

UNIVERSIDAD DEL BÍO-BÍO FACULTAD DE CIENCIAS EMPRESARIALES

# GPUs **Heterogeneous Computing**

Professor: Dr. Joel Fuentes - jfuentes@ubiobio.cl

Teaching Assistants:

- Daniel López <u>daniel.lopez1701@alumnos.ubiobio.cl</u> Sebastián González <u>sebastian.gonzalez1801@alumnos.ubiobio.cl</u>

Course website: <u>http://www.face.ubiobio.cl/~jfuentes/classes/hc</u>

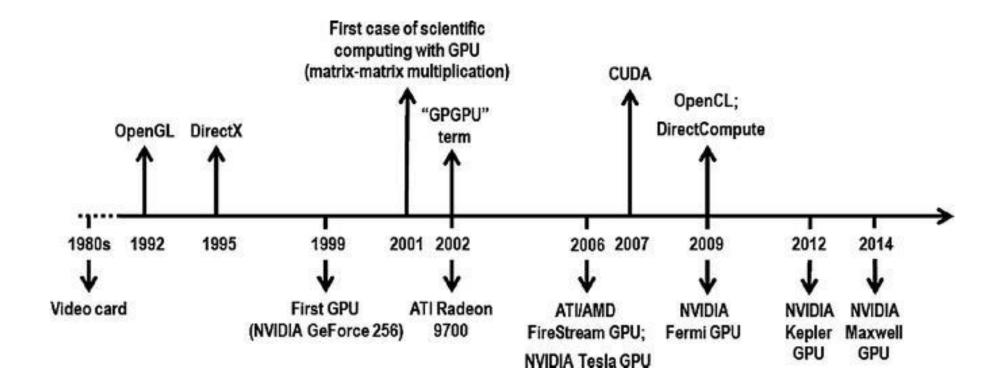
### Contents

- History of GPUs
- Sample Application: 3D Rendering
- Nvidia, AMD, Intel architectures
- GPU Programming

### History of GPUs

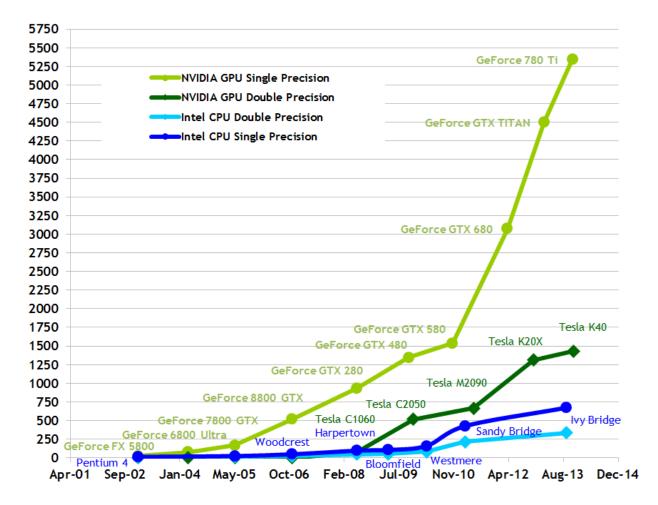
- GPUs (Graphics Processing Units) were initially created for 3D video and graphic acceleration.
- Initially designed with fixed functions without the possibility of flexible programming.
- Since 2001 it began to include the possibility of programming GPUs
- Today its uses have evolved and include:
- Computer vision
- Deep learning
- Scientific computing
- Cryptomining

#### History of GPUs



#### Performance en GFLOP/s vs CPU

#### Theoretical GFLOP/s



#### Sample Application: 3D Rendering

• Task consists of calculating how each triangle in the 3D mesh contributes to the appearance of each pixel of the rendered image.

