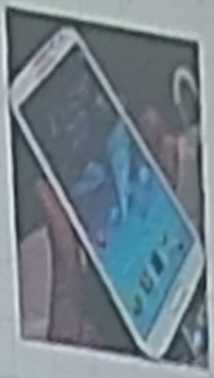
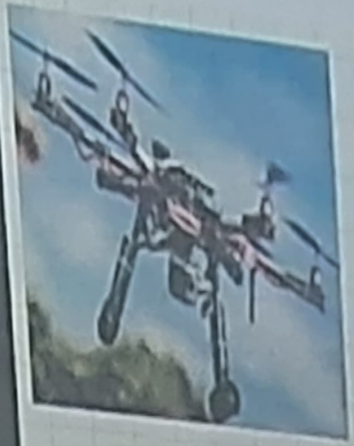
What is Computer Architecture?

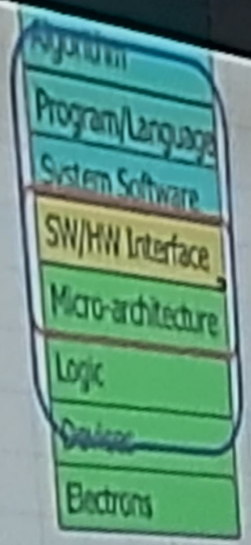
- The science and art of designing computing platforms
  - hardware, interface, system SW, and programming model
- Objective
  - Achieve a set of design goals
    - Performance
    - Performance/watt
    - etc





Computer Engineering Hierarchy

Computer Architecture
(expanded view)



Computer Architecture
(narrow view)

The Computer Engineering Hierarchy

Plenty of Room

- Top: Leiserson et. al.
  - "There's plenty of room at the Top: What will drive computer performance after Moore's law?", Science, 2020
- Bottom: Feynmann
  - "There's Plenty of Room at the Bottom: An Invitation to Enter a New Field of Physics", a lecture given at Caltech, 1959

