



CS 380 - GPU and GPGPU Programming Lecture 2: Introduction, Pt. 2

Markus Hadwiger, KAUST

Reading Assignment #1 (until Sep 6)



Read (required):

- Orange book, chapter 1 (Review of OpenGL Basics)
- Orange book, chapter 2 (*Basics*)

What are GPUs?

Graphics Processing Units

But evolved toward

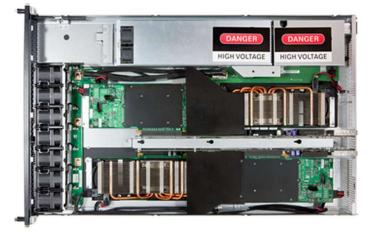
- Very flexible, massively parallel floating point co-processors
- But not entirely programmable!
- Fixed-function parts have definite advantages (e.g., texture filtering, z-buffering)

We will cover both perspectives

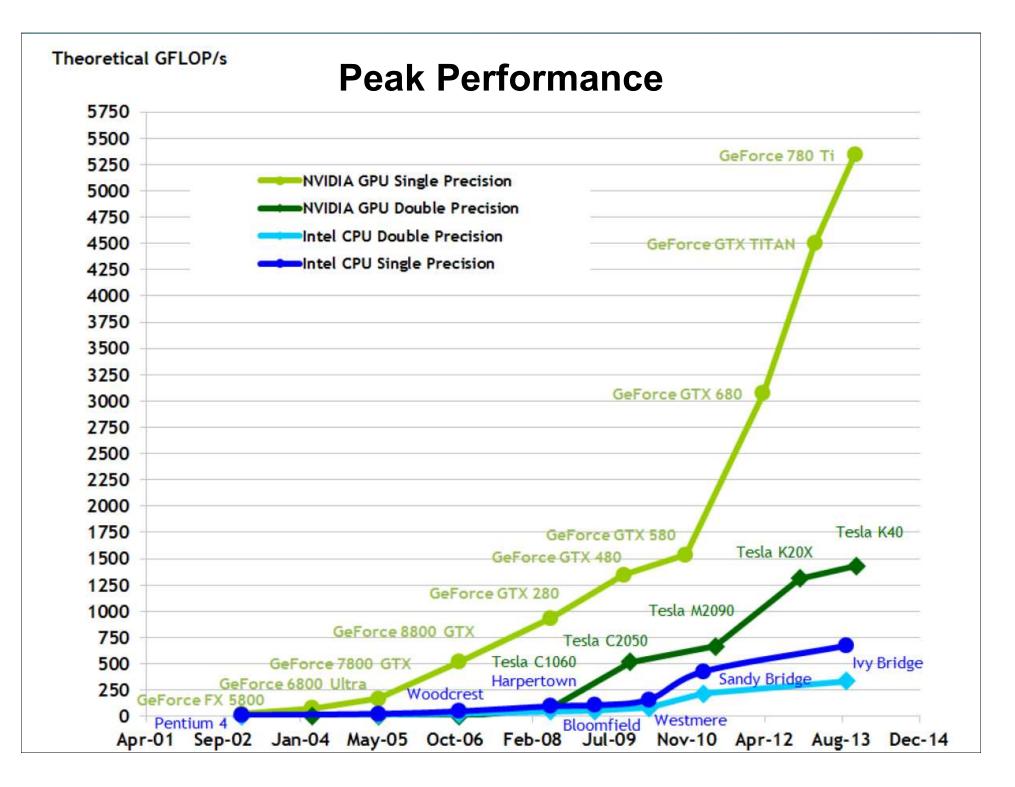
- GPUs for graphics
- GPU computing (GPGPU general purpose computation on GPU)









