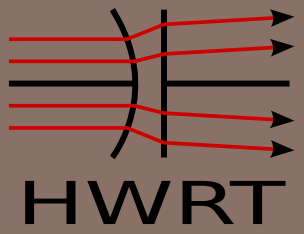
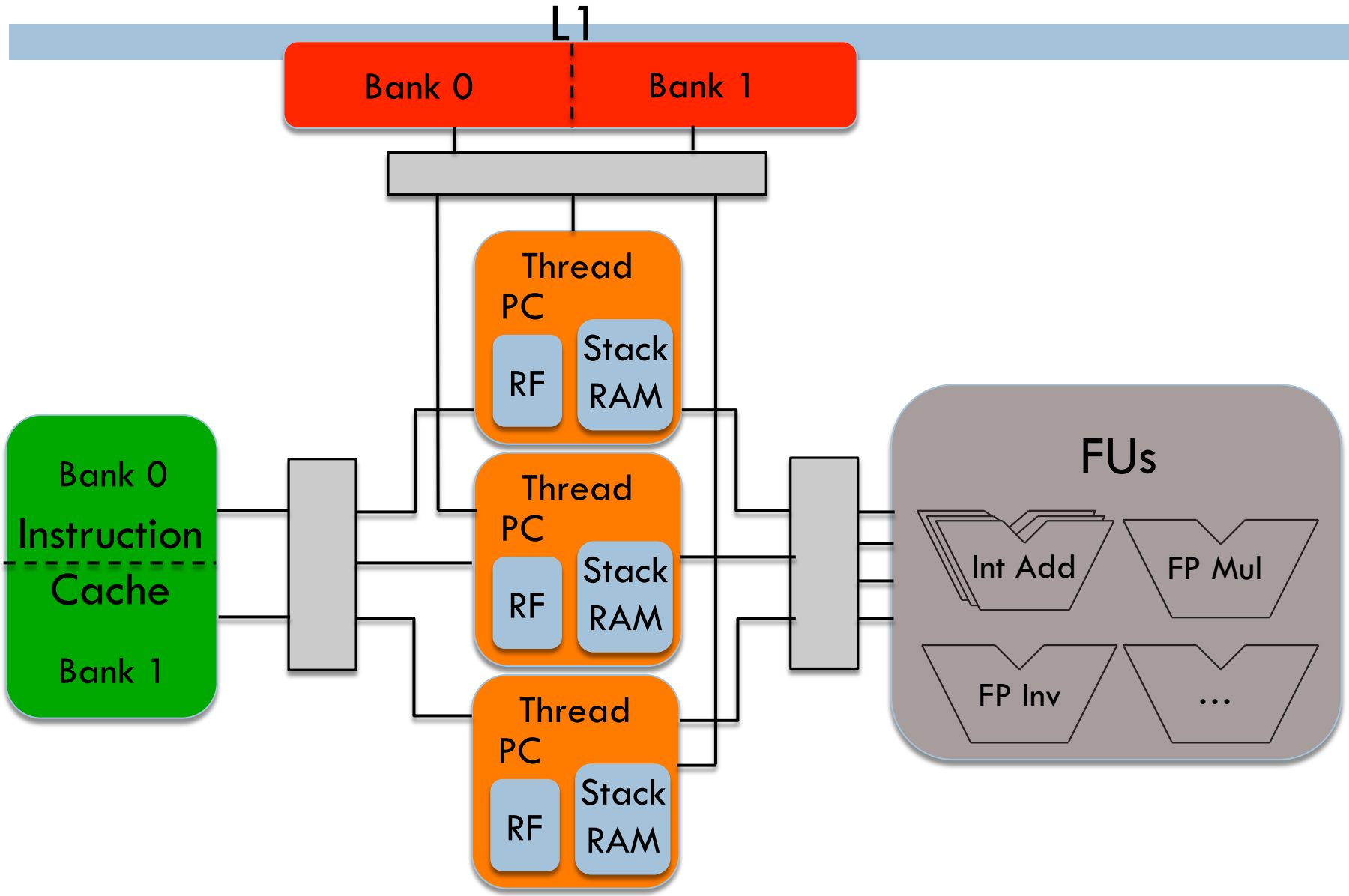