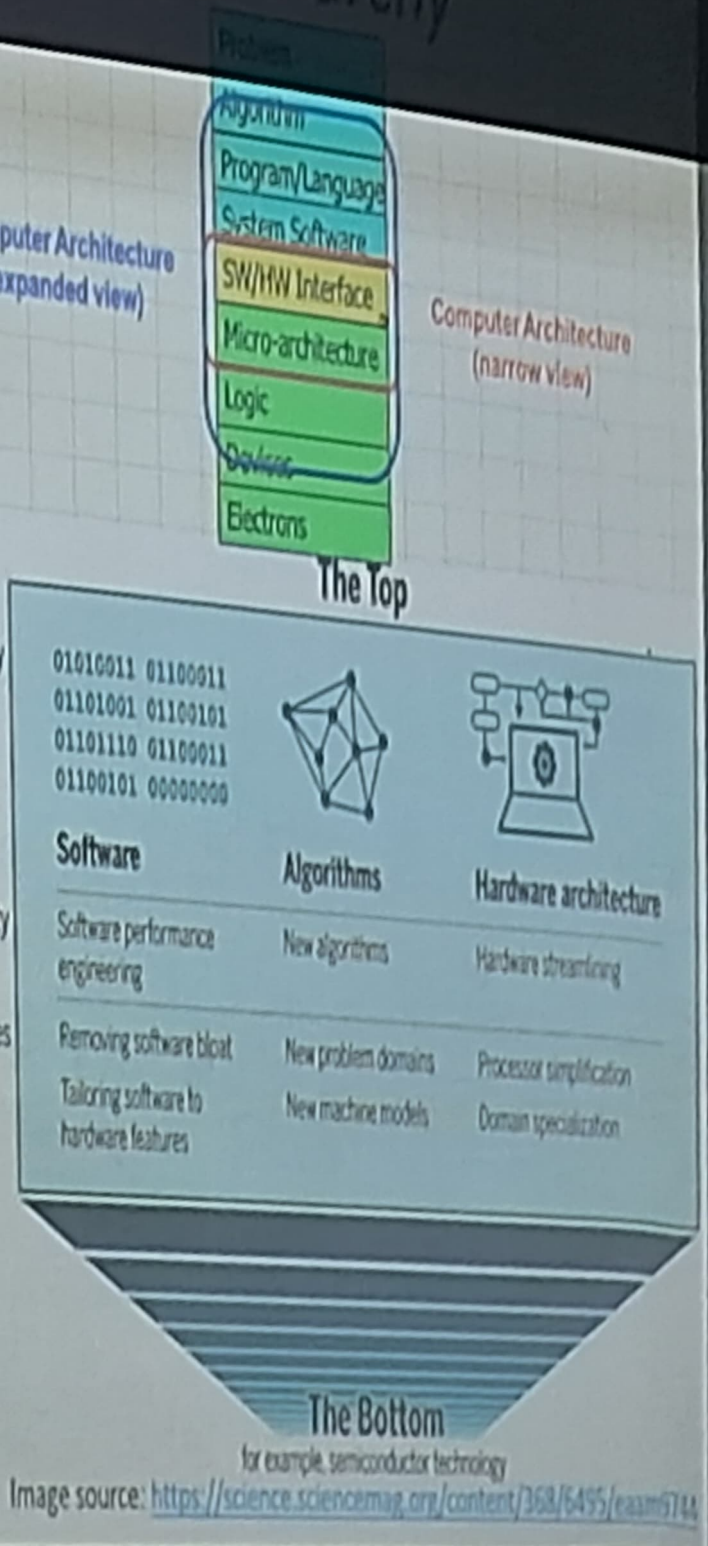


Image source: https://science.sciencemag.org/content/368/6495/eaam9744

Motivating Example

- Matrix Multiplication

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 58 & \\ & \end{bmatrix}$$

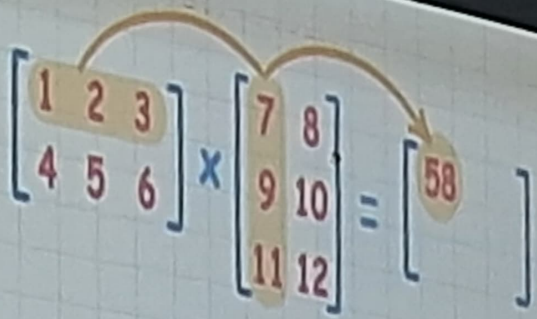
$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \begin{bmatrix} a_1 & b_1 \\ c & d_1 \end{bmatrix}$$

$$a\begin{bmatrix} \\ \end{bmatrix} \quad b\begin{bmatrix} \\ \end{bmatrix}$$

$$c\begin{bmatrix} \\ \end{bmatrix} \quad d\begin{bmatrix} \\ \end{bmatrix}$$

Matrix Multiplication

```
for i in xrange(4096):
  for j in xrange(4096):
    for k in xrange(4096):
      C[i][j] += A[i][k] *
B[k][j]
```

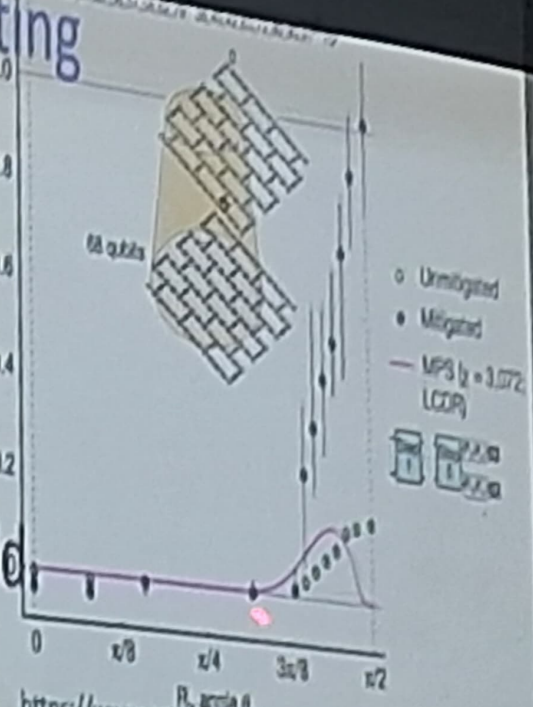


Implementation	Running time (s)	Absolute speedup
Python	25,552.48	1x
Java	2,372.68	11x
C	542.67	47x
Parallel loops	69.80	366x
Parallel divide and conquer	3.80	6,727x
plus vectorization	1.10	23,224x
plus AVX intrinsics	0.41	62,806x

