CPT_S 260 Intro to Computer Architecture
Lecture 3

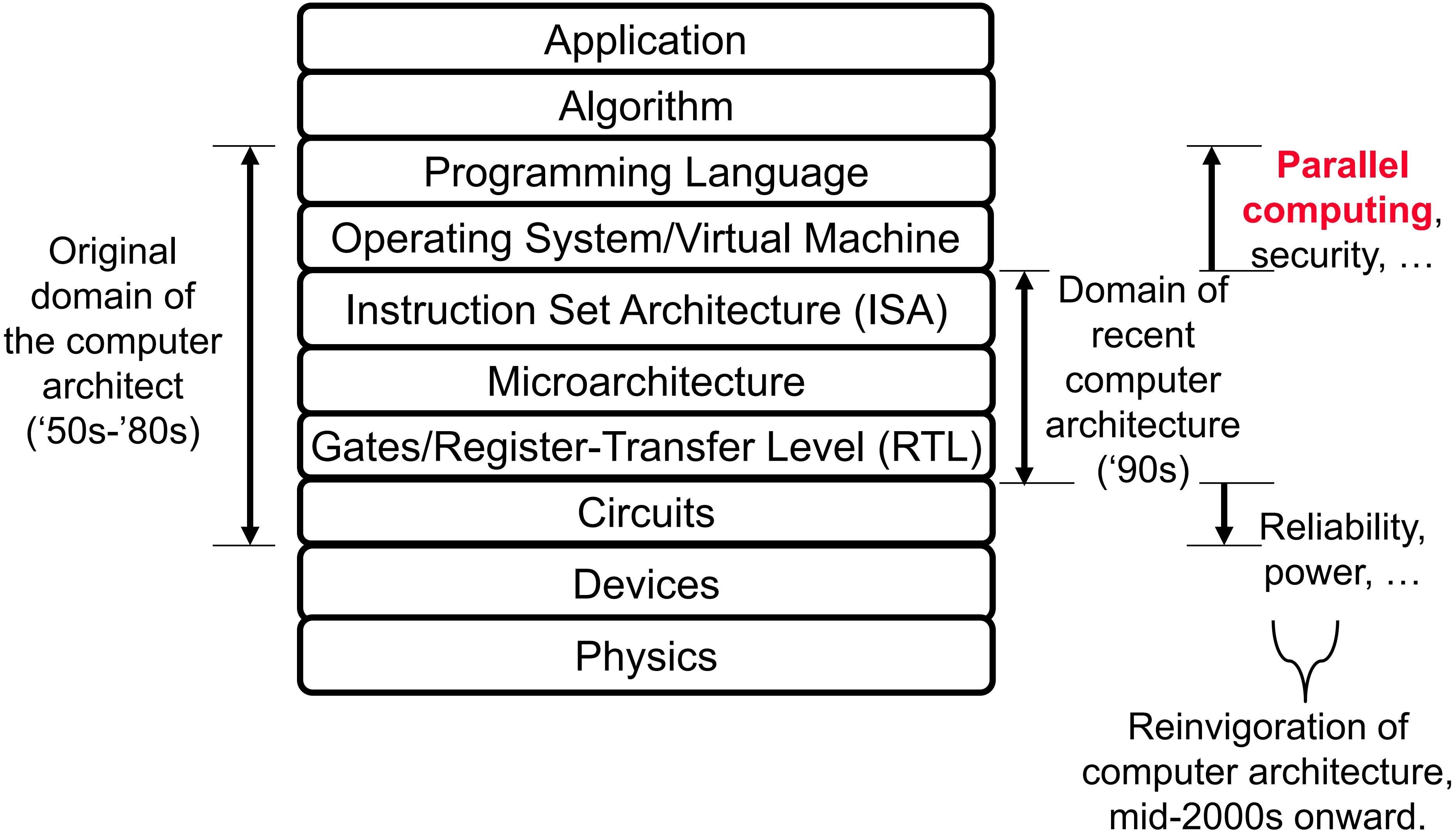
Computer Performance
January 19, 2022

Ganapati Bhat
School of Electrical Engineering and Computer Science
Washington State University

