CMPE 200
Computer Architecture & Design

Lecture 1. Computer Architecture & Design Overview (2)

Haonan Wang

SJSU SAN JOSÉ STATE UNIVERSITY

About Prerequisite – What to Submit?

- **If you have 180D in your admission letter (conditionally classified)**
  - You then need to submit your transcript with 180D highlighted
  - You cannot take this course before finishing 180D

- **If you do not have 180D in your admission letter**
  - You then need to submit your admission letter

- **If you are admitted with classified standing in your admission letter**
  - You then need to submit your admission letter

Relative Performance

- **Define Performance = 1/Execution Time**

- **"X is n times faster than Y"**

$$\text{Performance}_X / \text{Performance}_Y$$
$$= \text{Execution time}_Y / \text{Execution time}_X = n$$

Example: time taken to run a program -- 10s on a computer A, 15s on computer B

Execution Time_B / Execution Time_A = 15s / 10s = 1.5

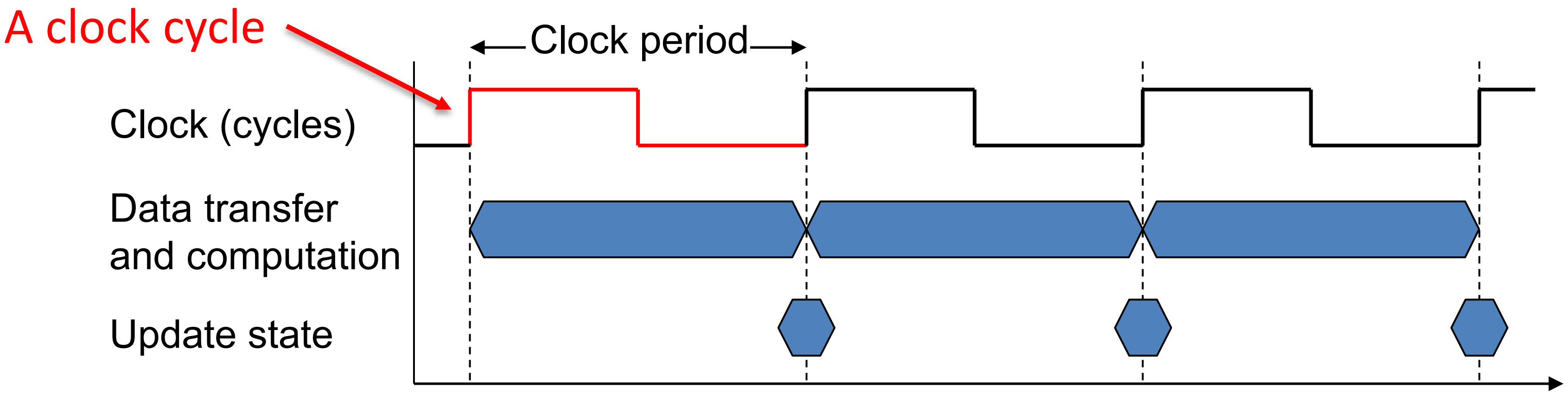
A is 1.5 times faster than B.

Measuring Execution Time

- **Elapsed time**
  - Total response time, including all aspects
    - Processing, I/O, OS overhead, idle time
  - Determines system performance
- **CPU time**
  - Time spent processing a given job on CPU
    - Discounts I/O time, other jobs' shares
  - Estimating CPU time is relatively easy based on the number of lines of code and processing speed of the CPU

CPU Clocking

- **Operation of digital hardware governed by a constant-rate clock**
- **Different system components may have different clock rates**



Clock period: duration of a clock cycle

e.g., 250ps = 0.25ns = 250×10^{-12}s

Clock frequency (rate): cycles per second

e.g., 4.0GHz = 4000MHz = 4.0×10^{9}Hz

CPU Time Estimation

- A code line of high-level language program can be translated to **one or multiple** assembly code lines

