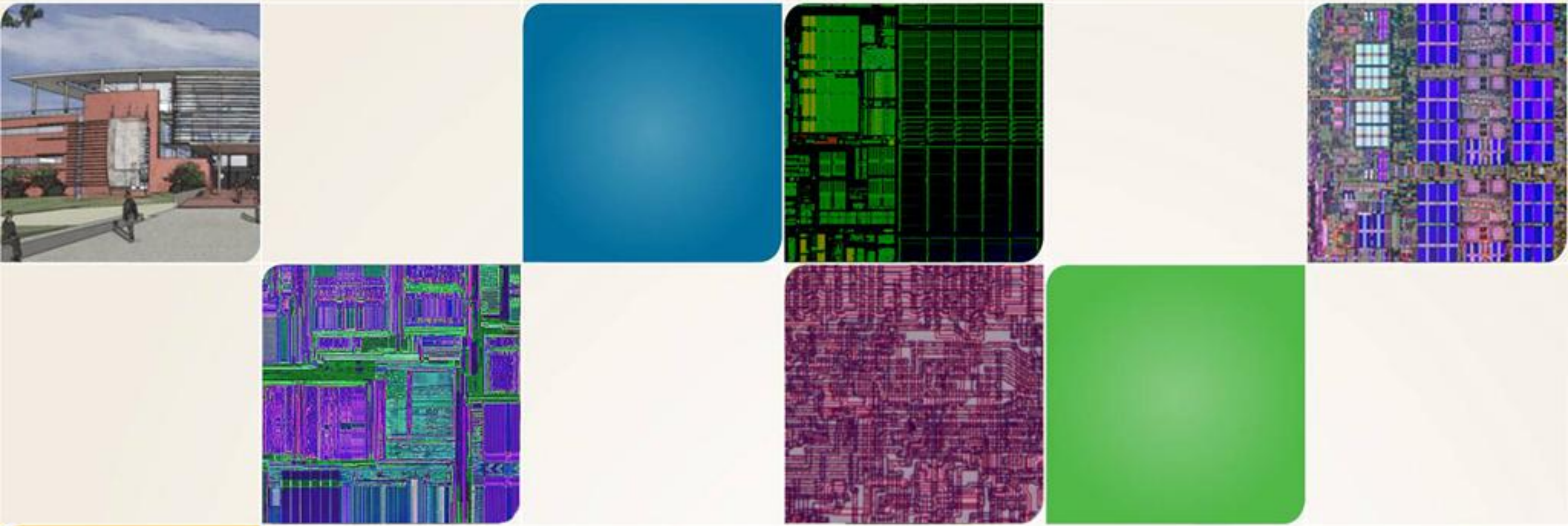


CS4803DGC Design Game Console

Spring 2010

Prof. Hyesoon Kim

Georgia Tech College of Computing

Review for Quiz-II

- G80 architecture
- CUDA programming
- Graphics pipelining
  - Differences between model space vs. world space? Camera space?
  - Basic concepts
    - Graphics pipeline stages
    - Z-buffer, Stencil, Anti-aliasing, clipping, culling

Questions

- Which memory would be good to communicate within a block?
- Only ready-to-go warps can be dispatched to SPs. Which unit decides which warps are ready?
- Why can G80 synchronize threads within a block but not across a block?
- What is a divergent branch in CUDA? Why does it hurt performance? How can we reduce the number of divergent branches?
- Why is it generally a good idea to increase the number of working threads in CUDA?
- What are the benefits of hardware based Z-depth test unit ? Why many filters are implemented in the hardware even in unified shaders?
- Why constant and texture caches are read only but shared memory is read/write?
- The benefit of AOS over SOA?

COMPUTING QUESTIONS

G80 warp instruction fetch

- There are 5 blocks and each block has 64 threads. The kernel has 3 instructions. How many instructions are fetched? 32 threads are executed together.