Announcements

- **Course schedule has been uploaded**
- **Assignment 1 is online**
  - Performance evaluation
  - Number representation
  - Arithmetic

Recap: Abstraction Layers in Modern Systems



Recap: Response Time and Throughput

- **Response time – Execution Time**
  - How long it takes to do a task
- **Throughput**
  - Total work done per unit time
    - » e.g., tasks/transactions/… per hour
- **How are response time and throughput affected by**
  - Replacing the processor with a faster version?
  - Adding more processors?

CPU Time

$$\text{CPU Time} = \#\,\text{CPU Clock Cycles} \times \text{Clock Cycle Time}$$
$$= \frac{\#\,\text{CPU Clock Cycles}}{\text{Clock Rate}}$$

- **Performance improved by**
  - Reducing number of clock cycles
  - Increasing clock rate
  - Hardware designer must often trade off clock rate against cycle count

Instruction Count and CPI

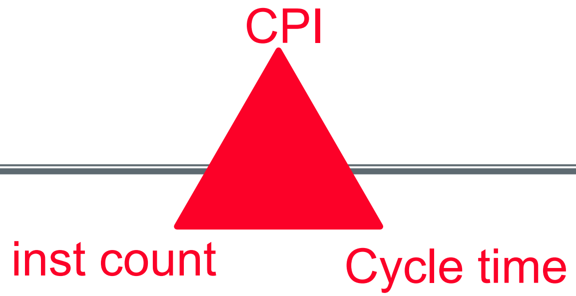
$$\text{Clock Cycles} = \text{Instruction Count} \times \text{Cycles per Instruction}$$

$$\text{CPU Time} = \text{Instruction Count} \times \text{CPI} \times \text{Clock Cycle Time}$$

$$= \frac{\text{Instruction Count} \times \text{CPI}}{\text{Clock Rate}}$$

- **Instruction Count for a program**
  - Determined by program, ISA and compiler
- **Average cycles per instruction**
  - Determined by CPU hardware
  - If different instructions have different CPI
    - Average CPI affected by instruction mix

Computer Performance



$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

	Inst Count	CPI	Clock Rate
Program	X		
Compiler	X	(X)	
Inst. Set.	X	X	
Organization		X	X
Technology			X

CPI Example

- Computer A: Cycle Time = 250ps, CPI = 2.0
- Computer B: Cycle Time = 500ps, CPI = 1.2
- Same ISA
- Which is faster, and by how much?

CPI Example

- **Computer A: Cycle Time = 250ps, CPI = 2.0**
- **Computer B: Cycle Time = 500ps, CPI = 1.2**
- **Same ISA**
- **Which is faster, and by how much?**

$$\begin{aligned}\text{CPU Time}_A &= \text{Instruction Count} \times \text{CPI}_A \times \text{Cycle Time}_A \\ &= I \times 2.0 \times 250\text{ps} = I \times 500\text{ps}\end{aligned}$$

A is faster…

$$\begin{aligned}\text{CPU Time}_B &= \text{Instruction Count} \times \text{CPI}_B \times \text{Cycle Time}_B \\ &= I \times 1.2 \times 500\text{ps} = I \times 600\text{ps}\end{aligned}$$

$$\frac{\text{CPU Time}_B}{\text{CPU Time}_A} = \frac{I \times 600\text{ps}}{I \times 500\text{ps}} = 1.2$$

…by this much

CPI in More Detail

- **If different instruction classes take different numbers of cycles**

$$\text{Clock Cycles} = \sum_{i=1}^{n} (\text{CPI}_i \times \text{Instruction Count}_i)$$

- **Weighted average CPI**

$$\text{CPI} = \frac{\text{Clock Cycles}}{\text{Instruction Count}} = \sum_{i=1}^{n} \left(\text{CPI}_i \times \underbrace{\frac{\text{Instruction Count}_i}{\text{Instruction Count}}}_{\boxed{\text{Relative frequency}}} \right)$$