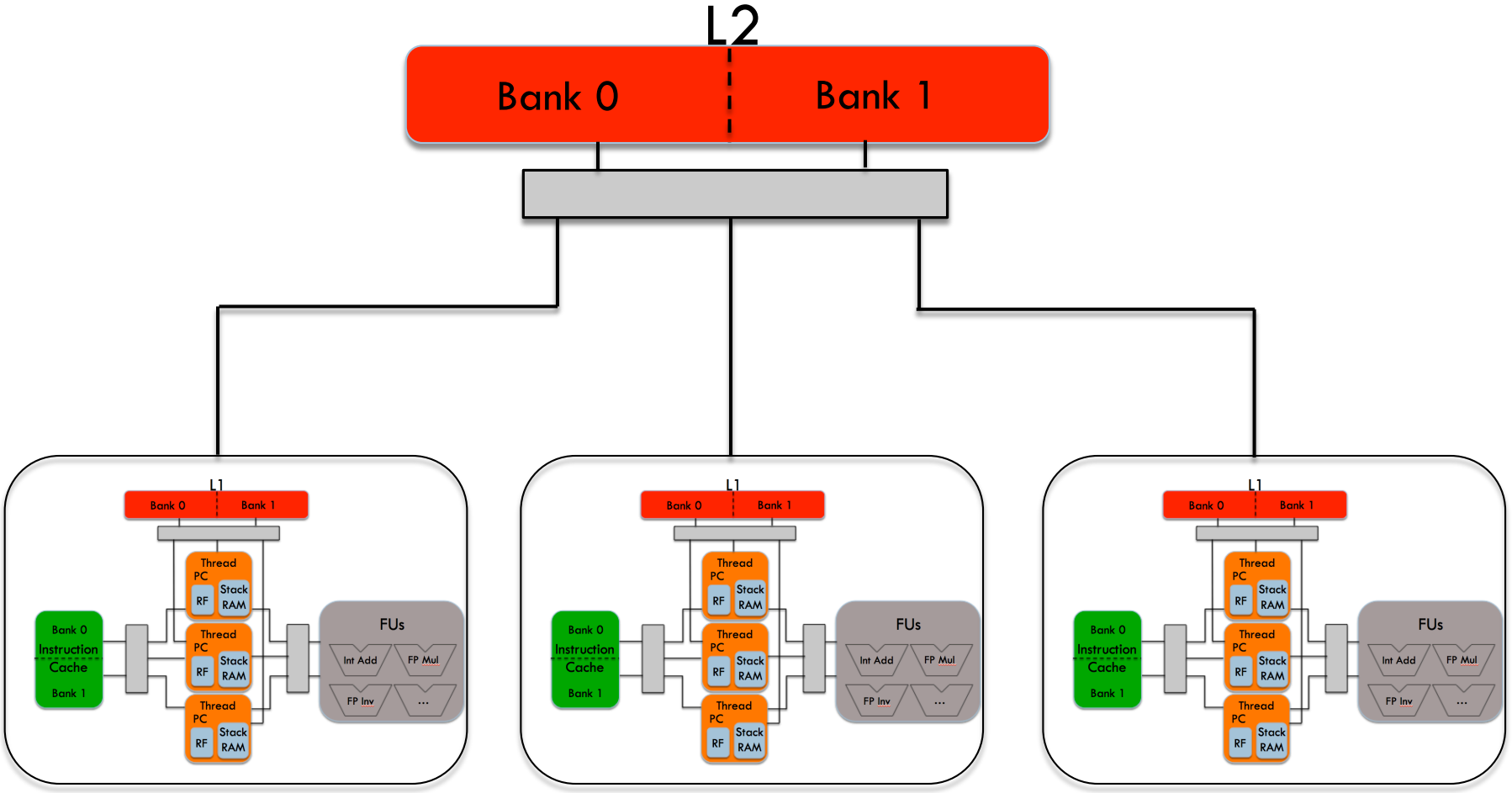
CS 6958
USIMM
PROJECT PHASE

March 5, 2014

Single TM


L1
Bank 0
Bank 1
Thread
PC
RF
Stack
RAM
Thread
PC
RF
Stack
RAM
Thread
PC
RF
Stack
RAM
Bank 0
Instruction
Cache
Bank 1
FUs
Int Add
FP Mul
FP Inv
…