Questions

Consider a 1GHz G80 architecture. Global memory access time is 400 CPU cycles. DRAM bandwidth is 80GB/s. One instruction takes 4 cycles to issue. A warp size is 32 threads. Global memory access time 400 CPU cycles are additional to the 4 cycle memory issue cycle.

```
#define POS(x,y) … // define the position for a global memory location
// globalM is the global memory structure
for (int ii = 0; ii < 10000; ii++) {
    sharedM[threadIdx.x] = globalM[POS(threadIdx.x, blockIdx.x)];
    // load 2B per thread
    aa = sharedM[threadIdx.x];
    bb = aa +10.00;
}
```

Find the minimum number of thread to get the peak GFLOPS for the code. There is only one block of threads. (We can assume that one block can have infinite number of threads.)

Questions

Consider a 1GHz G80 architecture. Global memory access time is 400 CPU cycles. DRAM bandwidth is 20GB/s. One instruction takes 4 cycles to issue. A warp size is 32 threads.

```
#define POS(x,y) … // define the position for a global memory location
// globalM is the global memory structure
for (int ii = 0; ii < 10000; ii++) {
    sharedM[threadIdx.x] = globalM[POS(threadIdx.x, blockIdx.x)];
    // load 2B per thread
    aa = sharedM[threadIdx.x];
    bb = aa +10.00;
}
```

Find the minimum number of thread to get the peak GFLOPS for the code. There is only one block of threads. (We can assume that one block can have infinite number of threads.)

VLIW

- An ATI like GPU architecture has 3-wide VLIW SIMD execution unit. The possible VLIW configurations are

  (1) INT, FP, BR

  (2) INT, INT, FP

  (3) INT, FP, FP

  (4) INT, INT, BR

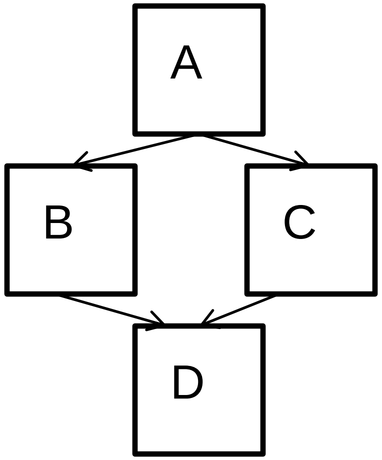
  Arrange the code so that the code can be executed in the processor.

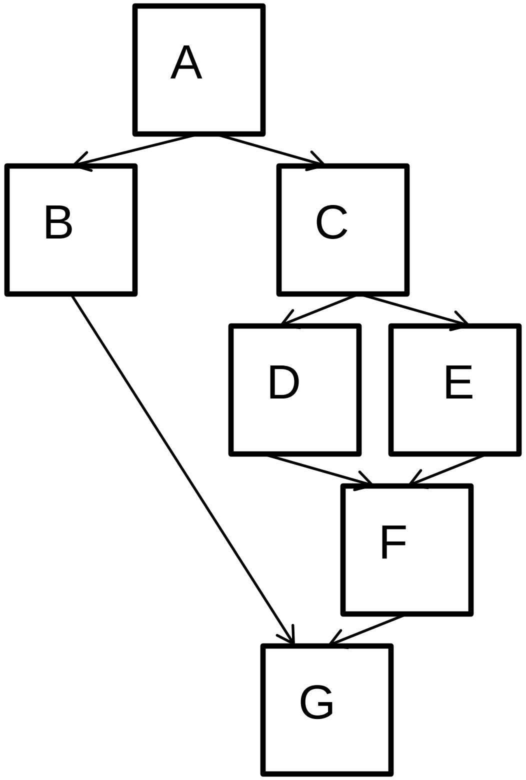
```
e.g.) ADD R1, R2, R3,
ADD R8, R6, R7
ADD R5, R6, R1
BR (R1) TARGET

TARGET MUL F10, F12, F13
       MUL F20,F12, F13
```

Post Dominators

- Which one is the control-flow merge points?