CPI Example

- Alternative compiled code sequences using instructions in classes A, B, C

Class	A	B	C
CPI for class	1	2	3
IC in sequence 1	2	1	2
IC in sequence 2	4	1	1

- **Sequence 1: IC = 5**
  - **Clock Cycles**
    $= 2\times1 + 1\times2 + 2\times3$
    $= 10$
  - **Avg. CPI = 10/5 = 2.0**

- **Sequence 2: IC = 6**
  - **Clock Cycles**
    $= 4\times1 + 1\times2 + 1\times3$
    $= 9$
  - **Avg. CPI = 9/6 = 1.5**

Comments on Previous Example

- **This example focuses on comparing code segments.**
- **We would like to know**
  – Which code sequence executes the most instructions?
  – Which code sequence will be faster?
  – What is the CPI for each code sequence?
- **Sequence 2 executes more instructions (6) compared to sequence 1 with only 5 instructions.**
- **Code sequence 2 is faster even though it executes one extra instruction. The reason is that code sequence 2 requires less CPU clock cycles (9 cycles) compared to sequence 1 with 10 cycles.**
- **It is clear now that because code sequence 2 takes fewer overall clock cycles but has more instructions, it must have a lower CPI.**
- **The CPI for each code sequence is computed in previous slide.**
  – CPI for sequence 1 = 2.0
  – CPI for sequence 2 = 1.5

Performance Summary

$$\text{CPU Time} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock cycle}}$$

- **Performance depends on**
  - Algorithm: affects IC, possibly CPI
  - Programming language: affects IC, CPI
  - Compiler: affects IC, CPI
  - Instruction set architecture: affects IC, CPI, T_c

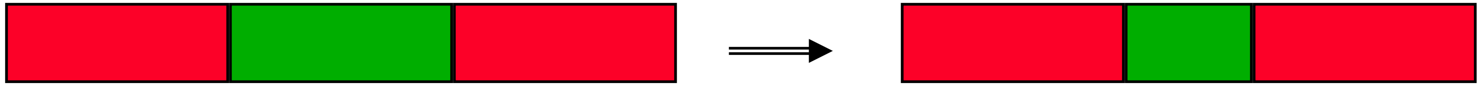
Amdahl's Law

- **How do we increase performance?**
  - Utilize parallelism
  - Principle of locality
  - Focus on the common case
- **Amdahl's law provides a method to quantify speedup**

$$Speedup_{overall} = \frac{t_{old}}{t_{new}} = \frac{1}{(1 - fraction_{enhanced}) + \frac{fraction_{enhanced}}{speedup_{enhanced}}}$$

- **Best achievable speedup is**

$$Speedup_{maximum} = \frac{1}{1 - fraction_{enhanced}}$$



Amdahl's Law

- **Improving an aspect of a computer and expecting a proportional improvement in overall performance**

$$T_{\text{improved}} = \frac{T_{\text{affected}}}{\text{improvement factor}} + T_{\text{unaffected}}$$

- **Example: multiply accounts for 80s/100s**
  » How much improvement in multiply performance to get 5X overall (5 times improvement in overall performance)?

$$20 = \frac{80}{n} + 20$$

Can't be done!