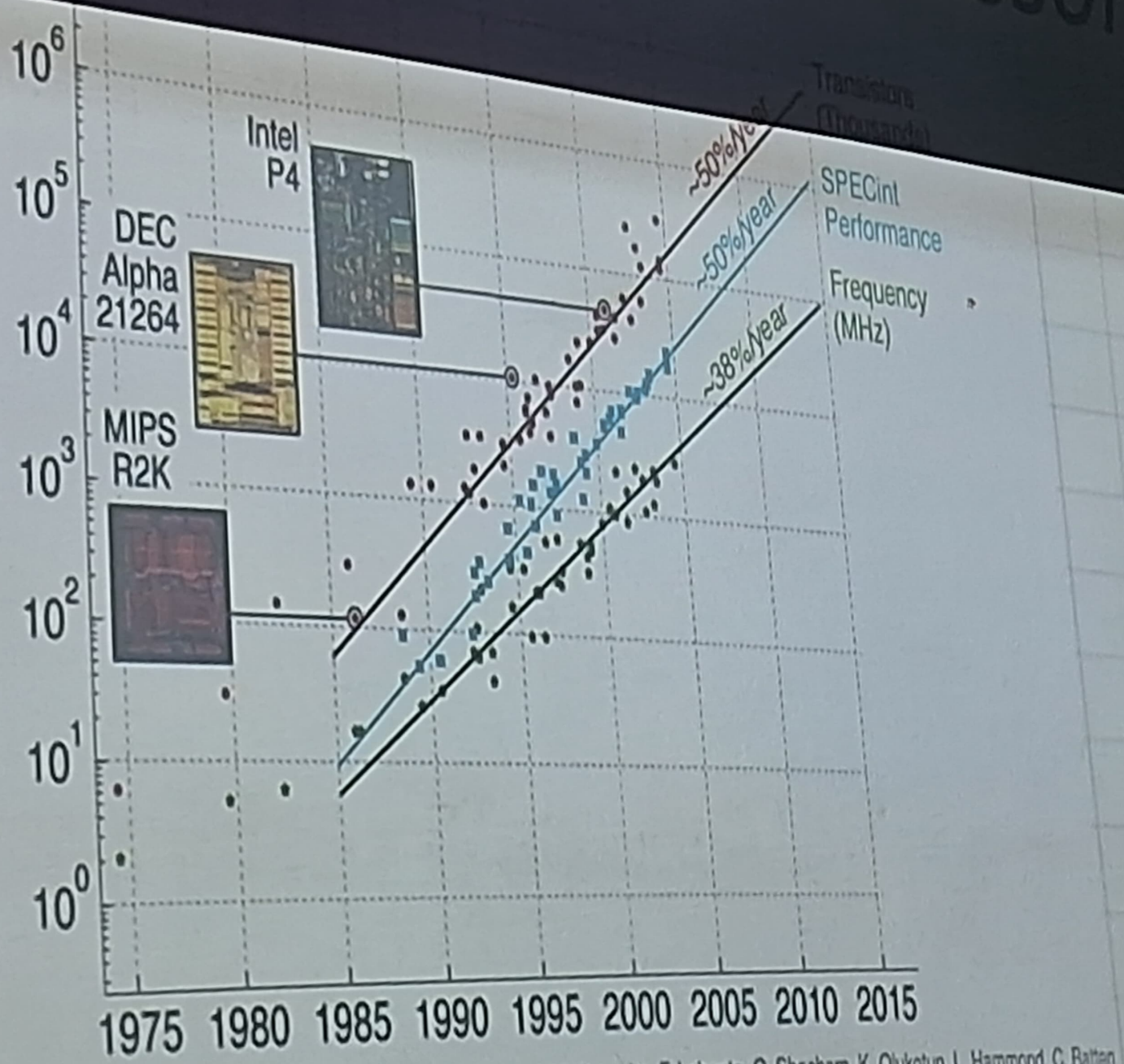
Leiserson+, "There's plenty of room at the Top: What will drive computer performance after Moore's law?", Science, 2020

Motivating Example

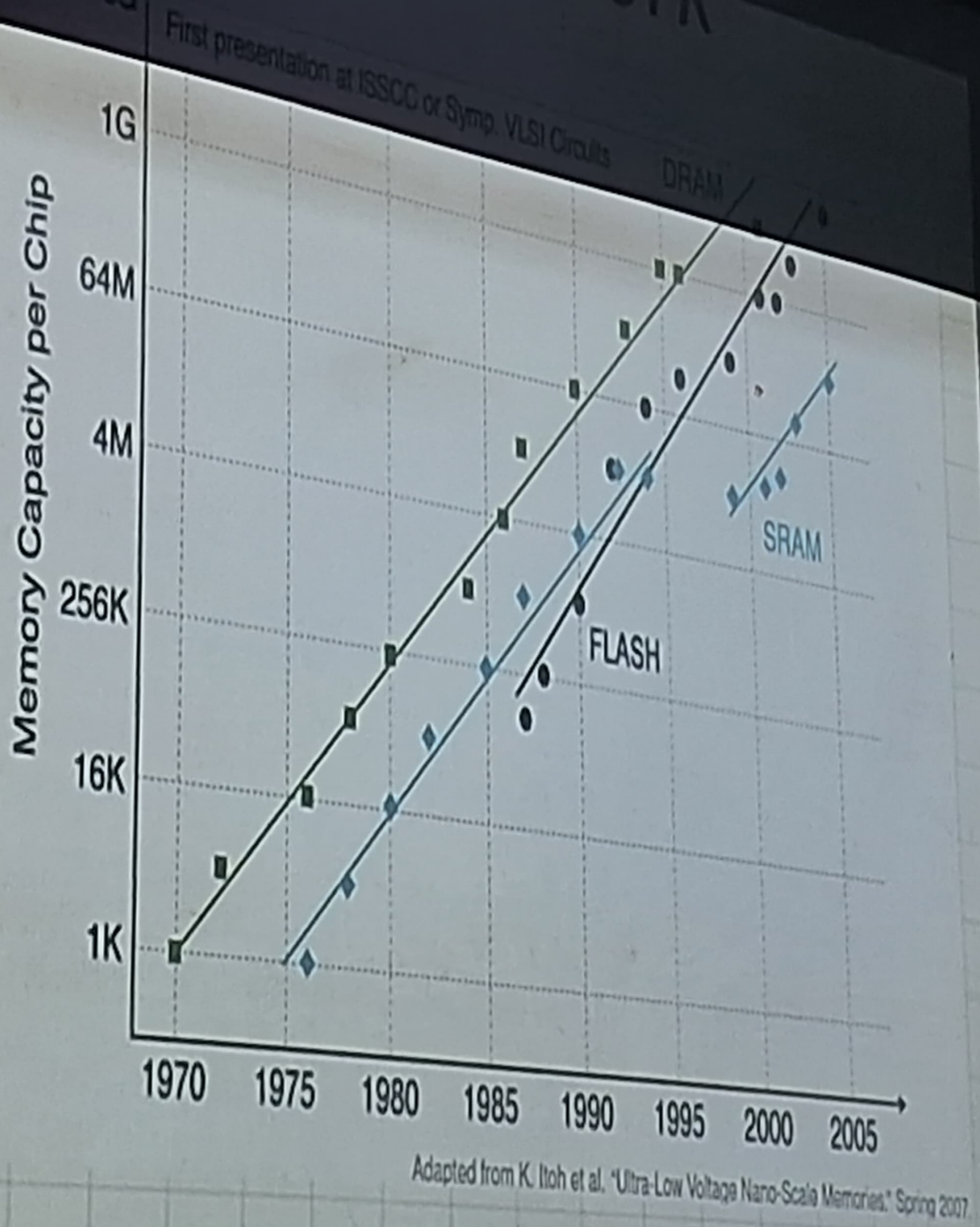
- Fault Tolerance in Quantum Computing
- Utility before FT
  - Evidence for the utility of quantum computing before fault tolerance
  - Time evolution of a 2D transverse-field Ising mode
    - 127 qubit system
    - Demonstrated that a quantum computer can outperform a classical computer.