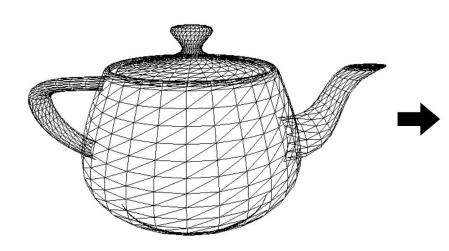




Image credit: Henrik Wann Jensen

Description of the scene based on: mesh of triangles, lights, camera, etc. Image of the scene

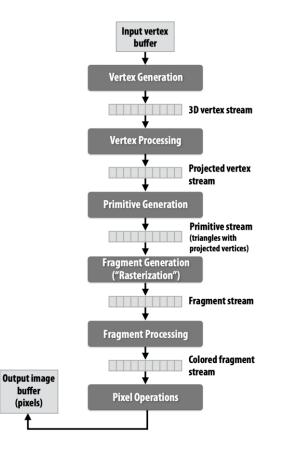
## Real-time 3D rendering

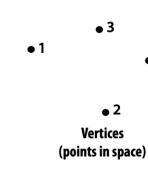


Flight Simulator (2021) 7

#### Sample Application: 3D Rendering

• Steps in rendering given a mesh of 3D triangles

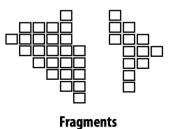


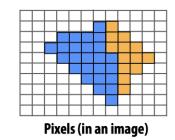


• 4



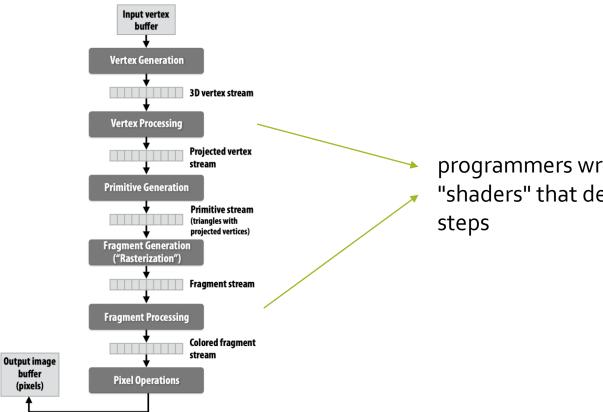
Primitives (e.g., triangles, points, lines)





#### Sample Application: 3D Rendering

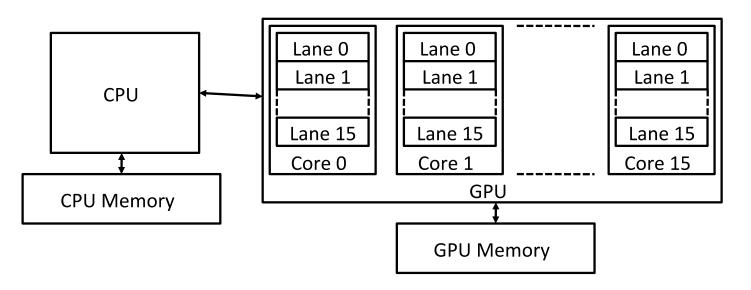
• Steps in rendering given a mesh of 3D triangles



programmers write mini-programs called "shaders" that describe the logic of these steps

#### **GPU** Architectures

#### Execution model



- GPU is built based on multiple simple parallel cores. Each core can execute SIMD (Single Instruction Multiple Data) instructions.
- The CPU sends processing tasks (meshes, buffers, etc.) to the GPU, which distributes the work to its cores.

#### **GPU** Architectures

- The cores on the GPU work under the thread model in shared memory.
- GPUs can run hundreds or thousands of threads in parallel.
- GPUs usually have their own DRAM
- GPUs are good for:
  - Data-parallel processing: The same operation executed on many data elements in parallel.
  - Processing with high arithmetic intensity.

#### **GPU** Architectures

- Comparison with CPU:
- GPUs occupy more transistors in data processing
- GPUs have smaller caches
- GPU ALU is simpler than CPU ALU

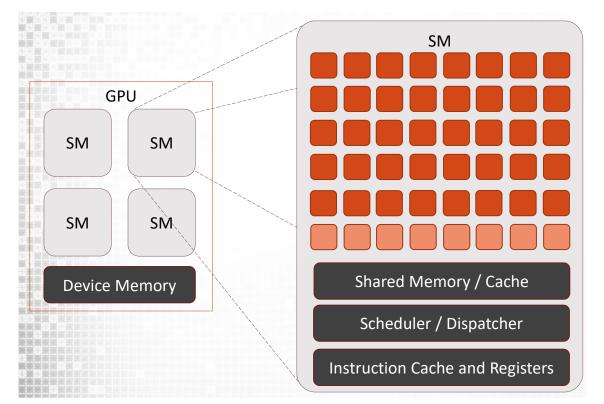
| Control | ALU | ALU |  |  |  |  |  |
|---------|-----|-----|--|--|--|--|--|
|         | ALU | ALU |  |  |  |  |  |
| Cache   |     |     |  |  |  |  |  |
|         |     |     |  |  |  |  |  |
| DRAM    |     |     |  |  |  |  |  |
| CPU     |     |     |  |  |  |  |  |