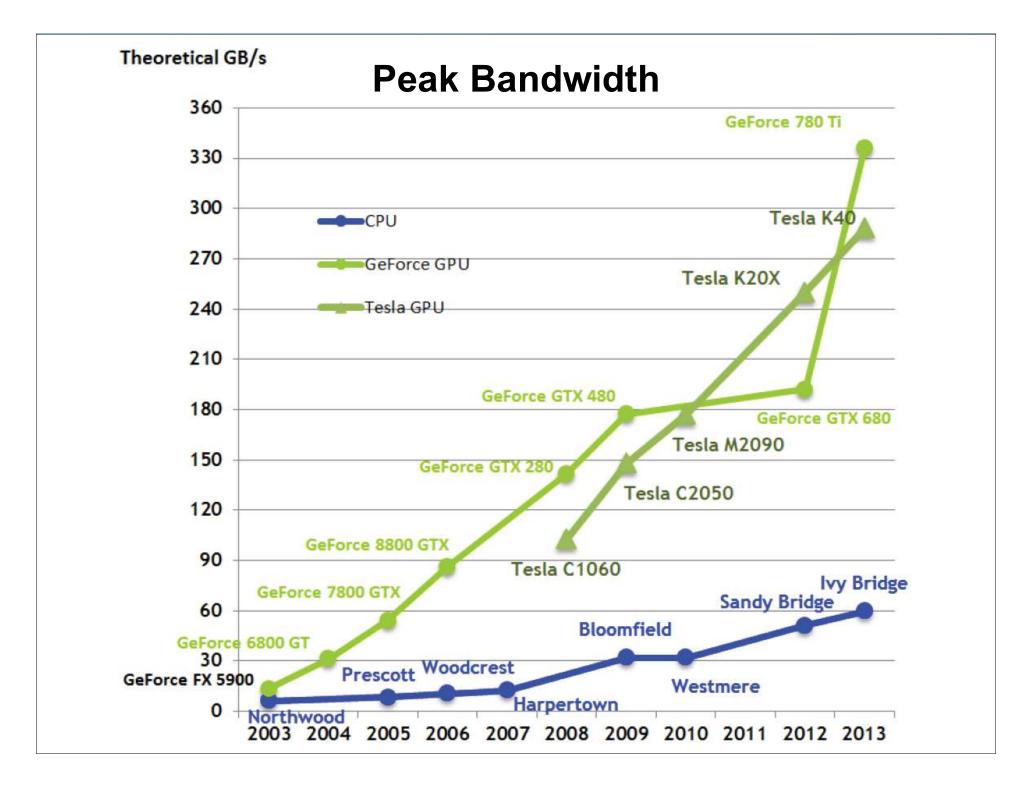


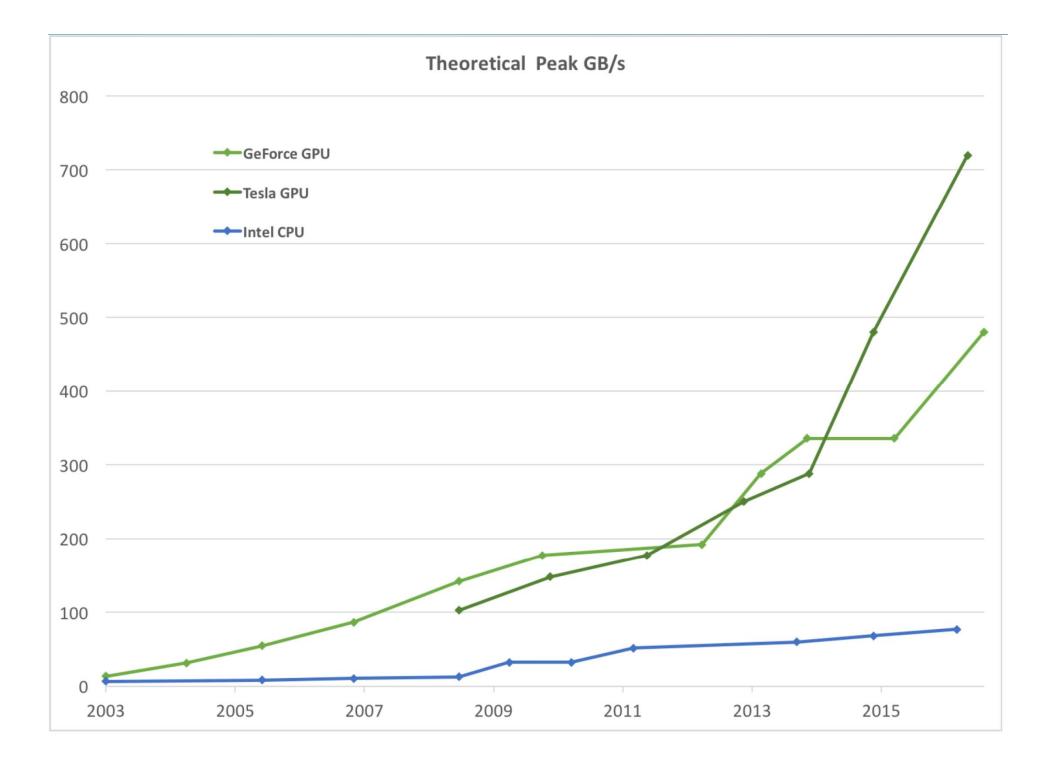
Peak Performance



			Theoretical GFLOP/s	at base clock		
11000						
10500	NVIDIA GPU Single	e Precision				
10000		le Precision				
9500	Intel CPU Single P	recision				
9000	Intel CPU Double					
8500						
8000						
7500						
7000						
6500						
6000						
5500					J.	
5000						
4500						
4000						
