com/blog/optimizing-gpu-performance-tensor-cores/
5. https://www.anandtech.com/Gallery/Album/6494#15
6. https://www.anandtech.com/Gallery/Album/6493#33
7. https://www.anandtech.com/show/12673/titan-v-deep-learning-deep-dive/3
8. https://www.anandtech.com/show/12673/titan-v-deep-learning-deep-dive/4
9. https://www.cnblogs.com/wujianming-110117/p/12993096.html
10. https://www.cnblogs.com/wujianming-110117/p/12993096.html
11. https://aijishu.com/a/1060000000286803#item-2
12. https://learnopencv.com/demystifying-gpu-architectures-for-deep-learning-part-2/