- CPU processes a piece of machine code for each assembly code line, one by one

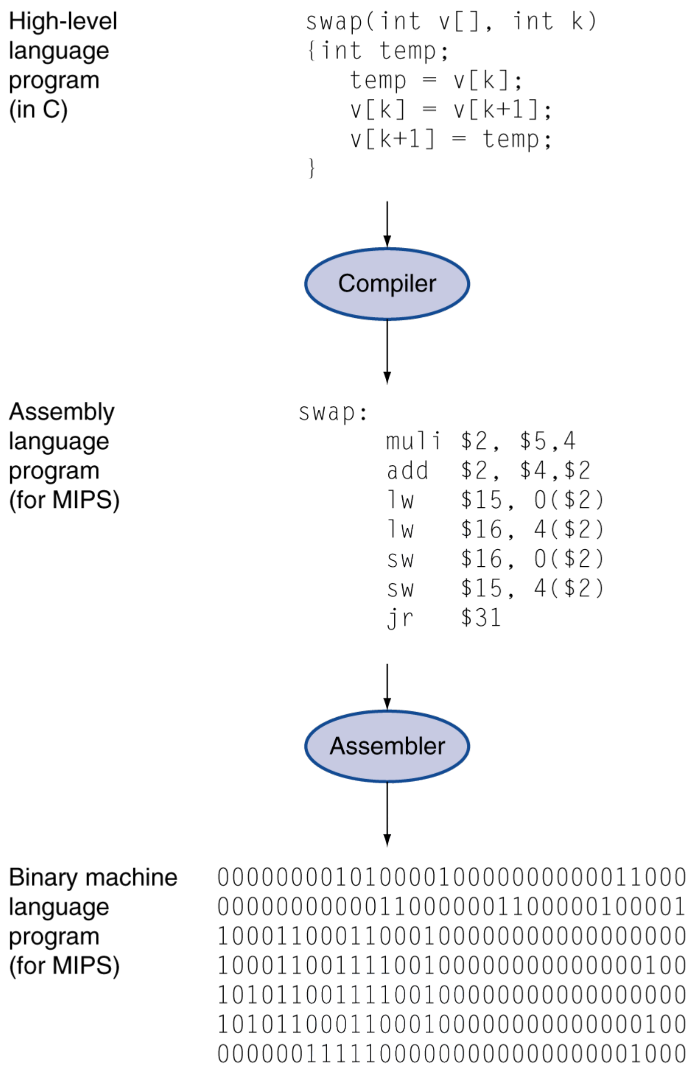
- **Example:** Application A has 1000 assembly lines. CPU has 250 ps clock period.

Clock cycles for Application A

= 1000 cycles

CPU time for Application A

= 1000 cycles x 250 ps = 250×10^{-9}s



CPU Time

CPU Time = Clock Cycles x Clock Period

= Clock Cycles / Clock Frequency

CPU Time Example

- **Computer A:** 2GHz clock, 10s CPU time to run an Application
- **Designing Computer B**
  - Aim for 6s CPU time for the same Application
  - Can do faster clock, but causes 1.2 × clock cycles
- **How fast must Computer B clock be?**

$$\begin{aligned}
\text{CPU Time}_B &= 6 \text{ seconds} \\
&= \text{Clock Cycles}_B \text{ x Clock Period}_B \\
&= 1.2 \text{ x Clock Cycles}_A \text{ x Clock Period}_B \\
&= 1.2 \text{ x } 20 \text{ x } 10^9 \text{ x Clock Period}_B
\end{aligned}$$