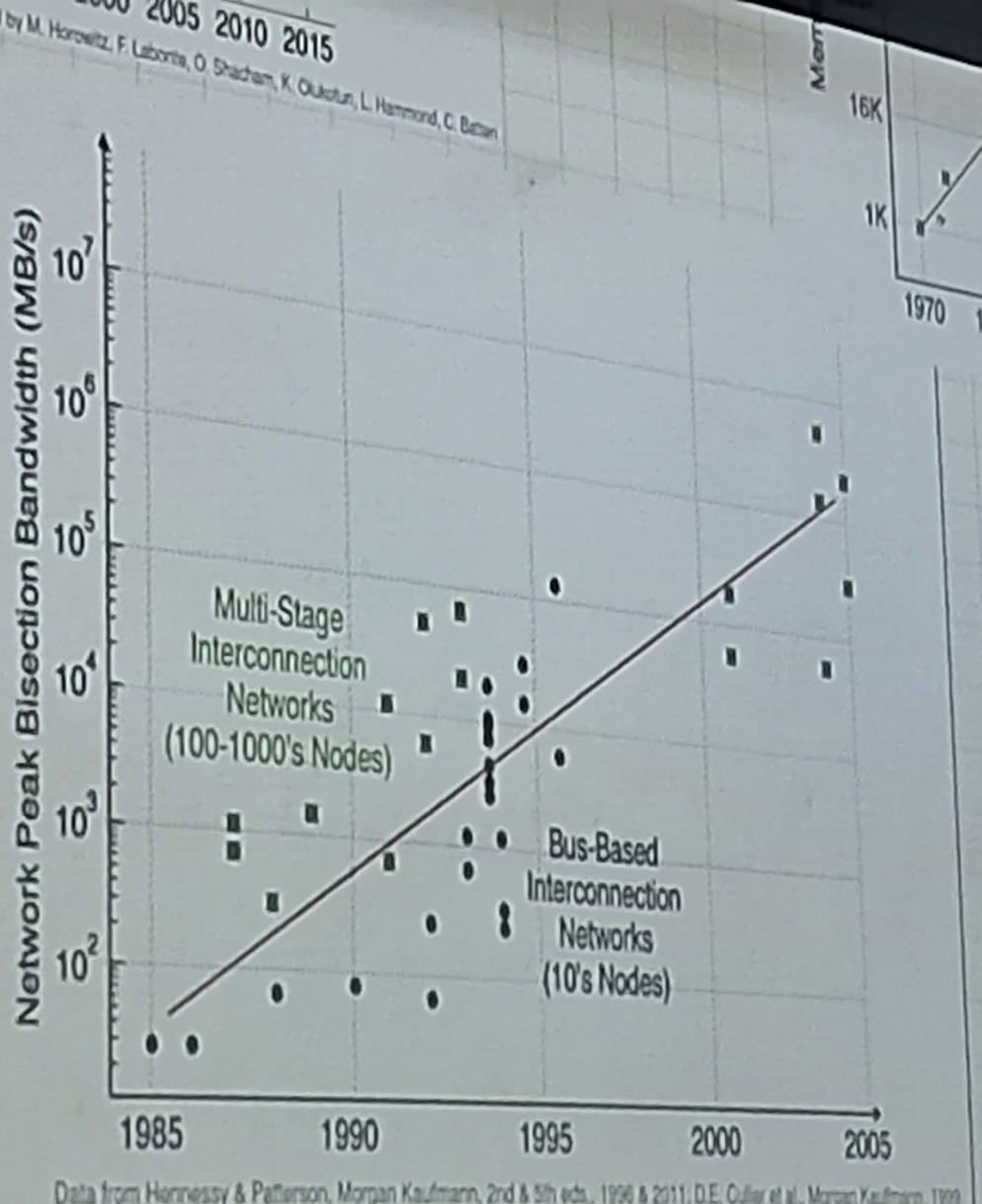
https://www.nature.com/articles/s41586-023-06096-3



