CS 152, Spring 2011
Section 10

Christopher Celio

University of California, Berkeley
Agenda

- Stuff (Quiz 4 Prep)

Intel Core 2 Duo (Penryn) Vs. NVidia GTX 280

- Intel Core 2 Duo (Penryn)
  - dual-core
  - 2007+
  - 45nm
  - 410 million transistors
  - ~2GHz
  - 3 or 6MB of cache
  - 10-35 Watts
  - 107mm²
    - each core is 22mm²
    - L2 SRAM is 6mm²/MB

- NVidia GTX 280
  - 10 core(?) (240 “stream” processors)
  - 2008
  - 65nm
  - 1.4 Billion transistors
  - 576mm²
  - 602 MHz(core clock)
  - 236 Watts !!!


Friday, April 8, 2011
Quiz 4

- VLIW
  - (for real this time)
  - able to write assembly for VLIW
- software
  - instruction re-ordering
  - loop unrolling
  - software pipelining
  - how code will get scheduled on different pipelines
  - conditional execution (for VLIW, vector, and GPU)
  - types of parallelism (ILP, TLP, DLP)
- Vector processors
  - able to write vector assembly (including how to strip-mine loops!)
  - chaining
- Multithreading
  - fine-grain, course-grain, SMT
- GPUs/SIMT model
  - how do they handle conditional execution/branches?
    - (spoiler alert: branch divergence)
VLIW: Very Long Instruction Word

- Multiple operations packed into one instruction
- Each operation slot is for a fixed function
- Constant operation latencies are specified
- Architecture requires guarantee of:
  - Parallelism within an instruction => no cross-operation RAW check
  - No data use before data ready => no data interlocks

**Note:** Iron Law questions about CPI are about counting the *instructions*, not the individual ops
Loop Unrolling

for (i=0; i<N; i++)

Unroll inner loop to perform 4 iterations at once

for (i=0; i<N; i+=4)
{
}

Need to handle values of N that are not multiples of unrolling factor with final cleanup loop
Software Pipelining
Loop Execution

for (i=0; i<N; i++)

How many FP ops/cycle?

1 fadd / 8 cycles = 0.125
Software Pipelining

```c
for (i=0; i<N; i++)

loop:  ld f1, 0(r1)
       add r1, 8
       fadd f2, f0, f1
       sd f2, 0(r2)
       add r2, 8
       bne r1, r3, loop
```

How does one do software pipelining?

Let’s run through an example that does software pipelining WITHOUT loop unrolling.
Software Pipelining

for (i=0; i<N; i++)

Compile

loop: ld f1, 0(r1)
    add r1, 8
    fadd f2, f0, f1
    sd f2, 0(r2)
    add r2, 8
    bne r1, r3, loop

March 14, 2011  CS152, Spring 2011
Software Pipelining

for (i=0; i<N; i++)

Compile

loop:  ld f1, 0(r1)
    add r1, 8
    fadd f2, f0, f1
    sd f2, 0(r2)
    add r2, 8
    bne r1, r3, loop

How many FLOPS/cycle?
1 fadds / 4 cycles = 0.25

<table>
<thead>
<tr>
<th>Int1</th>
<th>Int 2</th>
<th>M1</th>
<th>M2</th>
<th>FP+</th>
<th>FPx</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

March 14, 2011

CS152, Spring 2011
Pset 4, Question 4 (Vector Processors)
Problem 2: Vector
Vector machines often have a lot of memory bandwidth (SX-9 has 256GB/s!). Why do they need it and why do current superscalars not provide as much?