Therefore, $\text{Clock Period}_B = 6 \text{ second} / (24 \text{ x } 10^9)$

$\text{Clock Frequency}_B = 4 \text{ x } 10^9 = 4\text{GHz}$

$\text{CPU Time}_A = \text{Clock Cycles}_A / \text{Clock Frequency}_A$

$10 \text{ seconds} = \text{Clock Cycles}_A / 2\text{GHz}$

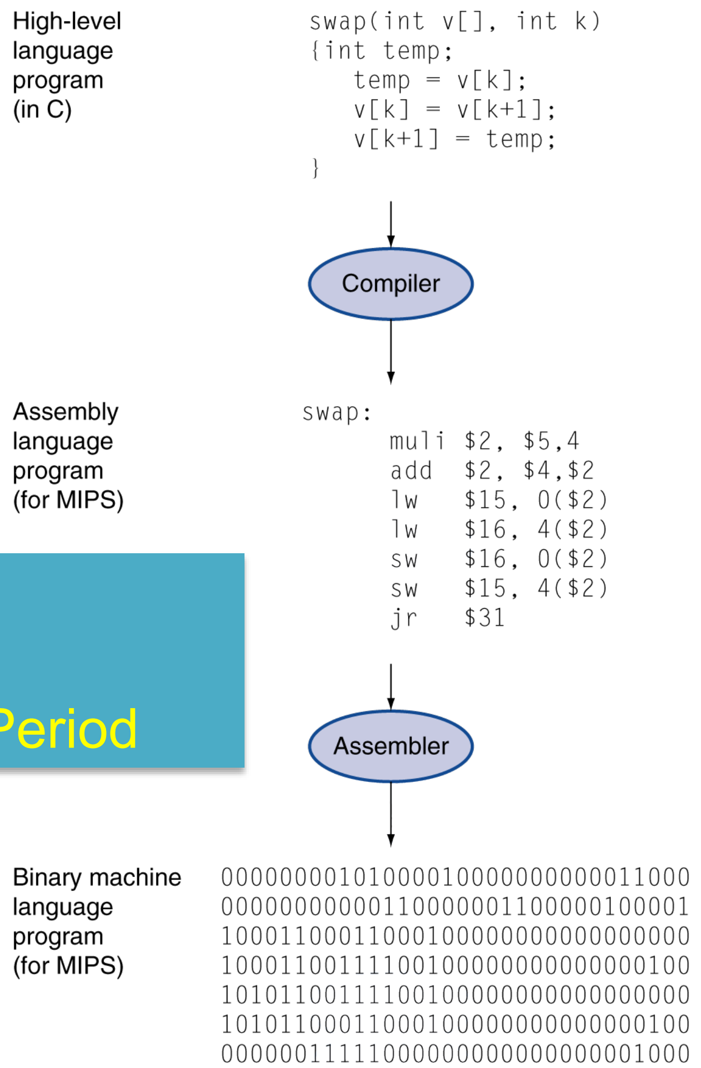
$\text{Clock Cycles}_A = 10 \text{ seconds x } 2\text{GHz} = 20 \text{ x } 10^9 \text{ cycles}$

Instruction Count and CPI

- An assembly line is also called one **Instruction**
- In some CPUs, an instruction may take **multiple clock cycles**
- **CPU Time equation can be rewritten**

CPU Time
= Clock Cycles x Clock Period
= Instruction Count x Cycles Per Instruction x Clock Period

- **Cycles Per Instruction** is also called **CPI**



CPI Example

- A program takes 33 billion instructions to run
- CPU processes instructions at 2 cycles per instruction
- Clock speed is 3GHz

What is estimated CPU Time for this program?

CPU Time = Instruction Count x CPI x Clock Period

$= 33 \times 10^9 \times 2 \times 1/(3 \times 10^9)$

= 22 seconds

CPI Example

- **Computer A:** Cycle Period = 250ps, CPI = 2.0
- **Computer B:** Cycle Period = 500ps, CPI = 1.2
- **Same number of instructions**

Which is faster, and by how much?