- **Corollary: make the common case fast**

Example on Amdahl's Law

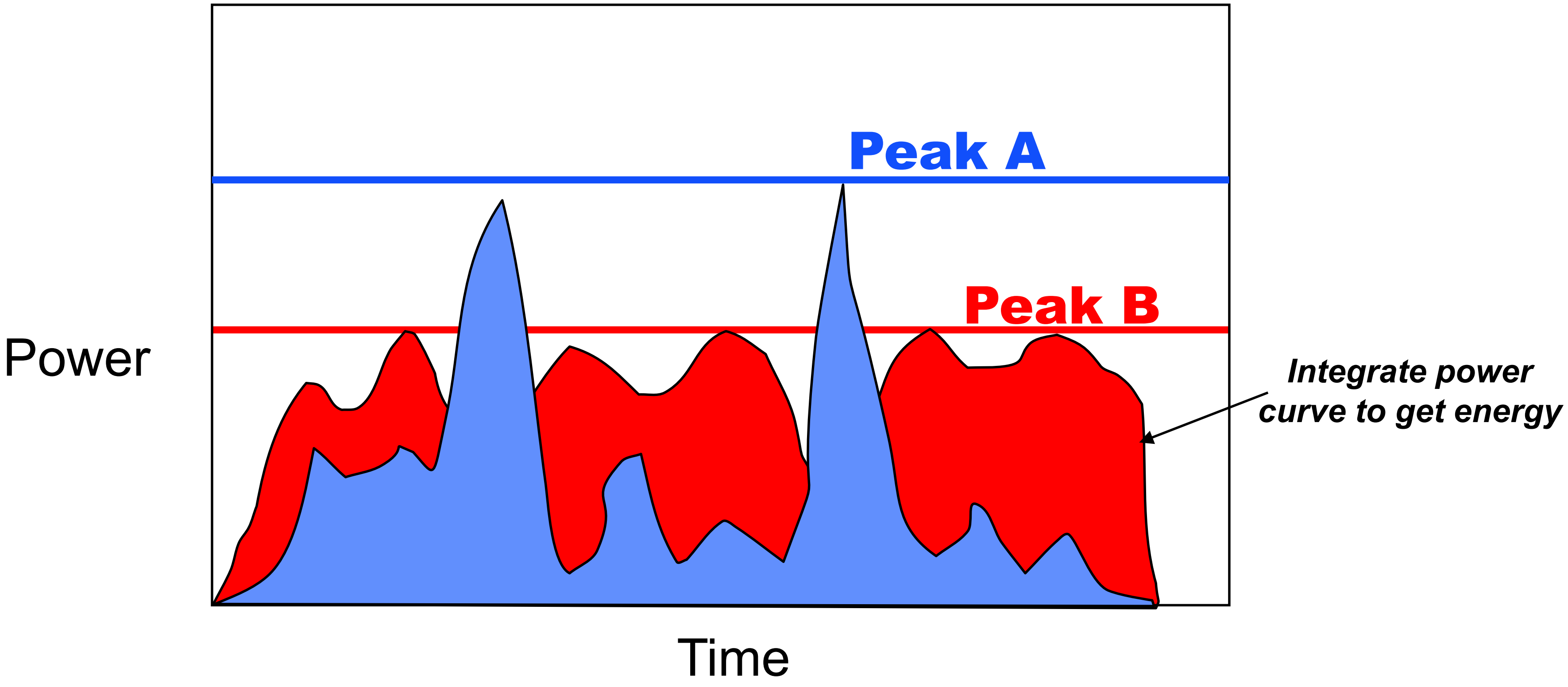
- **Problem**
  - Suppose that we can improve the floating point instruction performance of a machine by a factor of 15 (the same floating point instructions run 15 times faster on this new machine). What percent of the instructions must be floating point to achieve a Speedup of at least 4?
- **Solution**
  - Let x be percentage of floating point instructions.
  - Since the speedup is 4, if the original program executed in 100 cycles, the new program runs in 100/4 = 25 cycles.
  - $(100)/4 = (x)/15 + (100 - x)$
  - Solving for x, we get: $x = 80.36$
  - The percent of floating point instructions need to be 80.36.

Power and Energy

- **Energy to complete operation (Joules)**
  - Corresponds approximately to battery life
  - (Battery energy capacity actually depends on rate of discharge)
- **Peak power dissipation (Watts = Joules/second)**
  - Affects packaging (power and ground pins, thermal design)
  - Thermal considerations determine the peak power -> TDP
- **di/dt, peak change in supply current (Amps/second)**
  - Affects power supply noise (power and ground pins, decoupling capacitors)
- **Components of power consumption**

$$\begin{aligned} P &= P_{dyn} + P_{static} \\ &= \alpha C V^2 f + I_{leak} V \end{aligned}$$

