CS 61C
Spring 2010

I/O & Performance

Scott Beamer (cs61c-ta)

Week 14 (4/27)

Polling vs. Interrupts

Operation	Definition	Pro/Con	Good for
Polling	repeatedly check if ready to do I/O	+ fast if there is work to do + easy to implement	regular or high bandwidth transfers
Interrupts	do I/O when an event occurs	+ only does work if there is work - slow per unit of data	irregular or unpredictable events

CPU Performance

- *Iron Law (or Performance Equation)*: CPU time = (# instructions)(CPI)(Clock Period)
  - *# instructions*: how many instructions executed during program (insts/program)
  - *CPI (Cycles per Instruction)*: avg number of cycles to complete an instruction
  - *Clock Period*: Inverse of clock frequency

Performance Problem

- What is the CPI of our pipelined MIPS processor if it has forwarding, every branch delay slot is filled with useful work, and there are no load hazards?
  1
- What if only half of the branch delay slots contain useful work and branches occur about 20% of the time?
  0.8x1 + 0.1x1 + 0.1x2 = 1.1

Operation Type	Frequency	CPI for CPU A	CPI for CPU B
ALU	50%	2	1
Load	20%	4	2
Store	10%	2	1
Branch	20%	1	1

- If CPU A and B have the same clock speed, how many times faster is B?
  CPIB/CPIA = (0.5x2 + 0.2x4 + 0.1x2 + 0.2x1)/(0.5x1 + 0.2x2 + 0.1x1 + 0.2x1) = 2.1/1.2

- Clearly CPUA is bottlenecked by not having a cache. What would the CPI of Loads and Stores need to be to match the performance of B?
  0,but thats impossible
- By only changing the clock speed, how many times faster would that need to be for CPU A to match the performance of B?
  2.1/1.2