CPU Time_A = Instruction Count x CPI_A x Clock Period_A
= Instruction Count x 2.0 x 250 ps
= Instruction Count x 500 ps

CPU Time_B = Instruction Count x CPI_B x Clock Period_B
= Instruction Count x 1.2 x 500 ps
= Instruction Count x 600 ps

Computer A is faster than B by 600ps/500ps = 1.2 times

CPI and Average CPI

- **In some CPUs, different types of instructions** (i.e. integer instruction vs. floating point instruction) **may take different number of cycles**

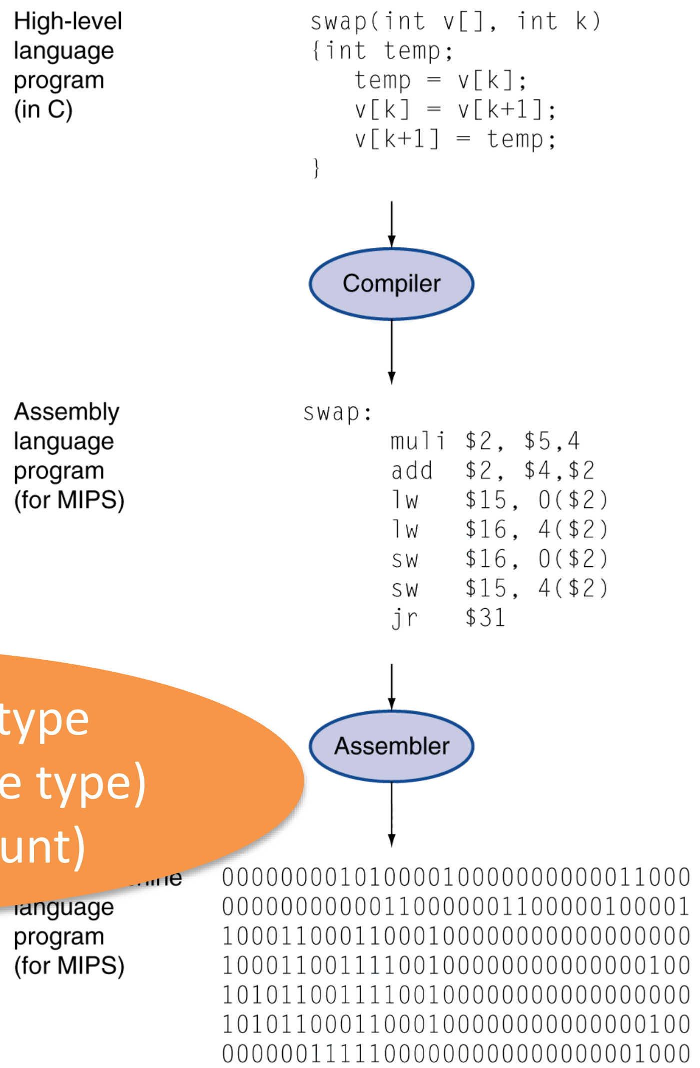
- **CPU Time equation can be rewritten**

$$\textbf{CPU Time} = \text{Clock Cycles} \times \text{Clock Period}$$
$$= \sum \left(IC_i \times CPI_i \right) \times \text{Clock Period}$$

Sum of (IC of each type instruction x CPI of the type)
(IC = instruction count)

- **Average CPI**

$$= \text{Total Clock Cycles} / \text{Total IC}$$
$$= \sum \left(IC_i \times CPI_i \right) / \text{Total IC}$$



Example: Average CPI

Instruction Type	Integer	Floating point	Branch	Load/ Store
CPI for type	1	2	4	3
Instruction Count in program A	50	10	10	10
Instruction Count in program B	20	0	10	10