Data collected by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, C. Batten



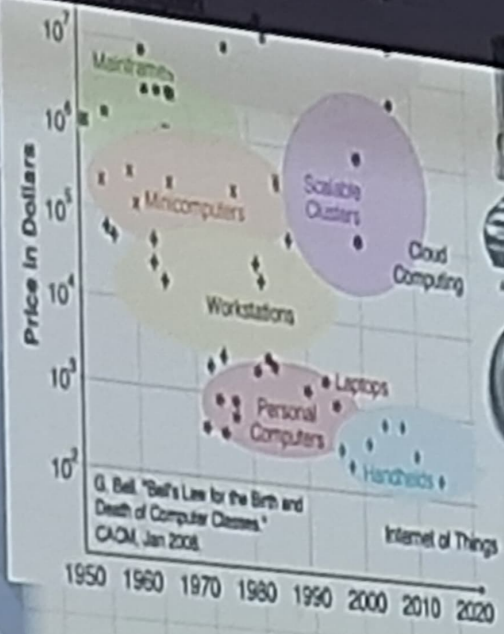
Adapted from K. Itoh et al. "Ultra-Low Voltage Nano-Scale Memories," Spring 2007

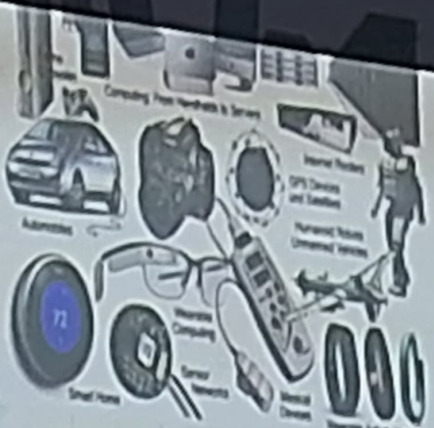

2005 2010 2015
Data collected by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, C. Batten
Network Peak Bisection Bandwidth (MB/s)
10^7
10^6
10^5
10^4
10^3
10^2
Multi-Stage Interconnection Networks (100-1000's Nodes)
Bus-Based Interconnection Networks (10's Nodes)
1985
1990
1995
2000
2005
Data from Hennessy & Patterson, Morgan Kaufmann, 2nd & 5th eds., 1996 & 2011; D.E. Culler et al., Morgan Kaufmann, 1999
16K
1K
1970
1975
1980
1985

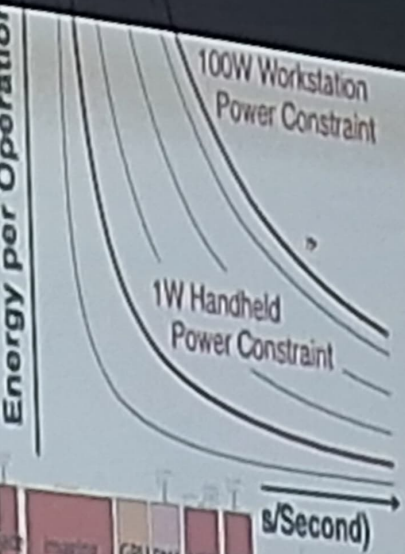
What drives CA today

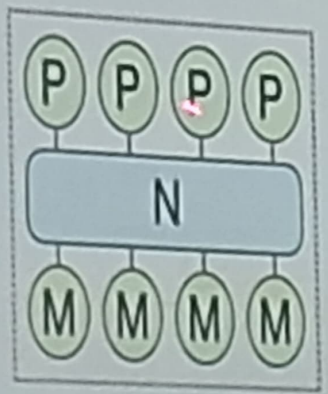
- Diversity in applications
  - Cloud and IoT
- Energy & power constrained systems
- Multiple cores
- Heterogeneous systems-on-chip
- Technology scaling challenges
  - New emerging compute, storage, and communication device technologies