3500						
3000						
2500						
2000						
1500						
1000						
500			+			
2003	2005	2007	2009	2011	2013	2015

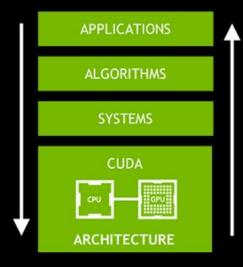
Markus Hadwiger, KAUST

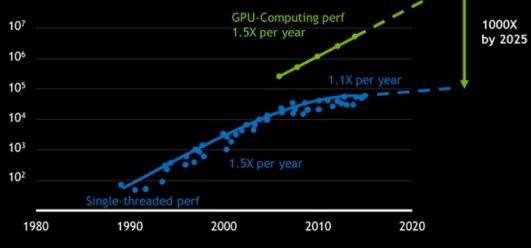






RISE OF GPU COMPUTING



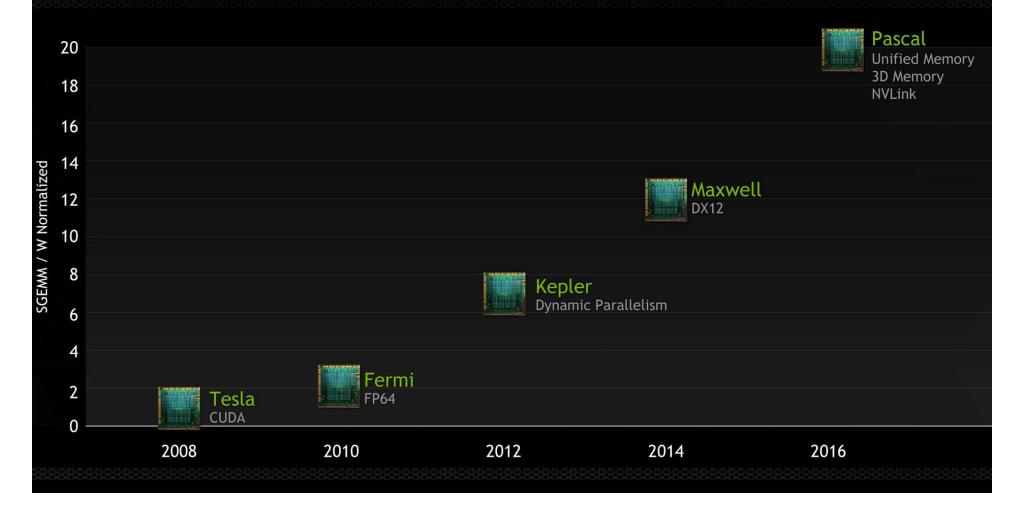


Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten New plot and data collected for 2010-2015 by K. Rupp