Peak Power versus Lower Energy



- **System A has higher peak power, but lower total energy**
- **System B has lower peak power, but higher total energy**

Example #5 on Amdahl's Law

- **Problem**
  - Suppose a program segment consists of a purely sequential part which takes 25 cycles to execute, and an iterated loop which takes 100 cycles per iteration. Assume the loop iterations are independent, and cannot be further parallelized. If the loop is to be executed 100 times, what is the maximum speedup possible using an infinite number of processors (compared to a single processor)?
- **Solution**
  - The sequential part takes 25 cycles and will remain unaffected. Each iteration of the loop (which takes 100 cycles) can be executed independently and there are totally 100 iterations. Thus,
    - Original (before improvement) execution time = (100*100) + 25 = 10025
  - Now let's apply Amdahl's law:
    - Execution time after improvement = (Execution time of the affected code)/(Amount of improvement in affected code) + Execution time of unaffected code = (100*100)/100 + 25 = 100 + 25 = 125
  - Thus,
    - Speedup = 10025/125 = 80.2

Terms

- Response time - How long it takes to do a task
- Throughput - Total work done per unit time
- Execution time - Total response time for a program
- CPU time - Time spent processing a given job
- Performance - 1 / Execution time
- CPU clocking - Operation of digital hardware governed by a constant-rate clock
- Clock period - Duration of a clock cycle
- Clock frequency - Cycles per second
- Instruction Count (IC) - Instructions for a program
- Cycles Per Instruction (CPI) - Average cycles per instruction
- Amdahl's Law - Improving an aspect of a computer and expecting a proportional improvement in overall performance

Example on CPU Time

- **Problem**

A student in CptS 260 runs a program in 5s on her computer that has a 1.5GHz clock. She breaks her computer in frustration one evening and replaces it with the same model but with a slightly better clock of 2GHz. The program runs in 4s on the replacement computer. Which computer ran the program in fewest clock cycles and by how many cycles?

- **Solution**

The given quantities have units of seconds and cycles per second in GHz and the calculated quantity has units of cycles.

#Clock cycles = CPU time × Clock rate

Original computer : 5s × 1.5GHz = 7.5×10^9 cycles

Replacement computer : 4s × 2GHz = 8×10^9 cycles

The replacement computer ran in $(8 – 7.5) \times 10^9$ more cycles.

That is, 0.5×10^9 more cycles.

Example on CPI & CPU Time

- **Problem**

  A student in CptS 360 runs a program with an instruction count of 1 million over 1 milliseconds on a computer that has a 2GHz clock. What's the CPI?

- **Solution**

  The given quantities have units of instructions, milliseconds, and cycles per second in GHz. The calculated quantity has units of cycles per instruction.

$$\text{Clock cycles} = \text{CPU time} \times \text{Clock rate} = \text{IC} * \text{CPI}$$

$$\begin{aligned}\text{CPI} &= (\text{CPU time} \times \text{Clock rate}) / \text{IC} \\ &= (1\text{ ms} \times 2\text{GHz}) / 1{,}000{,}000 \text{ instructions} \\ &= (1 \times 10^{-3} \times 2 \times 10^{9}) / (1 \times 10^{6}) \text{ cycles per instruction} \\ &= (2 \times 10^{6}) / (1 \times 10^{6}) \text{ cycles per instruction} \\ &= 2 \text{ cycles per instruction}\end{aligned}$$

Example on Weighted CPI

- **Problem**

  A student in CptS 460 has written a program containing 460 instructions, with the first 20% of instructions having CPI of 10^9 and the remainder having CPI of $2*10^9$. What's the weighted average CPI of the program?

- **Solution**

  The given quantities have units of instructions and cycles per instruction. The calculated quantity has units of cycles per instruction.

$$
\begin{aligned}
CPI &= sum(weight*CPI + \ldots + weight*CPI) \\
&= 0.2 * 10^9 + 0.8 * 2*10^9 \\
&= 1.8*10^9 \text{ cycles per instruction}
\end{aligned}
$$