--num-TMs

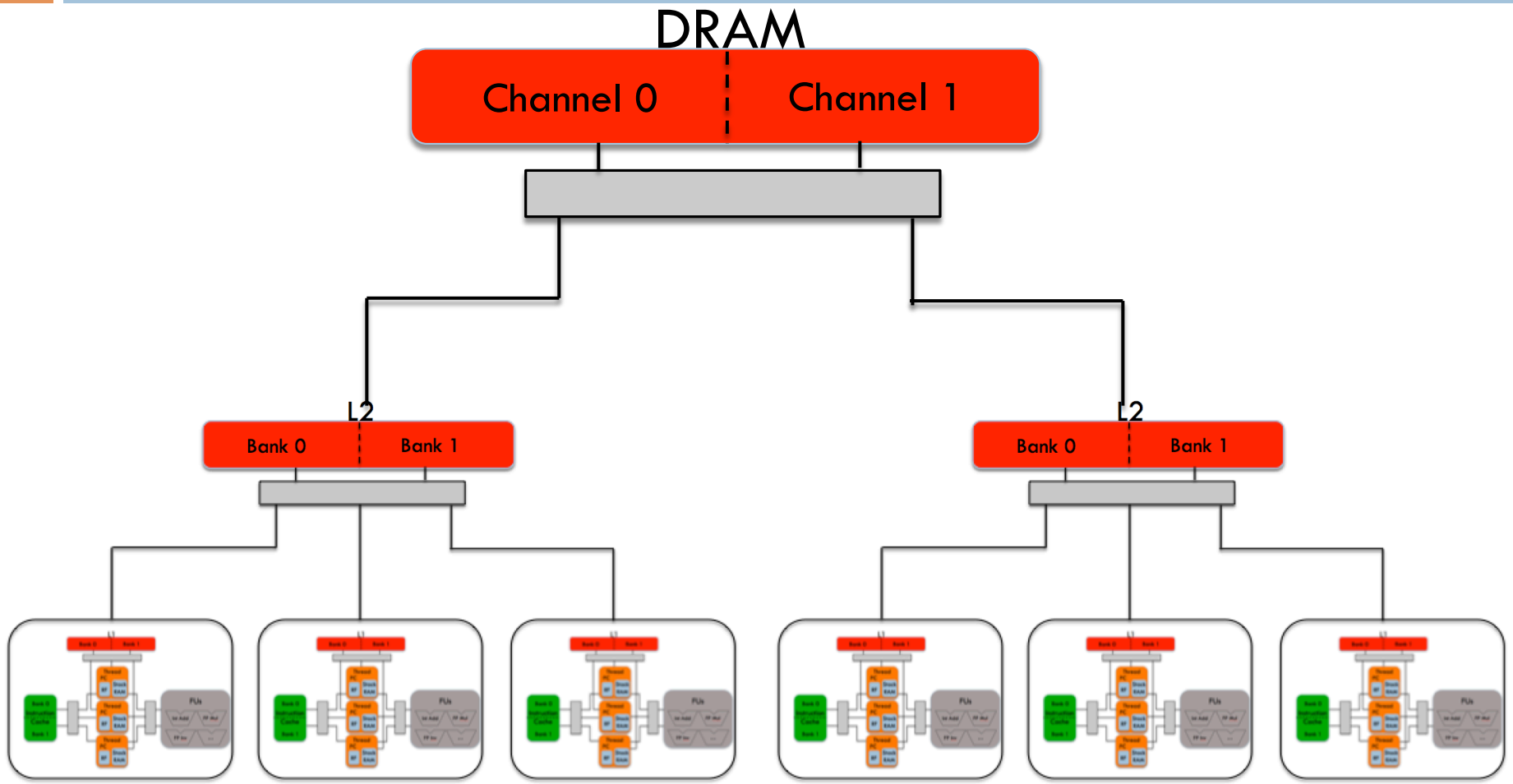

L2
Bank 0
Bank 1
L1
Bank 0
Bank 1
Thread
PC
RF
Stack
RAM
Bank 0
Instruction
Cache
Bank 1
FUs
Int Add
FP Mul
FP Inv
...

