Quiz 4 Review
GPU & Transactional memory

Guowei Zhang

Lab 4

- Request a deadline extension till Wednesday 13 midnight if needed

Quiz 4 logistics

- Time: 1pm on Tuesday May 12

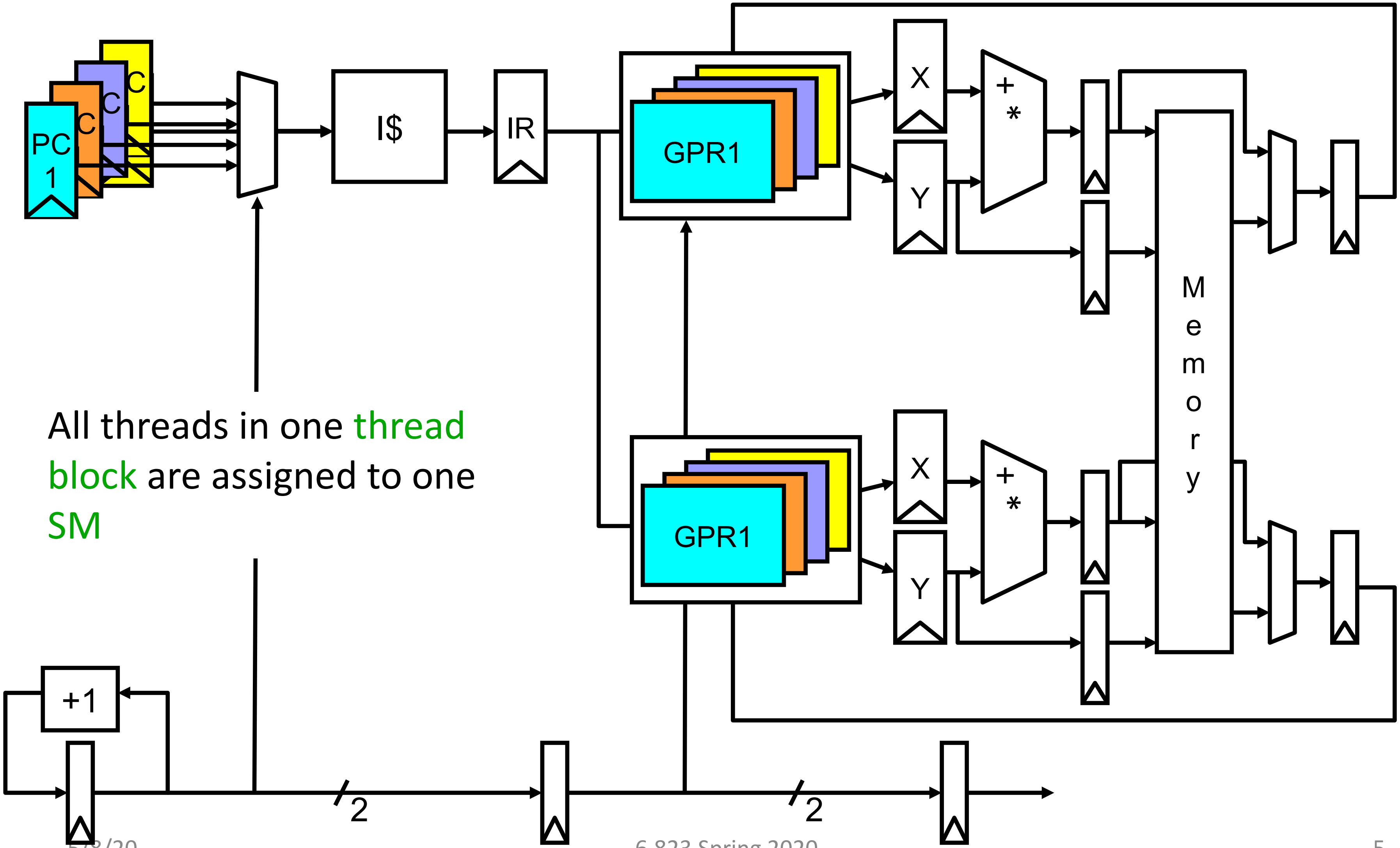
- Style: same as Quiz 3

- Zoom link: same as recitations

Topics

- Microcoded and VLIW processors
- Vector processing
- GPUs
- Transactional memory

GPU pipeline



All threads in one thread block are assigned to one SM

GPU memory system

- Memory types (with different scopes)
  - Per-thread memory
  - Scratchpad shared memory
  - Global memory

- Memory primitives: gathers and scatters

- Efficient code requires reducing conflicts

GPU caches

- Goal: saving bandwidth instead of reducing latency
  - Also enables data compression

- Allows flexible and power-efficient designs

Transactional memory

- Use speculation to provide atomicity and isolation without losing concurrency

- Properties of transactions
  - Atomicity (all or nothing)
  - Isolation
  - Serializability

- Declarative synchronization
- System implements synchronization

Advantages of TM

- Easy-to-use synchronization
- High performance
- Composability

TM implementation

- Choices
  - Hardware transactional memory (HTM)
  - Software transactional memory (STM)
  - Hybrid transactional memory

- Basic implementation
  - Version management
  - Conflict detection
  - Conflict resolution

Version management

- Eager versioning
  - Undo-log based
  - Fast commits and slow aborts

- Lazy versioning
  - Write-buffer based
  - Slow commits and fast aborts

Conflict detection

- Read-write and write-write conflicts

- Pessimistic detection
  - Checks during loads/stores
  - Typical resolution: requester wins/stalls
  - Detects conflicts early
  - Requires more to guarantee forward progress

- Optimistic detection
  - Checks when attempting to commit
  - Typical resolution: committer wins
  - Guarantees forward progress (still has fairness issues)
  - Detects conflicts late

HTM implementation

- Version management: use caches
  - Caching write-buffer or undo-log
  - Tracking read-set and write-set

- Conflict detection: use the cache coherence protocols

- Pros:
  - Low implementation overheads
  - Simplifies consistency
- Cons:
  - Performance pathologies
  - Capacity limitations
  - Interaction with Irrevocable execution
  - …

Wish you all the best!