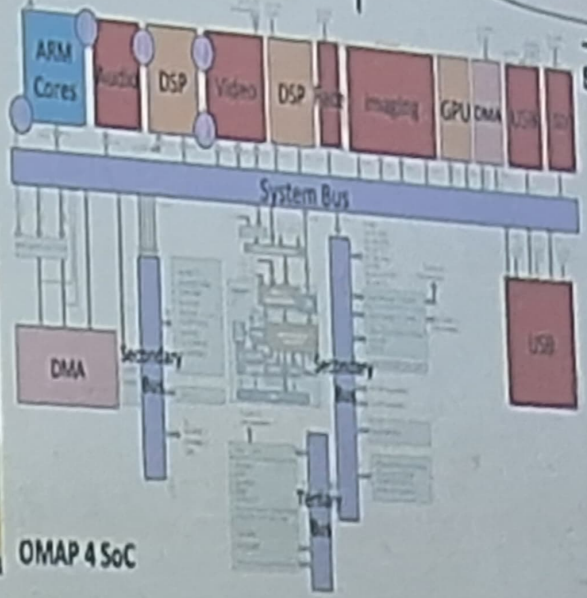
Some Trends



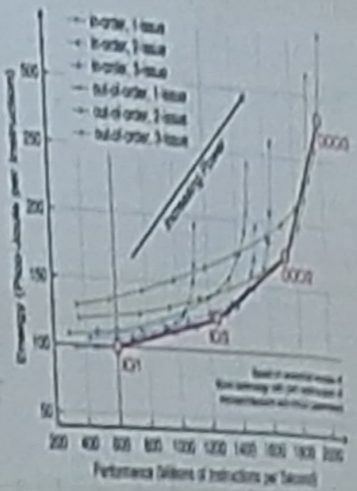




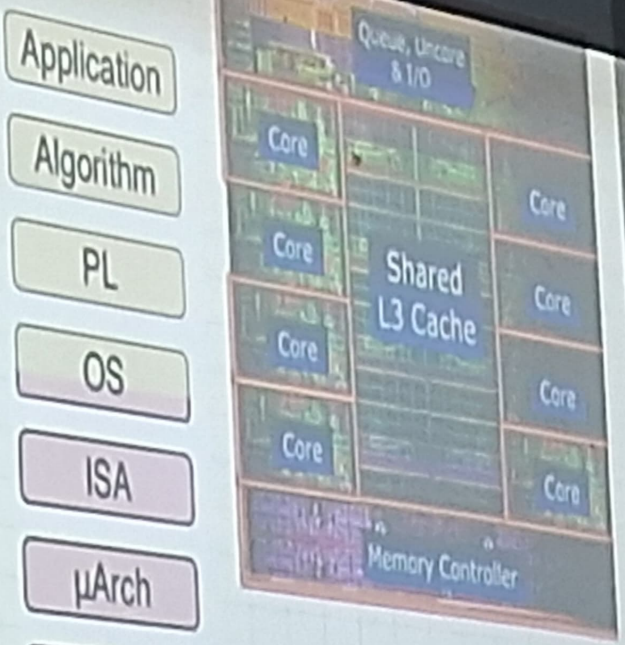




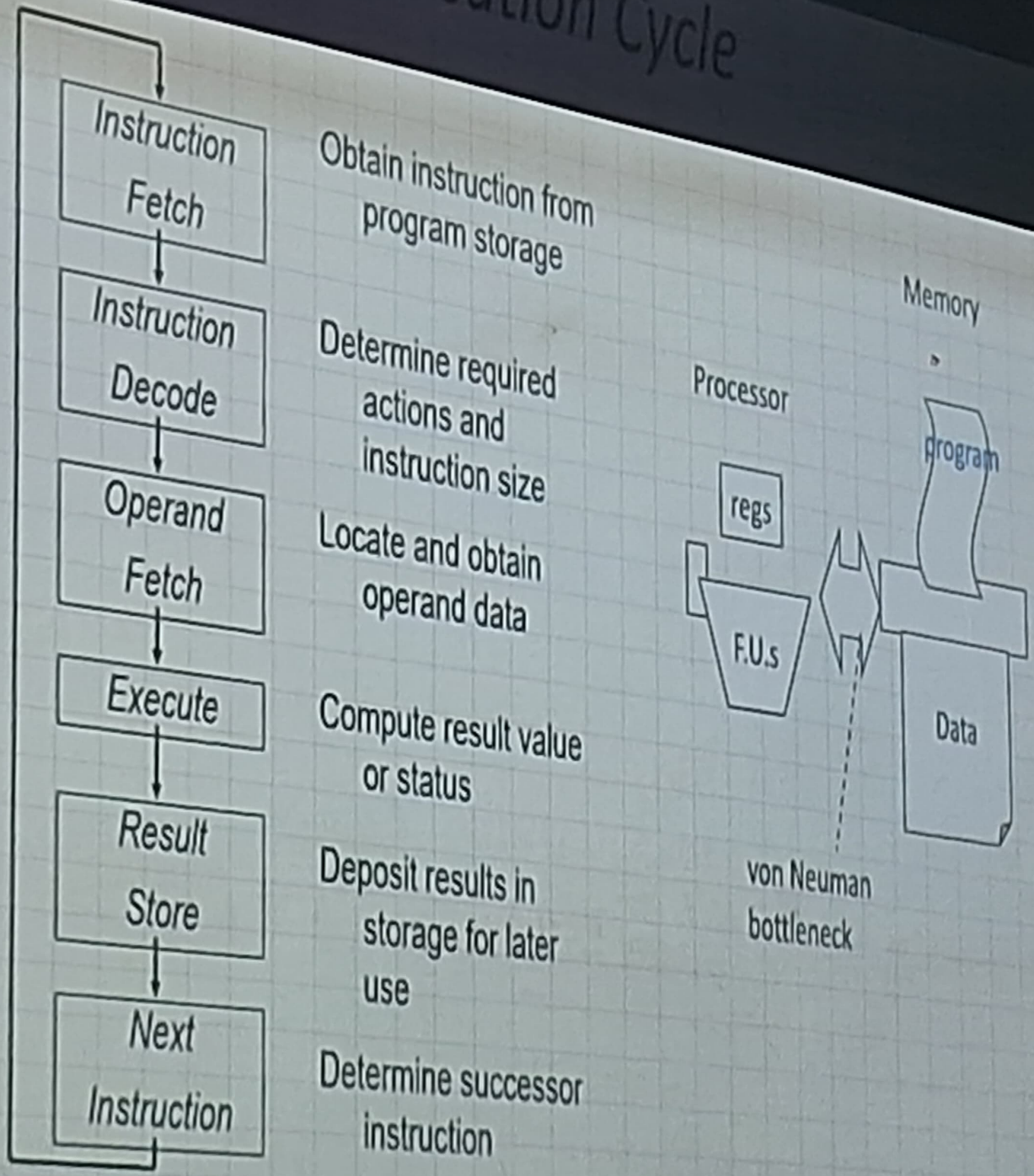
OMAP 4 SoC



- The Design of a Modern Processor
  - Fighter Plane
    - 10 k Parts
  - Intel Sandy Bridge E
    - 2.27 Billion Transistors
- Design Philosophy
  - Modularity
  - Hierarchy
  - Encapsulation
  - Regularity
  - Extensibility
- Design Patterns
  - Control/Datapath split