--num-TMs

- L2 requirements depends on how many accesses pass the L1

- Affected by:
  - Number of TMs connected to L2
  - L1 hit rate of each TM
  - L1 access rate
    - Affected by num threads and num banks

--num-l2s



Full System

- DRAM requirements similarly affected by:
  - Number of L2s
  - L2 access rate
  - L2 hit rate
- Aside from full design-space exploration, what can we do?
  - Pick a good TM
  - Then pick a good L2/num TMs
  - Then pick a good num L2s
  - Tweak…

Full System

- Number of TMs:
  - --num-TMs * --num-l2s
- Number of threads:
  - number of TMs * --num-thread-procs

--simulation-threads <N>

- *Attempts* to parallelize the simulator itself
  - Only works on > 1 TM
- TMs must synchronize on every cycle, and mutex every L2 access
  - Parallel scaling is not too great
  - Recommend 8 threads at most

USIMM

- Utah Simulated Memory Module
- Does two things:
  - Slows the simulator down a lot
  - Makes the simulator more accurate (a lot)
- Overhead is proportional to #cycles
  - More threads = fewer cycles, overhead becomes reasonable

USIMM Output

- Non-intuitive items:
  - Total reads/writes serviced = total cache lines transferred
    - != total loads/stores (coalescing)
  - Page Hit Rate = row buffer hit rate
  - Avg. column reads per ACT = How many reads to an open row before closing it
  - Single column reads = how many times was a row opened for just 1 read (worst case)

USIMM Output

- Energy/Power reported in two places:
  - Energy: along with all other energy numbers
  - Power: after per-channel stats

- Why does USIMM draw power even with no LOAD/STORE?
  - DRAM refresh
  - Energy consumed is a function of activity + running time (background energy)

USIMM Default Config

- 16 channels
- 16 banks
  - = 256 total row buffers
- 8KB rows
- 64B lines
- 2x TRaX clock (2GHz)
  - = 512GB/s peak
- Max queue length = 80 (per channel)

Address Mapping

- Two policies implemented
  - See configs/usimm_configs/gddr5_8ch.cfg
    - ADDRESS_MAPPING <0 or 1>

Policy	Most significant bit	…	…	Least significant bit
0	Row	Column	Bank	Channel
1	Row	Bank	Channel	Column

- Neither is inherently better
  - What matters is compatibility with access patterns

