A comparison of GPU architectures

- Disclaimer #1: the following slides describe "how I think" about the NVIDIA GTX 280, ATI Radeon 4870, and Intel Larrabee GPUs

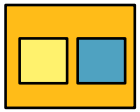
- Disclaimer #2: Many other factors play a role in actual chip performance

- These slides use the same hand-wavy notion of "core" as I established in the talk "From Shader Code to a Teraflop: How a Shader Core Works" in the SIGGRAPH 2008 Course "Beyond Programmable Shading: Fundamentals"

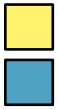
- Remember: you can substitute the term vertex, primitive, CUDA thread, compute shader thread, or OpenCL work item, for "fragment" (pick your favorite language)

GPU block diagram key

= single "physical" instruction stream fetch/decode (functional unit control)



= SIMD programmable functional unit (FU), control shared with other functional units. This functional unit may contain multiple 32-bit "ALUs"



= 32-bit mul-add unit
= 32-bit multiply unit

= execution context storage

= fixed function unit

NVIDIA GeForce GTX 280

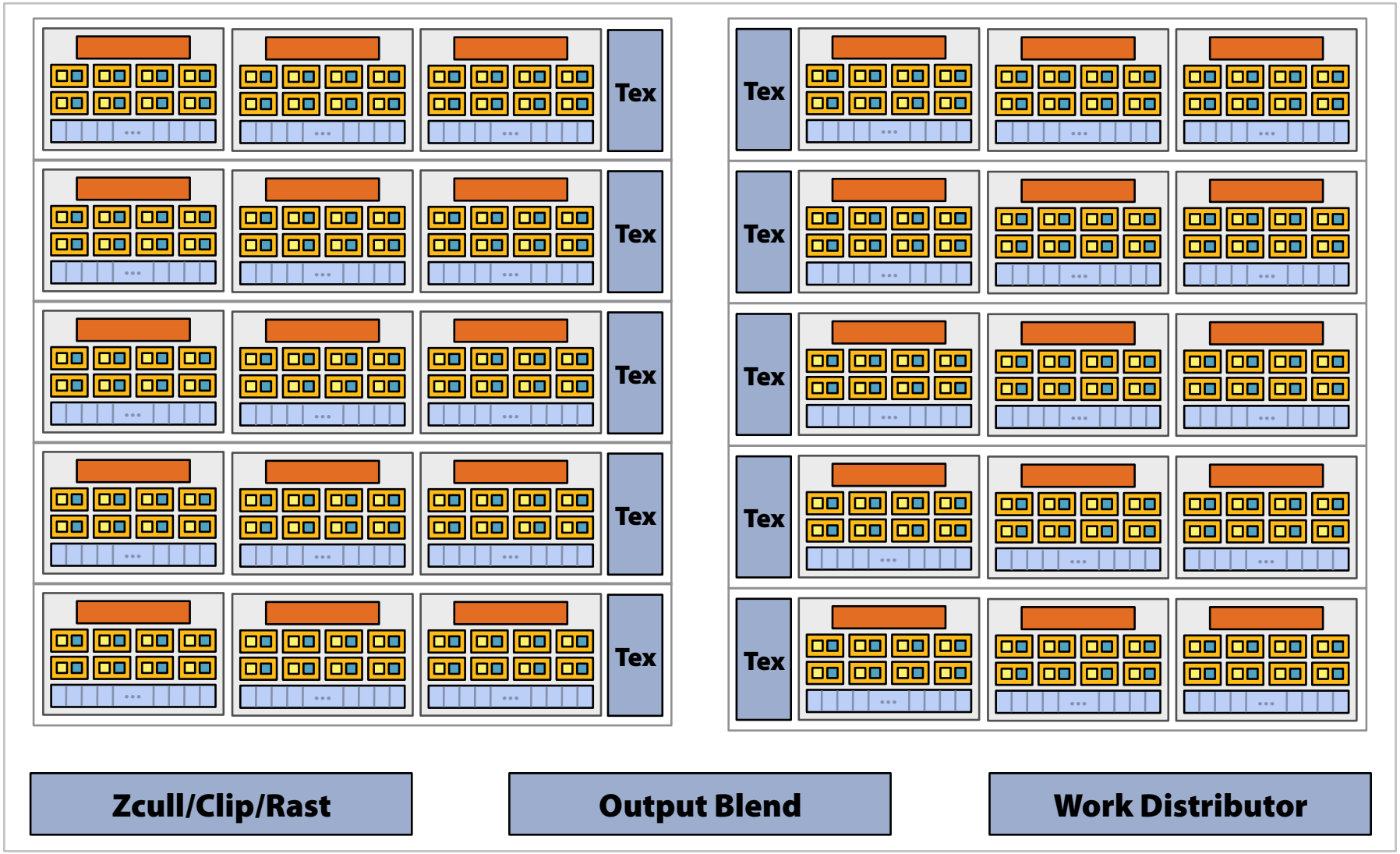
- NVIDIA-speak:
  - 240 stream processors
  - "SIMT execution" (automatic HW-managed sharing of instruction stream)

- Generic speak:
  - 30 processing cores
  - 8 SIMD functional units per core
  - 1 mul-add (2 flops) + 1 mul per functional units (3 flops/clock)
  - Best case: 240 mul-adds + 240 muls per clock
  - 1.3 GHz clock
  - 30 * 8 * (2 + 1) * 1.3 = 933 GFLOPS

- Mapping data-parallelism to chip:
  - Instruction stream shared across 32 fragments (16 for vertices)
  - 8 fragments run on 8 SIMD functional units in one clock
  - Instruction repeated for 4 clocks (2 clocks for vertices)