- **Program A: Total Instructions = 80**

Clock Cycles = 50×1 + 10×2 + 10×4 + 10x3 = 140

Avg. CPI = 140/80 = 1.75

CPU time (if Clock period = 100 ps)
= 1.75 x 80 x 100ps = 14000 ps = 14ns

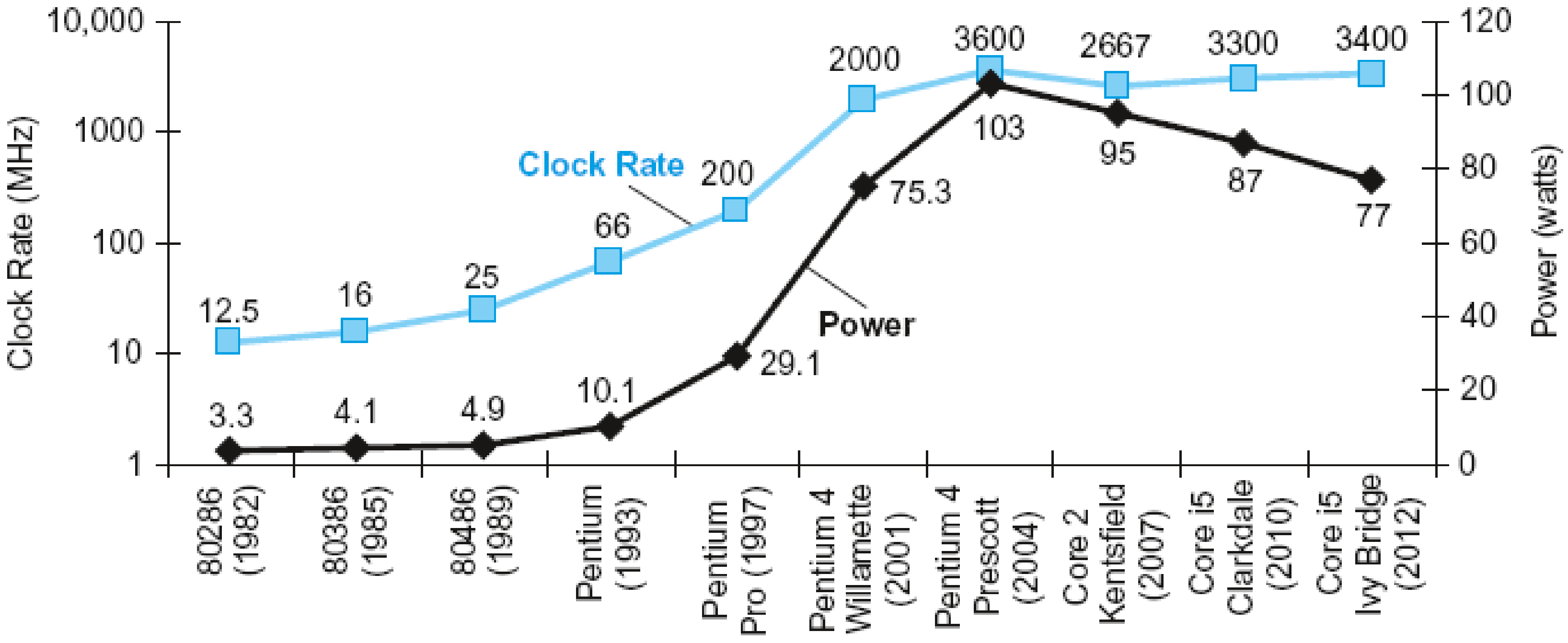
- **Program B: Total Instructions = 40**