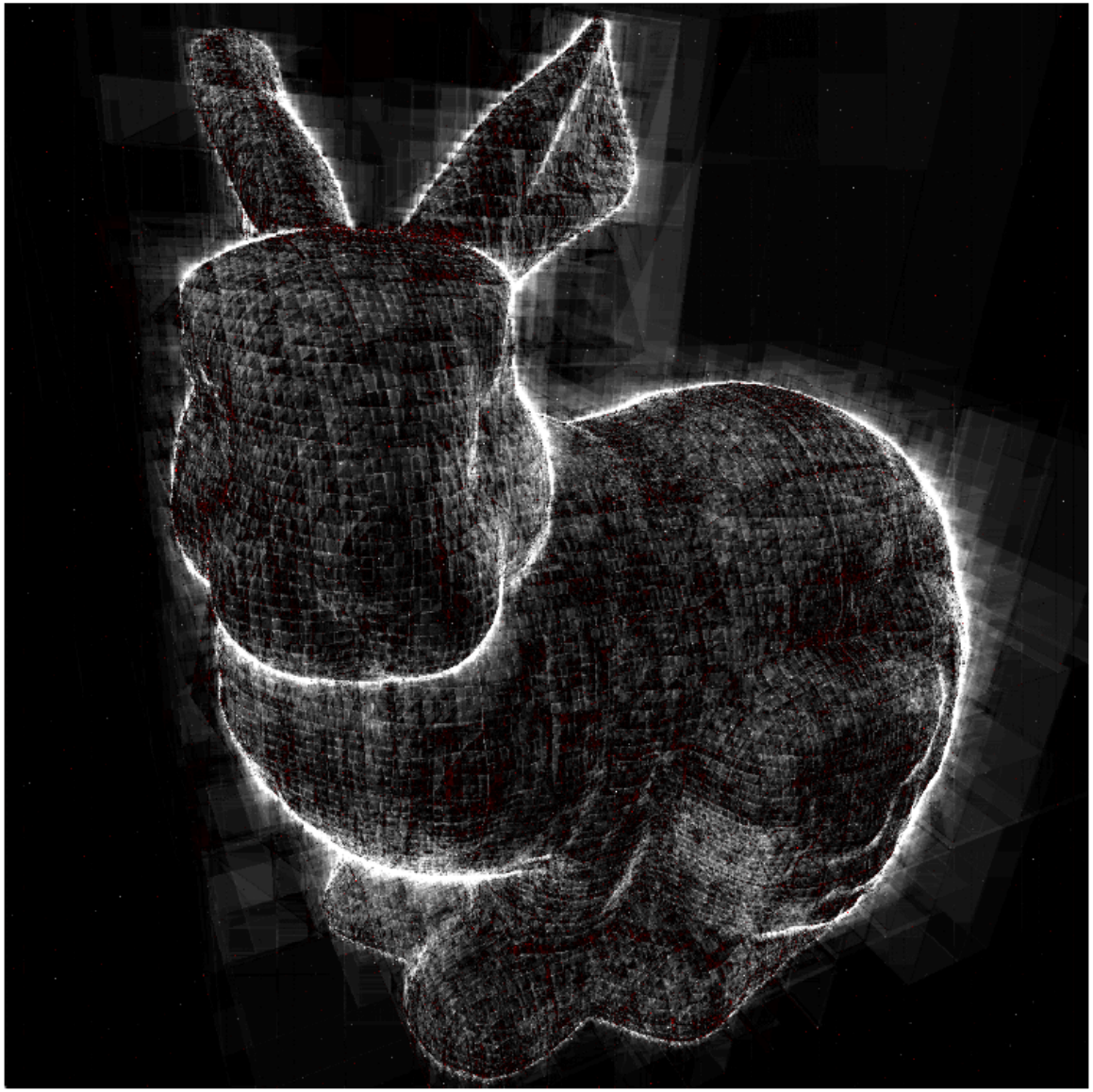
Final Projects

1. Proposal
   - Short description/proposal document
   - 5 minute introduction presentation
2. Weekly short status report
   - What have you achieved this week?
   - Where are you stuck, how can we help?
3. Midpoint report
   - 5 minute progress/future direction presentation
4. Final report
   - Project analysis and documentation
   - 10 minute final presentation