NVIDIA GeForce GTX 280

SIGGRAPH2008



AMD Radeon 4870

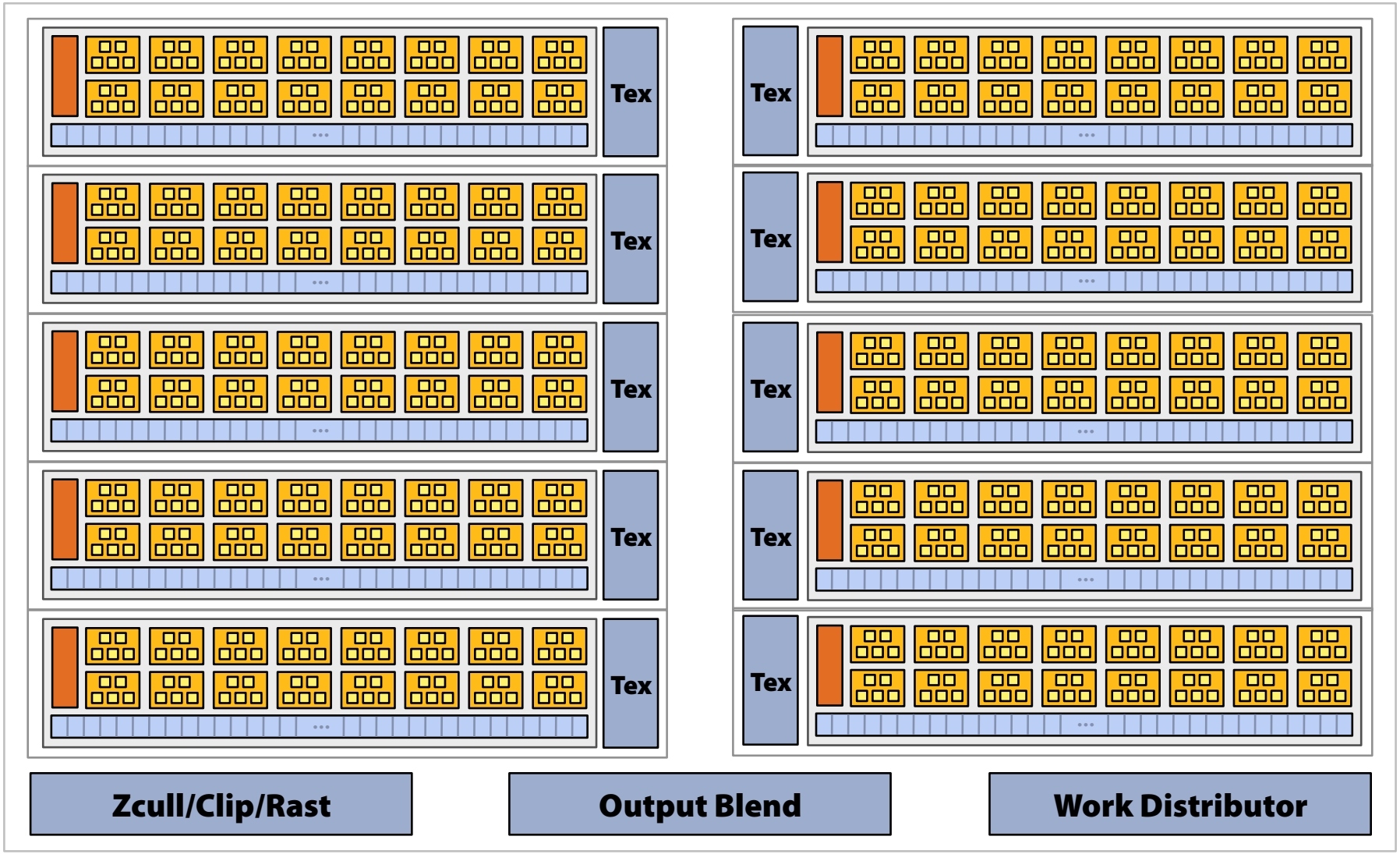
- AMD/ATI-speak:
  - 800 stream processors
  - Automatic HW-managed sharing of scalar instruction stream (like “SIMT”)

- Generic speak:
  - 10 processing cores
  - 16 SIMD functional units per core
  - 5 mul-adds per functional unit (5 * 2 =10 flops/clock)
  - Best case: 800 mul-adds per clock
  - 750 MHz clock
  - 10 * 16 * 5 * 2 * .75 = 1.2 TFLOPS

- Mapping data-parallelism to chip:
  - Instruction stream shared across 64 fragments
  - 16 fragments run on 16 SIMD functional units in one clock
  - Instruction repeated for 4 consecutive clocks

AMD Radeon 4870

SIGGRAPH2008



Intel Larrabee

- Intel speak:
  - We won't say anything about the number of cores or clock rate of cores
  - Explicit 16-wide vector ISA
  - If 1GHz clock (then 1 core = 1 LRB unit = 32 GFLOPS from paper)

- Generic speak:
  - X processing cores
  - 16 SIMD functional units per core
  - 1 mul-add per functional unit (2 flops/clock)
  - Best case: 16X mul-adds per clock
  - If you wanted to compete with current GPUs (~ 1 TFLOP), you need about 32 Larrabee units
    - 32 * 16 * 2 = 1 TFLOP

- Mapping data-parallelism to chip:
  - Compilation options determine instruction stream sharing across fragments (a multiple of 16)
  - 16 fragments run on 16 SIMD functional units in one clock

Intel Larrabee (SIGGRAPH paper)