Clock Cycles = 20×1 + 10×4 + 10×3 = 90

Avg. CPI = 90/40 = 2.25

CPU time (if Clock period = 100 ps)
= 2.25 x 40 x 100ps = 9000 ps = 9ns

Power and Performance



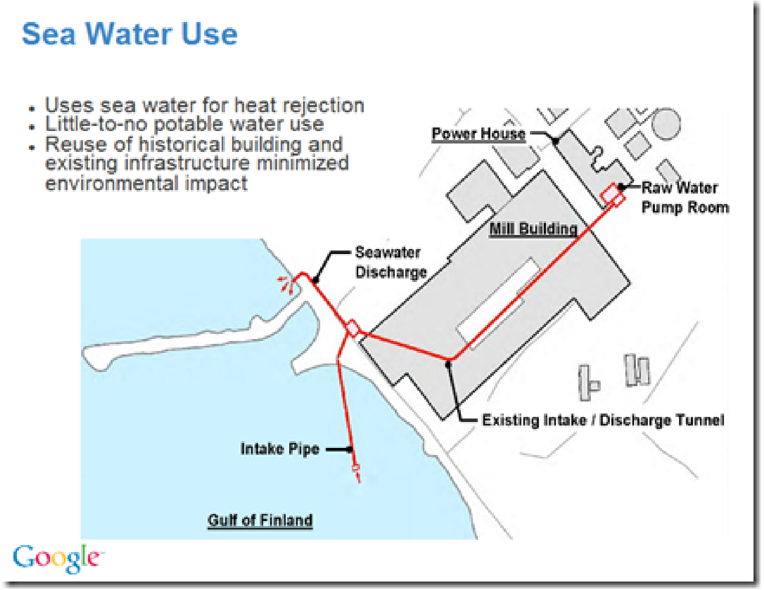
In CMOS IC technology

Power = Capacitive load x Voltage^2 x Frequency

Issue with Power Scaling

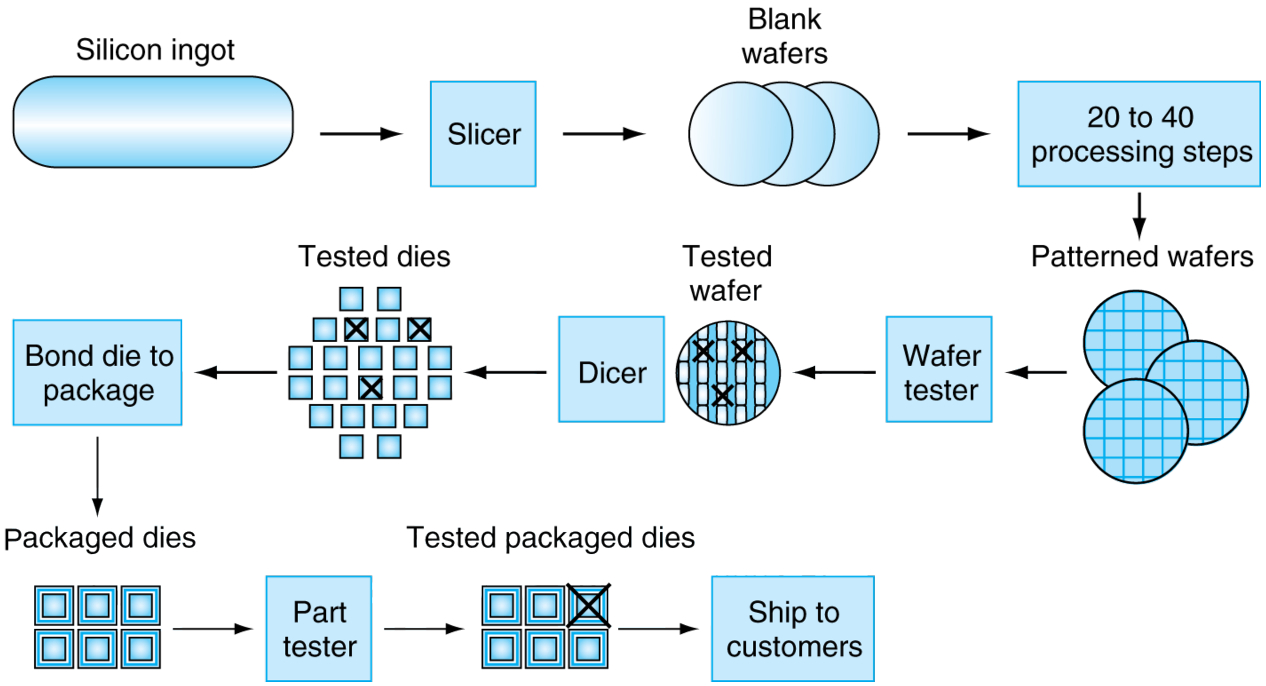
- **Increasing core frequency may burn your processor!**
  - Almost all energy consumption turns into heat