GPU Architectures Over the Years

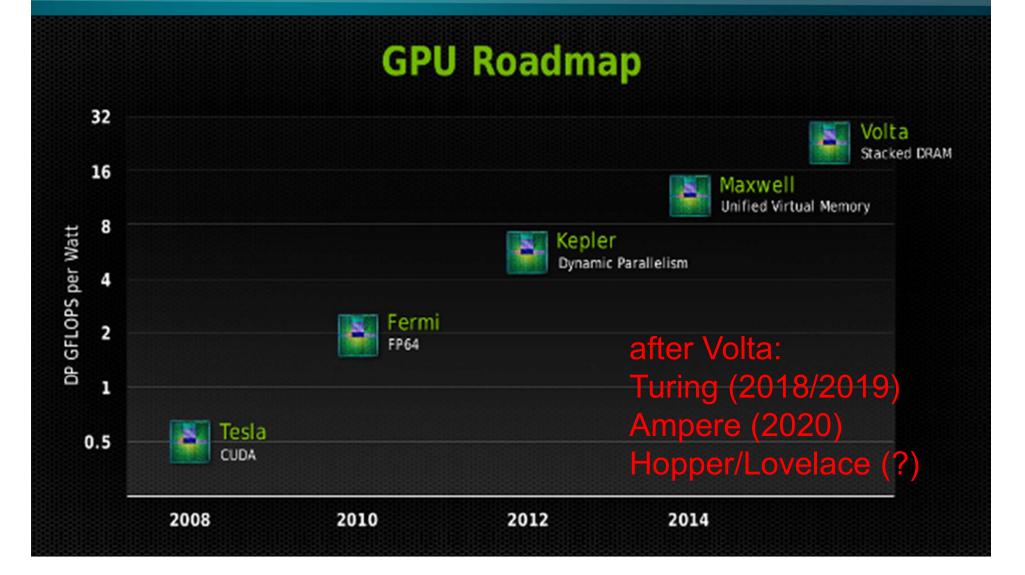


GPU Roadmap



GPU Architectures Over the Years





Recent Updates



NVIDIA Ampere architecture (2020)

https://en.wikipedia.org/wiki/Ampere_(microarchitecture)

Promo presentation from Sep 1, 2020:

https://www.nvidia.com/en-us/geforce/special-event/

Geforce 30-series (Ampere):

https://nvidia.com/en-us/geforce/graphics-cards/30-series/

RTX 3090 has 10,496 CUDA cores

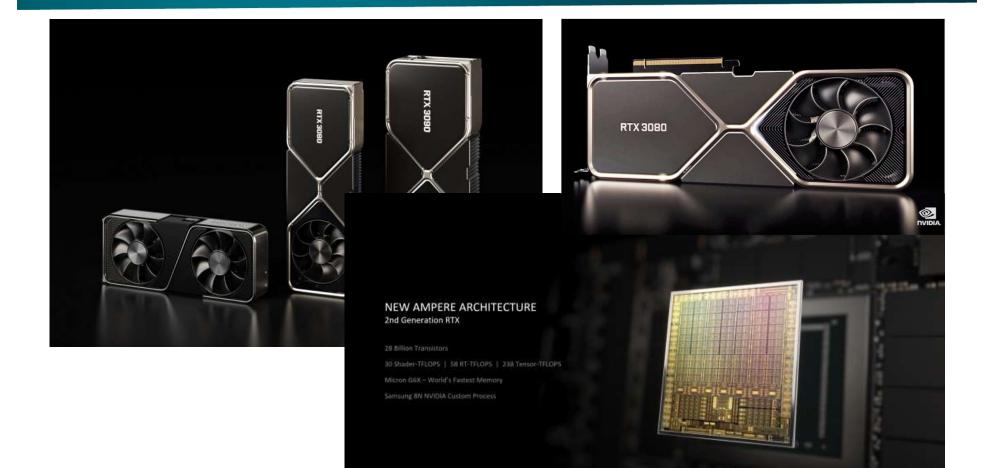
A100 (Ampere):

https://www.nvidia.com/en-us/data-center/a100/

A100 has 6,912 CUDA cores

Recent Updates





Markus Hadwiger, KAUST

Overviews and Specs



Wikipedia has many comprehensive lists of architectures and specs

https://en.wikipedia.org/wiki/ List_of_Nvidia_graphics_processing_units

https://en.wikipedia.org/wiki/ List_of_AMD_graphics_processing_units

What is in a GPU?



Lots of floating point processing power

• Stream processing cores different names: stream processors, CUDA cores, ...





• Was vector processing, now scalar cores!

