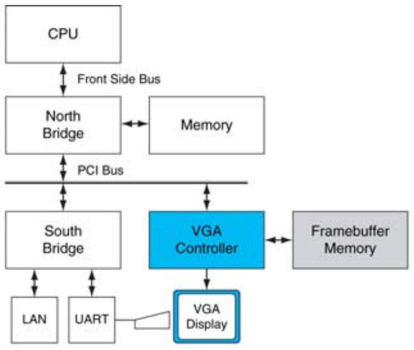
Graphic Processing Units (Section 6.6 and Appendix C)

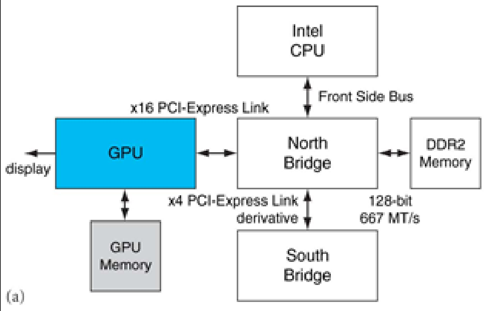
History of GPUs

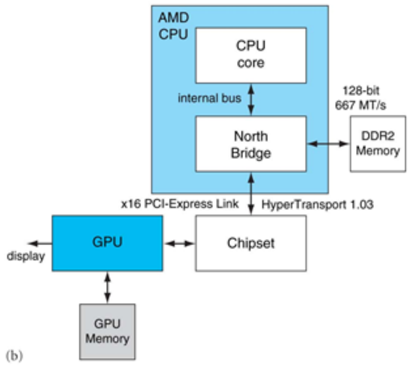
- VGA (Video graphic array) in early 90's -- A memory controller and display generator connected to some (video) RAM
- By 1997, VGA controllers were incorporating some acceleration functions
- mapping, rasterization



- In 2000, a single chip graphics processor incorporated almost every detail of the traditional high-end workstation graphics pipeline
  - Processors oriented to 3D graphics tasks
  - Vertex/pixel processing, shading, texture mapping, rasterization

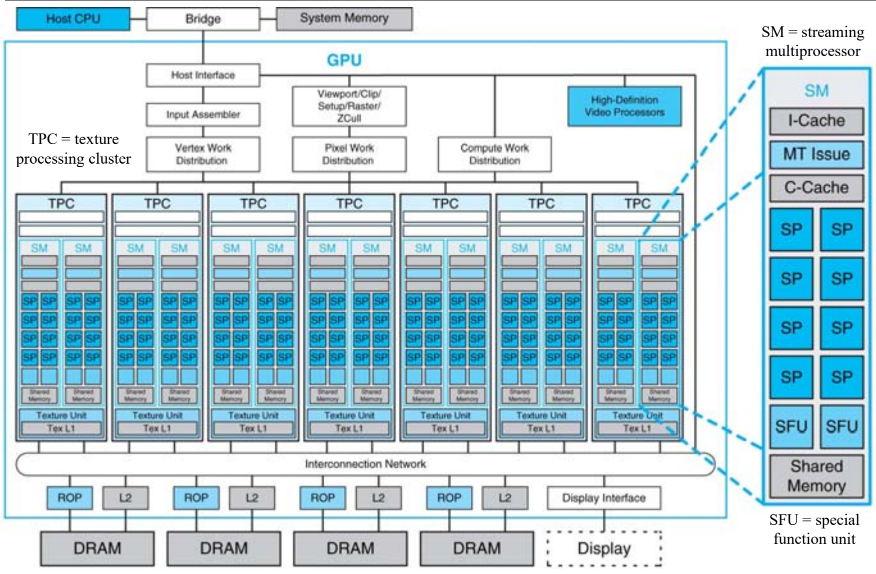
Contemporary PC architecture





- More recently, processor instructions and memory hardware were added to support general-purpose programming languages
- OpenGL: A standard specification defining an API for writing applications that produce 2D and 3D computer graphics
- CUDA (compute unified device architecture): A scalable parallel programming model and language for GPUs based on C/C++

Basic GPU architecture



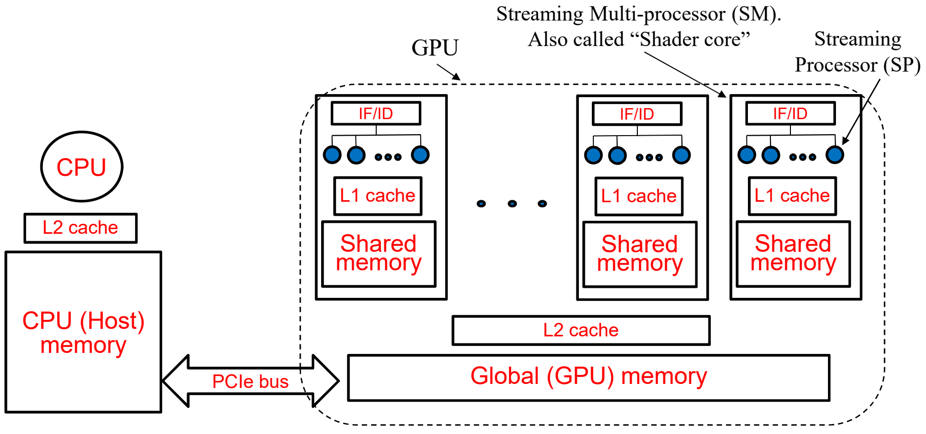
SM = streaming multiprocessor

TPC = texture processing cluster

SFU = special function unit

ROP = raster operations pipeline

The CPU+GPU architecture

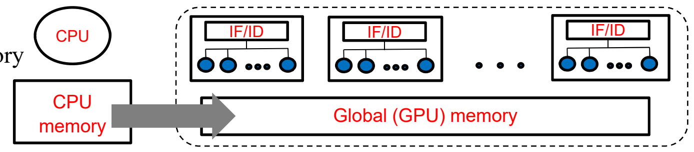


The programming model

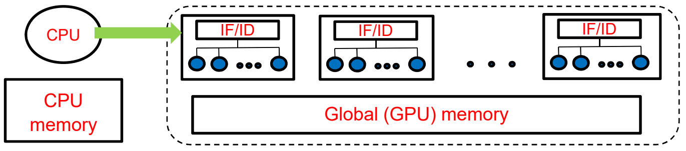
```
    total_hits =0;
    sample_points_per_thread = sample_points /num_threads;

    for (int i=0; i< num_threads; i++){
        my_arg[i].t_seed = i;        /* can chose any seed – here i is chosen*/
        pthread_create (&p_threads[i], &attr, compute_pi, &my_arg[i]);
    }

    for (i=0; i< num_threads; i++){
        pthread_join (p_threads[i], NULL);
        total_hits += my_arg[i].hits;
    }

    computed_pi = 4.0*(double) total_hits / ((double) (sample_points));
}
```

The programming model

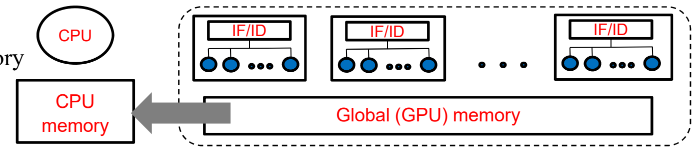
Copy data from CPU memory to GPU memory



Launch the kernel



Copy data from GPU memory to CPU memory


CPU
IF/ID
IF/ID
IF/ID
CPU memory
Global (GPU) memory