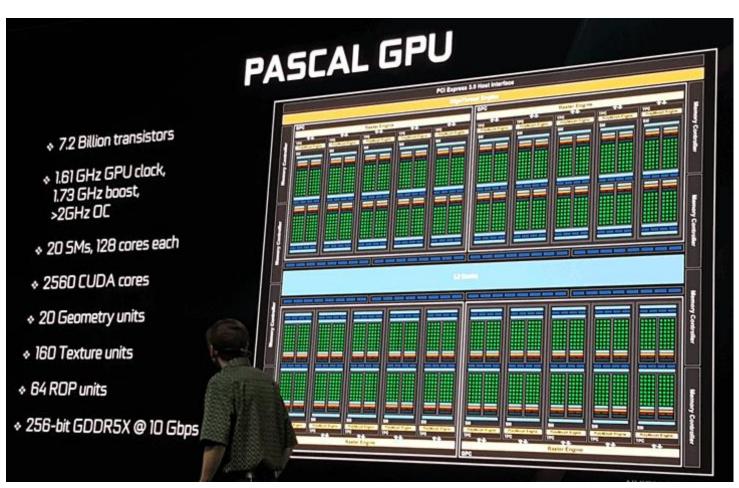
#### **GPU** Nvidia

- Have a 2-level hierarchy
- Each Streaming Multiprocessor (SM) has multiple CUDA cores
- The number of SMs varies depending on the type of GPU



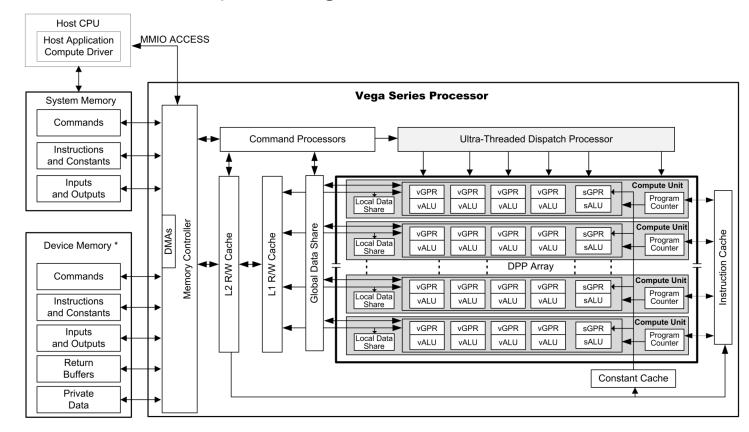
#### **GPU** Nvidia

• Image of Pascal architecture (1080 Ti)