Still lots of fixed graphics functionality

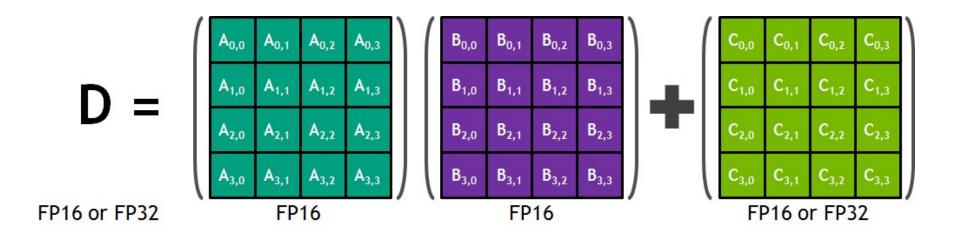
- Attribute interpolation (per-vertex -> per-fragment)
- Rasterization (turning triangles into fragments/pixels)
- Texture sampling and filtering
- Depth buffering (per-pixel visibility)
- Blending/compositing (semi-transparent geometry, ...)
- Frame buffers



Example for "Special Cores": Tensor Cores



Mixed-precision, fast matrix-matrix multiply and accumulate



From this, build larger sizes, higher dimensionalities, ...

NVIDIA Volta SM

Multiprocessor: SM

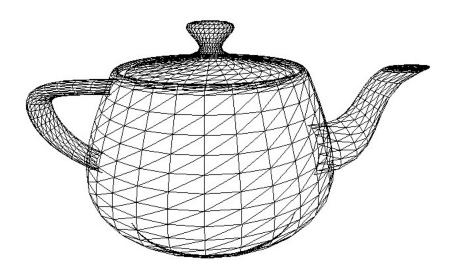
- 64 FP32 + INT32 cores
- 32 FP64 cores
- 8 tensor cores (FP16/FP32 mixed-precision)

4 partitions inside SM

- 16 FP32 + INT32 cores each
- 8 FP64 cores each
- 8 LD/ST units each
- 2 tensor cores each
- Each has: warp scheduler, dispatch unit, register file