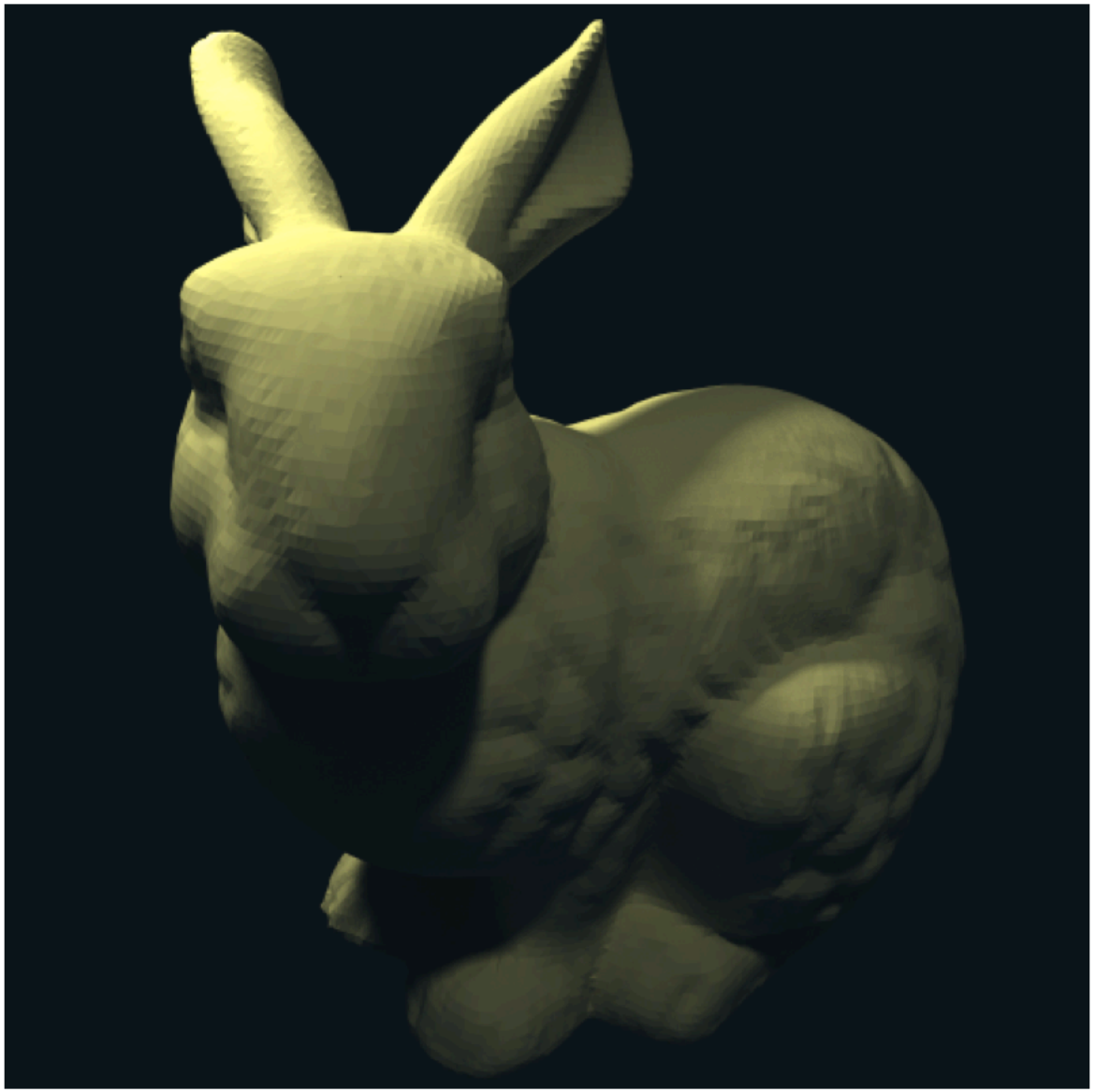
Final Projects

- Must be substantial
  - We will approve your proposal document
- Must be interesting/useful
  - Something we haven't already done
- Can focus on HW/SW or either
  - HW focus must analyze on interesting SW benchmark
  - SW focus must analyze HW requirements

Pitch 1 – Visual Analysis Suite



Visual Analysis



Visual Analysis

- Performa a full high quality rendering, but display something else about each pixel
  - Cache hit rates
  - Bandwidth consumption
  - Stack traffic
  - Row buffer hit rate
  - Resource stalls/data stalls
- Composites of 2 or more of the above may be very revealing
- Draw per-box heat map instead of per-pixel?