#### **GPU AMD**

- GPR: General Purpose Records
- vALU: ALU vector for SIMD processing



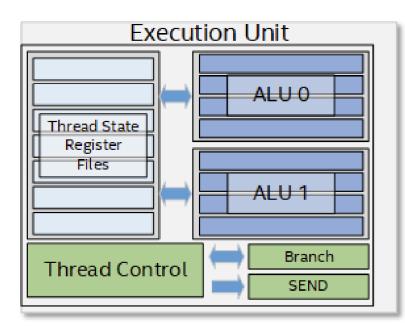
#### **GPU AMD**

- Vega 20 GPU image
- NCU: Next compute unit

|                                      | ACE                   | ACE          | H                                    | WS           |                                      | Graphics Com | mand                                 | Processor    | ни           | /5          | ACE                     | ACE |
|--------------------------------------|-----------------------|--------------|--------------------------------------|--------------|--------------------------------------|--------------|--------------------------------------|--------------|--------------|-------------|-------------------------|-----|
|                                      | Workgroup Distributor |              |                                      |              |                                      |              |                                      |              |              |             |                         |     |
| Graphics Pipeline<br>Geometry Engine |                       |              | Graphics Pipeline<br>Geometry Engine |              | Graphics Pipeline<br>Geometry Engine |              | Graphics Pipeline<br>Geometry Engine |              |              |             |                         |     |
| DSBR                                 |                       |              | DSBR                                 |              |                                      | DSBR         |                                      | DSBR         |              |             |                         |     |
| Г                                    | NCU                   | NCU          |                                      | Г            | NCU                                  | NCU          |                                      | NCU          | NCU          |             | NCU                     | NCU |
|                                      | NCU                   | NCU          |                                      |              | NCU                                  | NCU          |                                      | NCU          | NCU          |             | NCU                     | NCU |
| Engine                               | NCU                   | NCU          |                                      | Engine       | NCU                                  | NCU          | e Engine                             | NCU          | NCU          | Engine      | NCU                     | NCU |
| Eng                                  | NCU                   | NCU          |                                      | E Eng        | NCU                                  | NCU          |                                      | NCU          | NCU          | e Eng       | NCU                     | NCU |
| Compute                              | NCU                   | NCU          |                                      | Compute      | NCU                                  | NCU          | Compute                              | NCU          | NCU          | Compute     | NCU                     | NCU |
| Com                                  | NCU                   | NCU          |                                      | Corr         | NCU                                  | NCU          | Corr                                 | NCU          | NCU          | Corr        | NCU                     | NCU |
|                                      | NCU                   | NCU          |                                      |              | NCU                                  | NCU          |                                      | NCU          | NCU          |             | NCU                     | NCU |
|                                      | NCU                   | NCU          |                                      | L            | NCU                                  | NCU          |                                      | NCU          | NCU          |             | NCU                     | NCU |
| Pixel Engine Pixel Engine            |                       |              | Pix                                  | el Engine    | Pixel Engine                         | Pi           | xel Engine                           | Pixel Engine | P            | ixel Engine | Pixel Engine            |     |
| Pixel Engine                         |                       | Pixel Engine |                                      | Pixel Engine |                                      | Pixel Engine |                                      | xel Engine   | Pixel Engine | P           | Pixel Engine Pixel Engi |     |
|                                      |                       |              |                                      |              |                                      | L2 (         | Cache                                |              |              |             |                         |     |

#### **GPU** Intel

- Compute unit or core is called Execution Unit (EU)
- Multiple EUs grouped into Subslices
- Multiple Subslices grouped into Slices
- Memory hierarchy up to L3



| GTI  |                       | Global Assets         |                       | Media Fixed Function  |                      | Blitter               |                    |  |
|--|-----------------------|-----------------------|-----------------------|-----------------------|----------------------|-----------------------|--------------------|--|
|  | Geometry              |                       |                       |                       |                      |                       |                    |  |
| SubSlice   |                       | SubSlice              |                       | SubSlice              |                      | SubSlice              |                    |  |
|  | I\$ & thread dispatch |                       | I\$ & thread dispatch |                       | d dispatch           |                       | d dispatch         |  |
| EU   | EU                    | EU                    | EU                    | EU                    | EU                   | EU                    | EU                 |  |
| EU   | EU                    | EU                    | EU                    | EU                    | EU                   | EU                    | EU                 |  |
| EU   | EU                    | EU                    | EU                    | EU                    | EU                   | EU                    | EU                 |  |
| EU   | EU                    | EU                    | EŬ                    | EU                    | EU                   | EU                    | EU                 |  |
| Sampler  | SLM                   | Sampler               | SLM                   | Sampler               | SLM                  | Sampler               | SLM                |  |
| Tex\$  |                       | Tex\$                 |                       | Tex\$                 |                      | Tex\$                 |                    |  |
| Media<br>Sampler                                       | Dataport<br>(LD/ST)   | Media<br>Sampler      | Dataport<br>[LD/ST]   | Media<br>Sampler      | Dataport<br>(LD/ST)  | Media<br>Sampler      | Datapor<br>[LD/ST] |  |
| SubSlice   |                       | SubSlice              |                       | SubSlice              |                      | SubSlice              |                    |  |
| I\$ & thread   | d dispatch            | I\$ & thread dispatch |                       | I\$ & thread dispatch |                      | I\$ & thread dispatch |                    |  |
| EU   | EU                    | EU                    | EU                    | EU                    | EU                   | EU                    | EU                 |  |
| EU   | EU                    | EU                    | EU                    | EU                    | EU                   | EU                    | EU                 |  |
| EU   | EU                    | EU                    | EU                    | EU                    | EU                   | EU                    | EU                 |  |
| EU   | EU                    | EU                    | EU                    | EU                    | EU                   | EU                    | EU                 |  |
| Sampler  | SLM                   | Sampler               | SLM                   | Sampler               | SLM                  | Sampler               | SLM                |  |
| Tex\$  |                       | Tex\$                 |                       | Tex\$                 |                      | Tex\$                 |                    |  |
| Media<br>Sampler                                       | Dataport<br>(LD/ST]   | Media<br>Sampler      | Dataport<br>[L0/ST]   | Media<br>Sampler      | Dataport<br>(1.0/ST) | Media<br>Sampler      | Datapor<br>(LD/ST] |  |
| u l  |                       |                       | F                     | Raster                |                      |                       |                    |  |
| Raster<br>HiZ/Depth<br>Pixel Dispatch<br>Pixel Backend |                       |                       |                       |                       |                      |                       |                    |  |
| ୦ Pixel Dispatch                                       |                       |                       |                       |                       |                      |                       |                    |  |
| Pixel Backend  |                       |                       |                       |                       |                      |                       |                    |  |
|  | L3\$                  |                       |                       |                       |                      |                       |                    |  |
|  |                       |                       |                       | 8                     |                      |                       |                    |  |