М														
						L1 Instruct	tion Cache							
L0 Instruction Cache							L0 Instruction Cache							
Warp Scheduler (32 thread/clk)						Warp Scheduler (32 thread/clk)								
Dispatch Unit (32 thread/clk)							Dis	spatcl	h Unit	(32 th	read/cll	k)		
Register File (16,384 x 32-bit)						Register File (16,384 x 32-bit)								
FP64	INT	INT	FP32	FP32			FP64	INT	INT	FP32	FP32			
FP64	INT	INT	FP32	FP32			FP64	INT	INT	FP32	FP32			
FP64	INT	INT	FP32	FP32			FP64	INT	INT	FP32	FP32			
FP64	INT	INT	FP32	FP32	TENSOR		FP64	INT	INT	FP32	FP32	TENS		TENSOR
FP64	INT	INT	FP32	FP32	CORE	CORE	FP64	INT	INT	FP32	FP32	COF	₹E	CORE
FP64	INT	INT	FP32	FP32			FP64	INT	INT	FP32	FP32			
FP64	INT	INT	FP32	FP32			FP64	INT	INT	FP32	FP32			
								INT	INT	FP32	ED22			
FP64	INT	INT	FP32	FP32			FP64		6.6		FF32			
FP64 LD/ LD/ ST ST	INT LD/ ST	INT LD/ ST	FP32 LD/ ST	FP32 LD/ ST	LD/ ST ST	SFU	FP64 LD/ LD/ ST ST	LD/ ST	LD/ ST	LD/ ST	LD/ ST	LD/ ST	LD/ ST	SFU
LD/ LD/	LD/	LD/ ST	LD/	LD/ ST	ST ST	SFU	LD/ LD/	LD/	ST	LD/	LD/ ST	ST		SFU
LD/ LD/	LD/ ST	LD/ ST	LD/ ST	LD/ ST	ST ST	SFU	LD/ LD/	LD/ ST	ST L0 li	LD/ ST	LD/ ST	ST	ST	SFU
LD/ LD/	LD/ ST War	LD/ ST LO II	LD/ ST	LD/ ST tion C	ST ST	SFU	LD/ LD/	LD/ ST	ST LO II p Sch	LD/ ST	LD/ ST tion C	ST ache	ST	SFU
LD/ LD/	LD/ ST War Dis	LD/ ST LO II rp Sch spatcl	LD/ ST nstruc nedule h Unit	LD/ ST tion C r (32 th (32 th	ST ST ache hread/clk)	SFU	LD/ LD/	LD/ ST War Dis	ST L0 In p Sch spatcl	LD/ ST nstruct nedule h Unit	LD/ ST tion C r (32 t (32 th	ST ache hread/c	ST :lk) k)	SFU
LD/ LD/	LD/ ST War Dis	LD/ ST LO II rp Sch spatcl	LD/ ST netruct h Unit File (1	LD/ ST tion C r (32 th (32 th	ST ST ache hread/clk) read/clk)	SFU	LD/ LD/	LD/ ST War Dis	ST L0 In p Sch spatcl	LD/ ST nstruct nedule h Unit	LD/ ST tion C r (32 t (32 th	ST ache hread/c read/cll	ST :lk) k)	SFU
LD/ LD/ ST ST	LD/ ST War Di: Reg	LD/ ST L0 In p Sch spatch	LD/ ST netruct h Unit File (1	LD/ ST tion C (32 th (32 th (32 th (32 th (32 th (32 th) (32 t	ST ST ache hread/clk) read/clk)	SFU		LD/ ST War Dis Reg	ST LO II p Sch spatcl ister	LD/ ST nstruct nedule h Unit File (1	LD/ ST tion C r (32 t (32 th (32 th (32 th (32 th (32 th) (32	ST ache hread/c read/cll	ST :lk) k)	SFU
LD/ LD/ ST ST	LD/ ST War Dis Reg	LD/ ST L0 In p Sch spatc ister INT	LD/ ST netruc hedule h Unit File (1 FP32	LD/ ST tion C (32 th (32 th (32 th (32 th (32 th (32 th) (32 t	ST ST ache hread/clk) read/clk)	SFU	LD/ LD/ ST ST	LD/ ST War Dis Reg	ST LO II p Sch spatcl ister INT	LD/ ST nstruct nedule h Unit File (1 FP32	LD/ ST tion C r (32 th (32 th 16,384 FP32 FP32	ST ache hread/c read/cll	ST :lk) k)	SFU
LD/ ST ST FP64 FP64	LD/ ST War Dis Reg INT	LD/ ST L0 li p Sch spatcl ister INT INT	LD/ ST nstruct nedule h Unit File (1 FP32 FP32 FP32	LD/ ST tion C (32 th (32 th (32 th (32 th (32 th (32 th (32 th) (32 th	st st ache hread/clk) read/clk) 4 x 32-bit) 4 x 32-bit)	TENSOR	LD/ LD/ ST ST	LD/ ST War Dis Reg INT INT	ST LO II p Sch spatcl ister INT INT	LD/ ST netruch H Unit File (1 FP32	LD/ ST (32 th (32 th (6,384 FP32 FP32 FP32	st ache hread/c 4 x 32-l TENS	ST	TENSOR
LD/ ST ST FP64 FP64 FP64	LD/ ST War Dis Reg	LD/ ST LO II TP Sch spatc ister INT INT INT	LD/ ST nstruct nedule h Unit File (1 FP32 FP32 FP32 FP32	LD/ ST (32 th (32 th (6,384 FP32 FP32 FP32	ST ST ache hread/clk) read/clk) 4 x 32-bit)		LD/ LD/ ST ST	LD/ ST War Dis Reg INT INT INT	ST LO In p Sch spatcl ister INT INT INT	LD/ ST nstruct h Unit File (1 FP32 FP32 FP32	LD/ ST Lion C (32 th (32 th (32 th (32 th FP32 FP32 FP32 FP32	st ache hread/cll 4 x 32-l	ST	