Visual Analysis

Visual Analysis

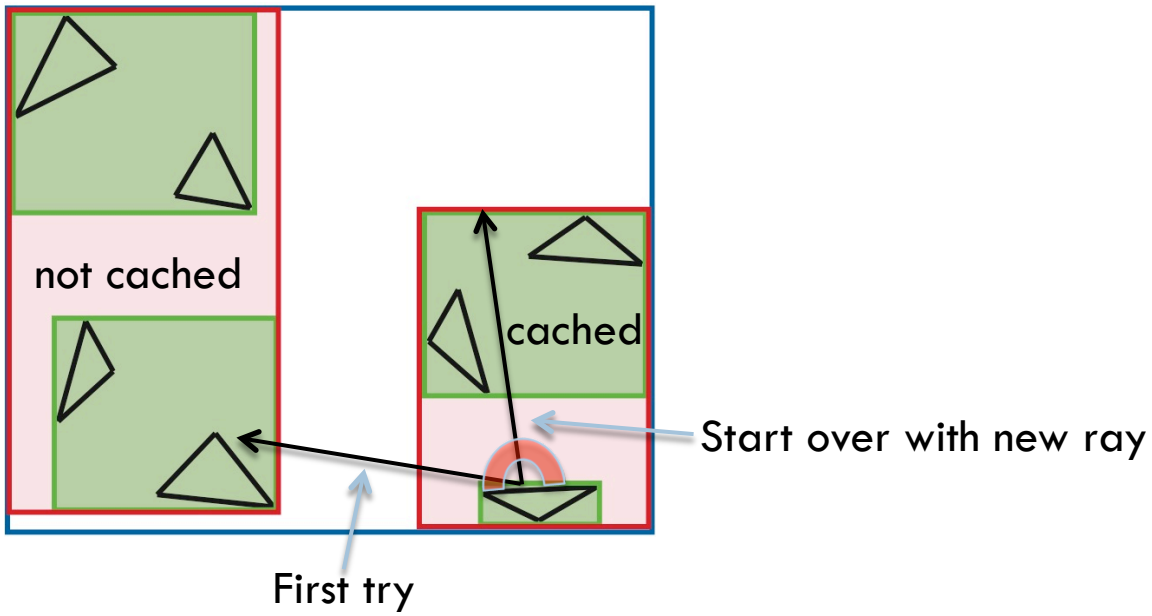
Visual Analysis

Pitch 2 – Cache Aware Computing

- TRaX has special “loadl1” and “loadl2” instructions
  - Returns whether or not certain address is cached
- As a programmer, use this to re-order computation
  - Or alter the algorithm altogether

Cache Aware (i.e. Path Tracing)

- Direction of any given indirect ray not too important
- Favor rays traveling in a "cached" direction
  - Quantize and limit bias



- Determine good restart heuristic

Pitch 3 – Cache Upgrade

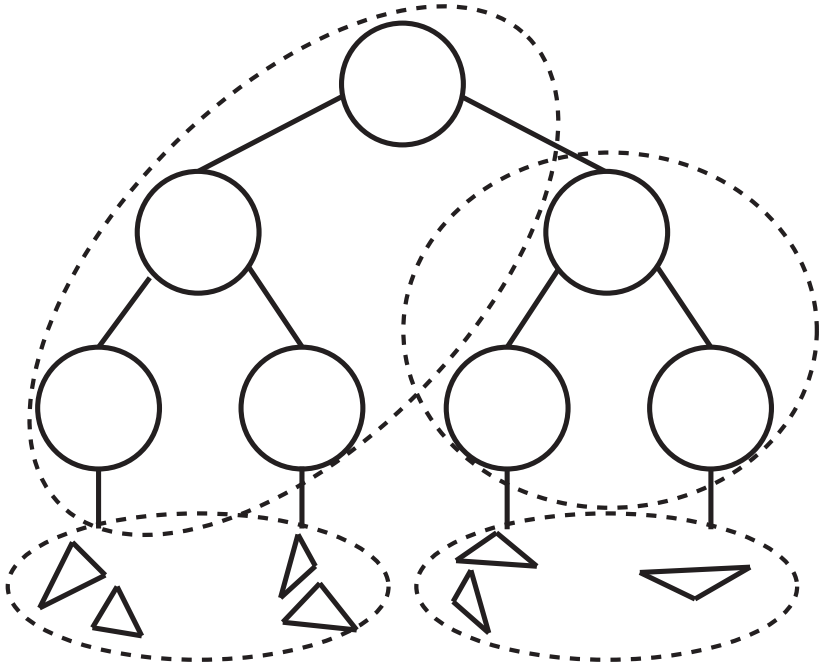
- Associativity
- Victim caches
- RT-aware caches
  - Box cache
  - Triangle cache (odd line size)
  - Material cache (odd line size, low pressure)
  - Prefetching

Pitch 4 - DRAM

- Row buffer friendly data layout
  - and/or row buffer friendly access patterns
  - i.e. rearrange BVH/traversal order
- Address mapping policies
- Memory controller algorithms
  - RT-aware scheduling

DRAM Pitch (i.e. access patterns)

- If ray leaves current “row buffer region”, pause processing until later



Pitch 5 – Cache Coherence

- Currently, simtrax models write-only or read-only
  - Caches are write-around
  - read-after-write behaves correctly, but reports fake performance
- Add correct modeling
  - Caches need to signal each other to invalidate lines
  - Could add new instructions:
    - Read-around
    - Write-through
    - Write-around
    - …

Final Projects

- Keep in mind you have a simulator
  - Way more info available than a CPU program
- You can do "anything" you want
  - Add new instructions
  - Add new HW units
  - Add new memories, change memory controller
  - Instrument new stats gathering