

---

CS152  
Computer Architecture and Engineering  
Discussion #1

January 26, 1995  
Nikunj C. Oza

---

## Formulas for Integrated Circuits

---

$$\text{Die Cost} = \frac{\text{Wafer Cost}}{\text{Dies per wafer} * \text{Die yield}}$$

$$\text{Dies per wafer} = \frac{\pi * (\text{Wafer diameter} / 2)^2}{\text{Die Area}} - \frac{\pi * \text{Wafer diameter}}{\sqrt{2} * \text{Die area}} - \text{Test dies} \approx \frac{\text{Wafer Area}}{\text{Die Area}}$$

$$\text{Die Yield} = \text{Wafer yield} * \left( 1 + \frac{\text{Defects per unit area} * \text{Die Area}}{\alpha} \right)^{-\alpha}$$

Just keep these handy for the homework and the test.

# Processor Metrics

---

- ◆ Execution time = Clock cycles/program \* clock cycle time [s/cycle]
- ◆ Execution time = Clock cycles/program / clock rate [cycles/s]
- ◆ Clock cycles/pgm = Instructions/program \* average clock cycles/instruction
- ◆ CPI (cycles/instruction) = Clock cycles/program / (Instructions/program)
- ◆ CPI = (CPU Time \* Clock Rate) / (Instruction Count)

**Understand** these formulas. Know all the units. You should be able to derive these formulas.

Also, note that the last formula corrects Dave's notes (Lec 2.21).

# Amdahl's Law

---

Speedup due to enhancement E:

$$\text{Speedup due to E} = \frac{\text{Execution time without E}}{\text{Execution time with E}}$$

If the enhancement accelerates a task that takes up a fraction F of the task by a factor of S, then:

$$\text{Execution Time with enhancement} = \text{Execution time without enhancement} * \left( \frac{F}{S} + (1 - F) \right)$$

## Who Are We?

---

Nikunj C. Oza  
oza@cs.berkeley.edu  
[http://http.cs.berkeley.edu/  
~oza](http://http.cs.berkeley.edu/~oza)



Young Hyun Cho  
young@uclink.berkeley.edu



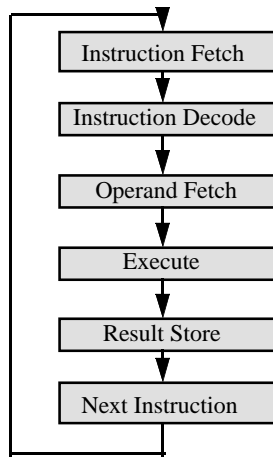
## More Administrative Stuff

---

- ◆ Advice:
  - Read the book BEFORE lecture!
  - Time management (Franklin Day Planner)
- ◆ Information:
  - Structure of Discussion Sections
  - Extra Credit
- ◆ Questions on administrative matters?

## Execution Cycle

---



- ◆ This is a preview of pipelining.
- ◆ Simple instructions are helpful:
  - Pipeline remains full.
  - Less pipeline stopping and/or flushing.

## Why are General Purpose Registers Cool?

---

- ◆ Registers are faster than memory.
- ◆ Memory traffic is reduced, therefore program is faster.
- ◆ Registers are easier to use than a stack because registers can be accessed in any order.
- ◆ Code density improves.

## RISC Argument

---

Have Simple Instructions:

- ◆ They are the ones that mostly get used anyway.
- ◆ Allow greater flexibility.

## Variable vs. Fixed-Length Instructions

---

- ◆ Advantage of variable: smaller code size because no space is wasted.
- ◆ Advantage of fixed: No interpretation of instruction format is needed.