LD/ ST ST FP64 FP64 FP64 FP64	LD/ ST War Dis Reg INT INT INT	LD/ ST LO In rp Sch spatcl ister INT INT INT INT	LD/ ST netruc netule h Unit FP32 FP32 FP32 FP32 FP32	LD/ ST (32 th (32 th (32 th (32 th FP32 FP32 FP32 FP32	st st ache hread/clk) read/clk) 4 x 32-bit) 4 x 32-bit)	TENSOR	FP64 FP64 FP64 FP64	LD/ ST War Dis Reg INT INT INT	ST LO II p Sch spatcl ister INT INT INT	LD/ ST netruct nedule h Unit File (1 FP32 FP32 FP32 FP32	LD/ ST (32 th (32 th 16,384 FP32 FP32 FP32 FP32 FP32 FP32	st ache hread/c 4 x 32-l TENS	ST	TENSOR
LD/ ST ST ST ST ST ST ST ST ST ST ST ST ST ST S	LD/ ST War Dia NT INT INT INT INT	LD/ ST LO In p Sch spatcl ister INT INT INT INT	LD/ ST nestrucci nedule h Unit FP32 FP32 FP32 FP32 FP32 FP32	LD/ ST (32 th (32 th 6,384 FP32 FP32 FP32 FP32 FP32 FP32	st st ache hread/clk) read/clk) 4 x 32-bit) 4 x 32-bit)	TENSOR	LD/ LD/ ST ST FP64 FP64 FP64 FP64 FP64 FP64	LD/ ST War Dis Reg INT INT INT INT	ST LO II p Sch spatcl ister INT INT INT INT	LD/ ST netruc/ netuler h Unit FP32 FP32 FP32 FP32 FP32	LD/ ST tion C (32 th (32 th (32 th (32 th (32 th (32 th (32 th (32 th (32 th) (32 th)	st ache hread/c 4 x 32-l TENS	ST	TENSOR
LD/ ST LD/ ST ST ST ST ST ST ST ST ST ST ST ST ST	LD/ ST War Dis Reg INT INT INT INT INT	LD/ ST LO li pp Sch spatcl ister INT INT INT INT	LD/ ST netruc/ netule h Unit FP32 FP32 FP32 FP32 FP32 FP32 FP32	LD/ ST idon C (32 th (32 th (32 th (32 th (32 th (32 th (32 th (32 th (32 th) (32 th)	st st ache hread/clk) read/clk) 4 x 32-bit) 4 x 32-bit)	TENSOR	LD/ LD/ ST ST FP64 FP64 FP64 FP64 FP64 FP64 FP64	UD/ ST War Dis Reg INT INT INT INT INT	ST LO II p Sch spatcl ister INT INT INT INT INT	LD/ ST netruc/ netruc/ File (1 FP32 FP32 FP32 FP32 FP32 FP32	LD/ ST ition C (32 th (32 th))))))))))))))))))))))))))))))))))))	st ache hread/c 4 x 32-l TENS	ST	TENSOR
LD/ ST LD/ ST FP64 FP64 FP64 FP64 FP64 FP64 FP64	LD/ ST War Dis Reg INT INT INT INT INT INT	LD/ ST LO II p Sch spatcl ister INT INT INT INT INT INT	LD/ ST netruc/ netule h Unit FP32 FP32 FP32 FP32 FP32 FP32 FP32	LD/ ST Ldon C (32 th (32 th))))))))))))))))))))))))))))))))))))	st st ache hread/clk) read/clk) 4 x 32-bit) 4 x 32-bit)	TENSOR	FP64 FP64 FP64 FP64 FP64 FP64 FP64 FP64	LD/ ST War Dis Reg INT INT INT INT INT INT	ST LO III P Sch Spatc ister INT INT INT INT INT INT	LD/ ST nstruc neduler h Unit FP32 FP32 FP32 FP32 FP32 FP32 FP32	LD/ ST ition C (32 th (32 th))))))))))))))))))))))))))))))))))))	st ache hread/c 4 x 32-l TENS COF	ST	TENSOR
LD/ ST ST ST ST ST ST ST ST ST ST ST ST ST ST S	UNT INT INT INT INT INT INT INT INT INT	LD/ ST LO II p Sch spatcl ister INT INT INT INT INT INT INT INT INT	LD/ ST nstrucc nedule h Unit FP32 FP32 FP32 FP32 FP32 FP32 FP32 FP32	LD/ ST Lion C (32 th (32 th))))))))))))))))))))))))))))))))))))	st st ache hread/clk) read/clk) 4 x 32-bit) 4 x 32-bit) TENSOR CORE	TENSOR	FP64 FP64 FP64 FP64 FP64 FP64 FP64 FP64	LD/ ST Dis Reg INT INT INT INT INT INT INT LD/ ST	ST LO III P Sch ister INT INT INT INT INT INT INT LD/ ST	LD/ ST nstruc redule h Unit File (1 FP32 FP32 FP32 FP32 FP32 FP32 FP32 FP32	LD/ ST (32 th (32 th) (32 th) (3	ST ache hread/c 4 x 32-l TENS COF	st ilk) k) bit) GOR RE LD/	