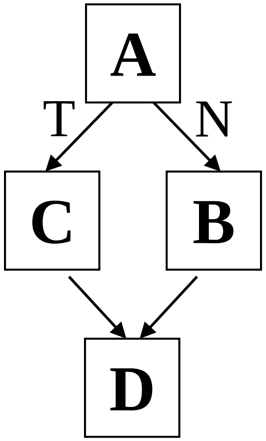




Predicated Code vs. Branch Code

```
if (cond) {
      b = 0;
}
else {
      b = 1;
}
```

(normal branch code)

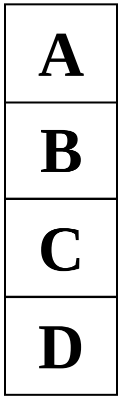


```
A  p1 = (cond)
   branch p1, TARGET

B  mov b, 1
   jmp JOIN

C  TARGET:
       mov b, 0
```

(predicated code)



```
A       p1 = (cond)

B  (!p1) mov  b, 1

C   (p1) mov  b, 0
```

- Questions: The branch direction is TTNTTNT.
- Total number of fetched instructions if the code is normal branch code vs. predicated code?

Coalescing vs. Uncolaescing

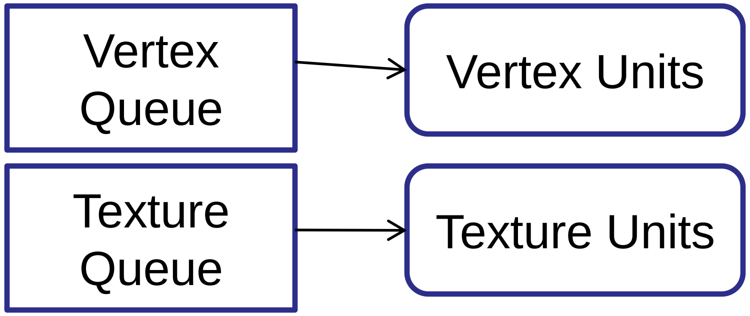
- Sequential memory addresses can be coalesced but not for non-sequential addresses. Assume that 128B memory addresses can be coalesced.
- 32-bit address space. Calculate the number of address Bytes that need to be sent from the core and the number of data (Byte) needs to be sent from DRAM for the following two cases. Assume that command information is sent separately (load/write, size)

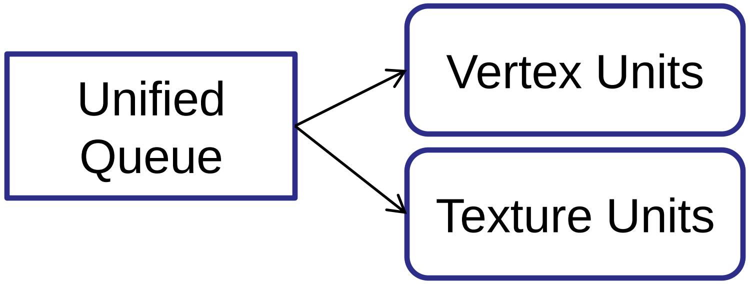
  (a) LDB 1001, LDB1002, LDB1003, LDB1004, LDB1005 LDB 1006, LDB 1007, LDB1008,

  (b) LDB 1001, LDB 1003, LDB 1004, LDB 101a, LDB 1020, LDB1030, LDB 1040, LDB 1041

Thread Dispatch Units

- ATI has dedicated arbiter units for texture and vertex fetches. Among the following three cases, when having dedicated units will provide benefits? There are two design options. Having one unified arbiter with a doubled entry sizes or having two unified arbiters. (2 insts/cycle dispatch model)
- Case 1: vertex, vertex, texture, texture (vertex latency 1, texture latency 1)
- Case 2: vertex, vertex, vertex, vertex (vertex latency 1, texture latency 1)
- Case 3: texture, texture, texture, texture (vertex latency 2, texture latency 1)
- Case 4: vertex, texture, vertex, texture (vertex latency 2, texture latency 1)





FP Precisions

- The latest ATI processors has the following performance for single and double precision.
  - 2.7 TeraFlop/s for a single precision
  - 544 Gflop/s for a double precision
- There is a software emulation method to calculate double precisions from single precision unit. What if a double precision ADD requires only 3 single precision computations and 10 integer instructions. What is the minimum int/s in order to software emulation would be better than hardware double precision computation?

Announcement

- 3/15/10 Quiz-II, bring a calculator and buzzcard
- Lab #4's deadline is March 19th.
- Friday, please bring your windows notebook. (at least one for each group.)