Figure 4: Gen11 detailed block diagram.

#### **GPU** Intel

- Xe architecture block diagram (integrated GPU)
- Soon Intel will launch discrete GPUs

| SHAR                      | ED FUNCTIONS   | COPY ENGINE  | MEDIA  | ENGINE  |  |  |  |
|---------------------------|--|--|--|---|--|--|--|
|                           | GEON   | METRY  | RASTER   | PIXEL DISPATCH  |  |  |  |
|                           | SUBSLICE   | SUBSLICE   | SUBSLICE SUBSLICE  | SUBSLICE SUBSLICE<br>IS THREAD DISPATCH IS THREAD DISPATCH              |  |  |  |
|                           | EU EU EU EU  | EU EU EU EU  | EU EU EU EU EU EU EU   | EU EU EU EU EU EU EU  |  |  |  |
| СE                        | EU EU EU EU  | EU EU EU EU  | EU EU EU EU EU EU EU   | EU EU EU EU EU EU EU  |  |  |  |
| SLI                       | EU EU EU EU  | EU EU EU EU  | EU EU EU EU EU EU EU   | EU EU EU EU EU EU EU  |  |  |  |
| $X^{\circ}_{\circ}$ slice | EU EU EU EU<br>WEDIY<br>WEDIY<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store<br>Store | EU EU EU EU<br>WEDI<br>WEDI<br>WEDI<br>VOVO<br>LI/TOXS SUM | EU EU< | EU E                                |  |  |  |
|                           | PIXEL B  | ACKEND   | PIXEL BACKEND  | PIXEL BACKEND   |  |  |  |
|                           |  | L3 CACH  |  | GAM   |  |  |  |
| -                         | -  |  |  | GTI GTI<br>R - 64B/CLK R - 64B/<br>W - 64B/CLK W - 64B<br>MEMORY FABRIC |  |  |  |

## GPU Programming

- What do we program GPUs in?
- CUDA (Nvidia)
- OpenCL (open standard)
- OpenACC
- SYCL (open standard)
- DPC++
- In the next unit we will see different programming models.

## References

- Parallel Computing, CS 149 (Fall 2019), Stanford University
- Paul Richmond. GPU Arquitectures <a href="http://paulrichmond.shef.ac.uk/teaching/COM4521/">http://paulrichmond.shef.ac.uk/teaching/COM4521/</a>
- Qin CZ. (2017) Cuda/GPU. In: Shekhar S., Xiong H., Zhou X. (eds) Encyclopedia of GIS. Springer, Cham. https://doi.org/10.1007/978-3-319-17885-1\_1606