Real-time graphics primitives (entities)



Represent surface as a 3D triangle mesh

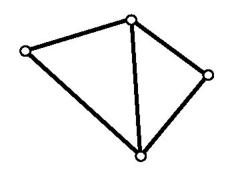
01

o 4

o 2

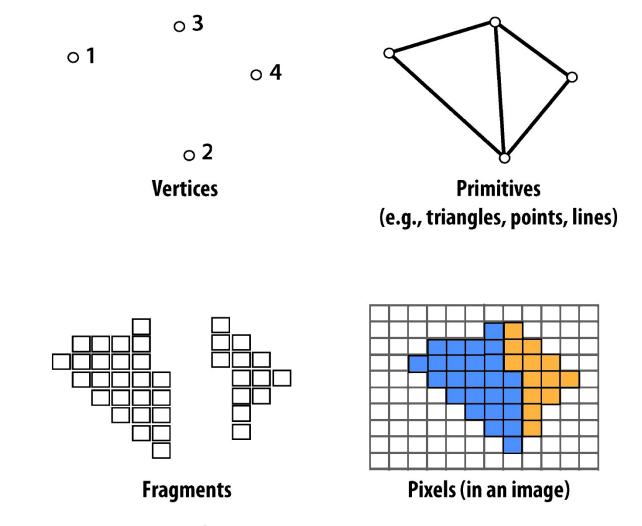
Vertices

o 3



Primitives (e.g., triangles, points, lines)

Real-time graphics primitives (entities)



Courtesy Kayvon Fatahalian, CMU

CMU 15-418, Spring 2015

What can the hardware do?

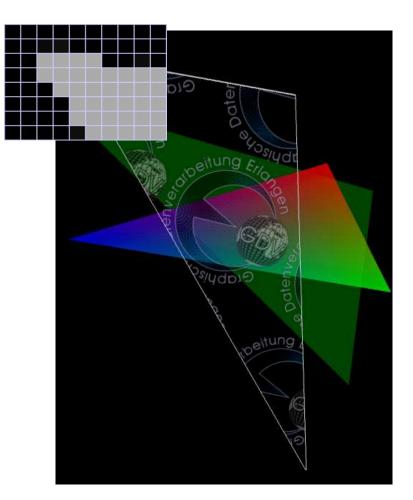


Rasterization

- Decomposition into fragments
- Interpolation of color
- Texturing
 - Interpolation/Filtering
 - Fragment Shading

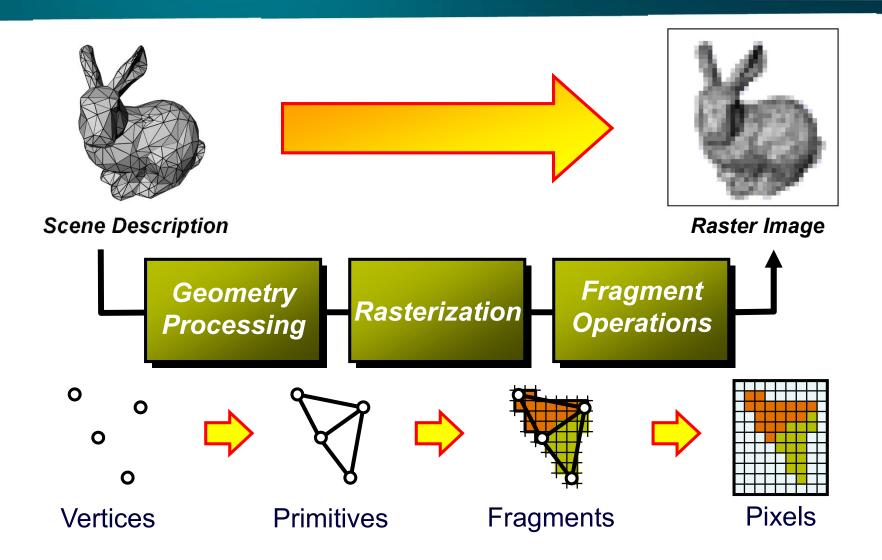
Fragment Operations

- Depth Test (Z-Test)
- Alpha Blending (Compositing)



Graphics Pipeline





Thank you.