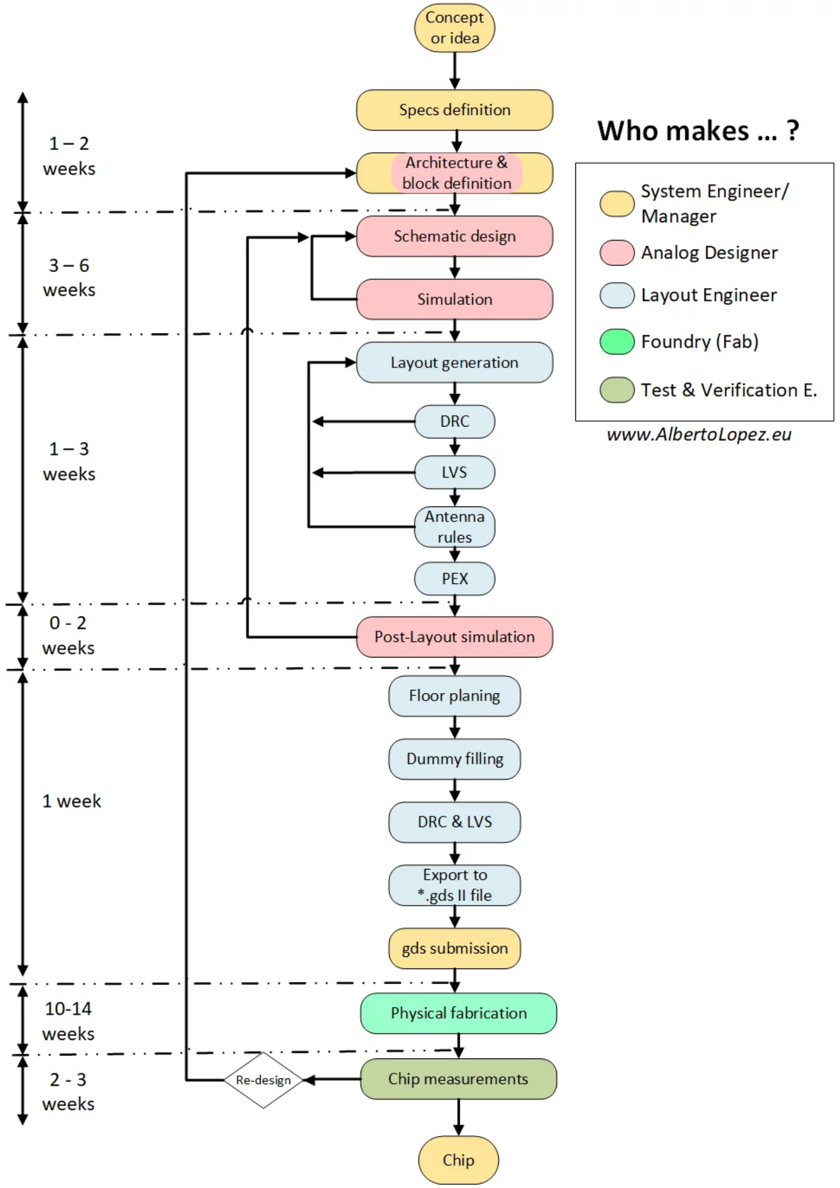
- **Cooling is costly**



- **Design more efficient system instead**
  - Multicore, etc..

Design and Manufacturing Process





Course Modules

- **ISA**
- **Processor**
- **Memory**
- **Multicore**

Conclusion Time

What is average CPI?

- $\sum \mathrm{IC}_i \times \mathrm{CPI}_i$ / Total IC

What is wrong with increasing power to the system?

- **Heat**
- **Cost**

SAN JOSÉ STATE UNIVERSITY *powering* SILICON VALLEY