Chapter - 1

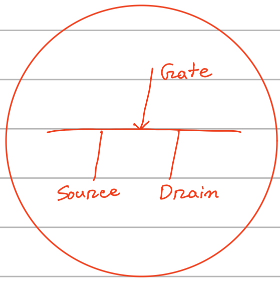
Moore's Law: The number of transistors on a microchip would double approximately every 18 - 24 months, leading to a corresponding increase in computer power.

** However, in recent years, it has become increasingly challenging to sustain the pace of doubling transistor counts on a chip at the same rate as before.

→ physical limitations.

→ Quantum effects.

→ Quantum Tunneling effect



⇒ transistor

Understanding Performance of a program...

Software Based

- Algorithm / original program
  - Determines number of operations executed
- Programming language, compiler, architecture — software used to translate the program into machine instructions.
  - Determine number of machine instructions executed per operation — Chapter 2,3

Hardware based

- Processor and memory system
  - Determine how fast instructions are executed
- I/O system (including OS)
  - Determines how fast I/O operations are executed

— Chapter 4,5,6

Eight Great Ideas

- Design for ***Moore's Law*** ✓

- Use ***abstraction*** to simplify design

  Hide the lower level details to keep things simple.

- Make the ***common case fast***

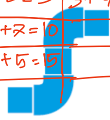
  you should know the common case first
  via experimentation

- Performance *via* ***parallelism*** → sum of (1 to 5)

	Single Core	Multi Core (2)		
5s	1+2=3	1+2=3	3+4=7	5s
5s	3+3=6	3+7=10		5s
5s	6+4=10	10+5=15		5s
5s	10+5=15			=15s
=20s				

- Performance *via* ***pipelining***

  will be explained in chap-4



- Performance *via* ***prediction***

- ***Hierarchy*** of memories

  will be explained in chap-5

- ***Dependability*** *via* redundancy

computers need to be both fast and reliable. Since any hardware can break, we make systems reliable by adding extra components that can take over if something breaks and to help spot any problem.

Why do we carry a spare tire in the trunk??
backup / redundancy

Imagine you're at a crosswalk, and the traffic light is about to change. Instead of waiting for the "Walk" signal to definitely light up, you start crossing when you see the cars slowing down and the opposite light turning red. You predict that it's safe to cross based on these cues. Most of the time, your prediction is accurate, and you save a bit of time by starting to cross earlier.

Statistically, the guesses are often correct and the cost of recovering from a wrong guess is relatively low.

Now, if you mispredict and the "Walk" signal doesn't light up, you quickly step back to the curb. The cost of stepping back is low (you just lose a few seconds), and the mechanism to recover (stepping back) is simple. Because your predictions are usually correct, you cross the street faster on average compared to waiting every time for the "Walk" signal to appear.

In computing, this concept is similar to speculative execution in CPUs. The processor guesses the likely path of a program and starts executing instructions ahead of time. If the guess is correct, the program runs faster. If the guess is wrong, the processor discards the work done on the wrong path and starts over from the correct point. As long as the prediction is usually right and the cost of recovering from a wrong guess is low, this approach speeds up the overall performance.

Datapath: Imagine your kitchen as a computer's datapath. The datapath is the part of the computer where data is processed, just like your kitchen is where you prepare food

1. Ingredients (Data):
   - The ingredients you use for cooking (vegetables, spices, etc.) represent the data the computer needs to process.
2. Chef (Processor):
   - The chef, is like the computer's processor. You take the ingredients and transform them into a meal. Similarly, the processor takes data and performs operations on it.
3. Kitchen Tools (Functional Units):
   - The tools you use (knives, pots, pans) are like the functional units in a datapath (ALU, multipliers, etc.). These tools help you process the ingredients. For example, a knife is used to chop vegetables, and a pan is used to cook them. In a datapath, the ALU might add or subtract numbers, while other units handle multiplication or data storage.
4. Recipe (Instructions):
   - The recipe you follow to make a dish is like the set of instructions the computer follows to process data. Each step in the recipe tells you what to do with the ingredients. In a computer, instructions tell the processor what operations to perform on the data.
5. Kitchen Workflow (Data Flow):
   - The way you move ingredients through your kitchen, from the fridge to the cutting board to the stove, represents the flow of data through the datapath. The datapath moves data from memory to the processor, through the functional units, and back to memory if needed.

ISA => Different ISAs define how a processor understands and executes instruction.

ABI => ABI defines how software interacts with the system's hardware and OS.