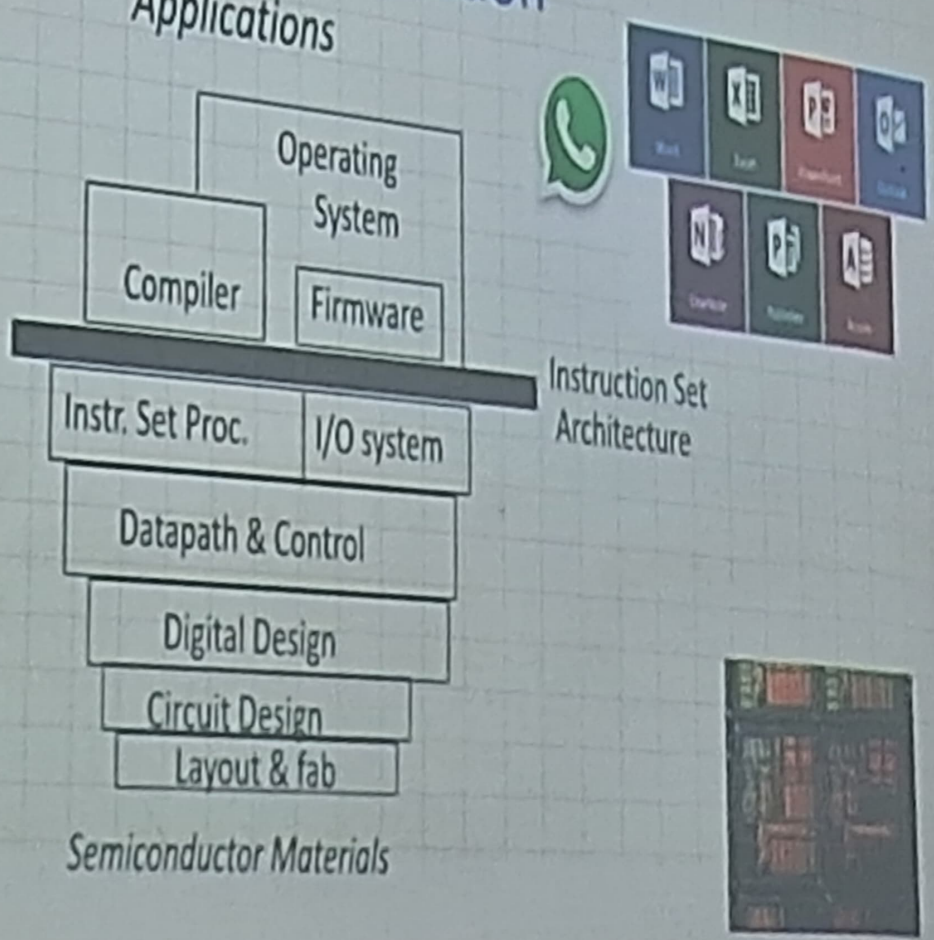


...damental Execution Cycle



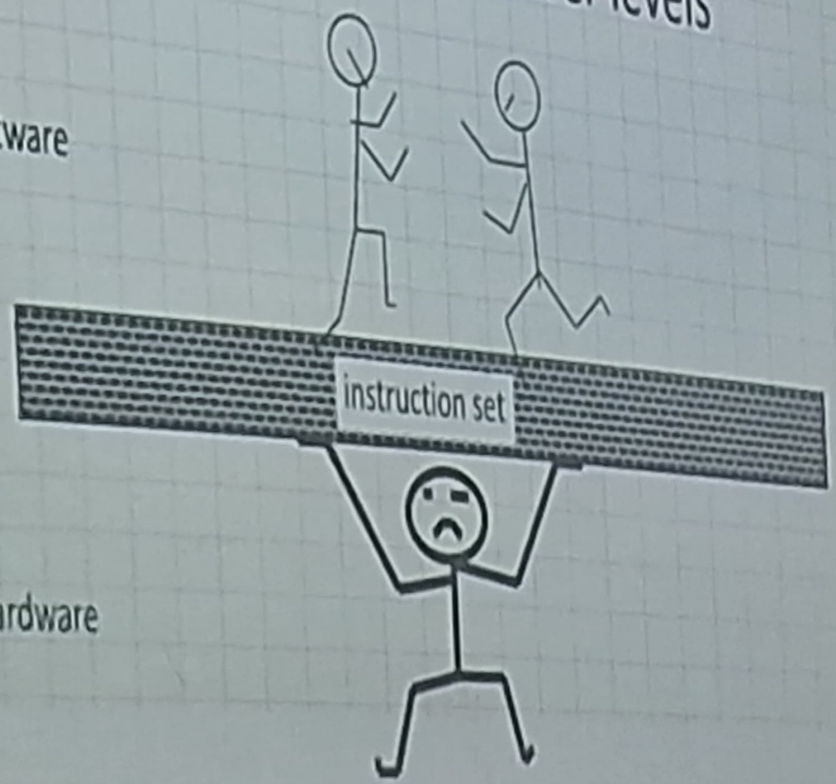
Computer Architecture"?

- Coordination of many *levels of abstraction*
- Under a rapidly *changing* set of forces
- Design, Measurement, *and* Evaluation

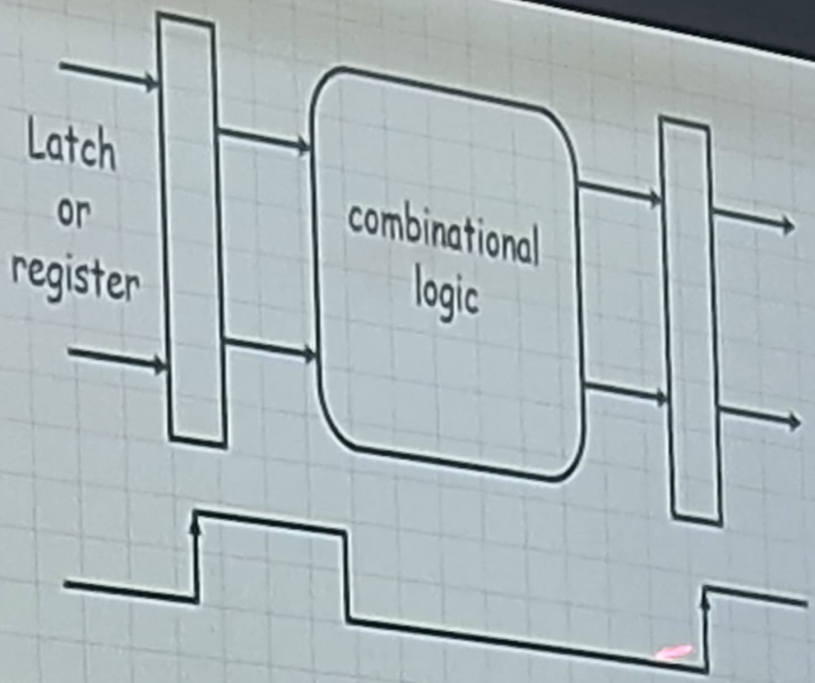


Instruction Set: a Critical Interface

- Properties of a good abstraction
  - Lasts through many generations (portability)
  - Used in many different ways (generality)
  - Provides convenient functionality to higher levels
  - Permits an efficient implementation at lower levels

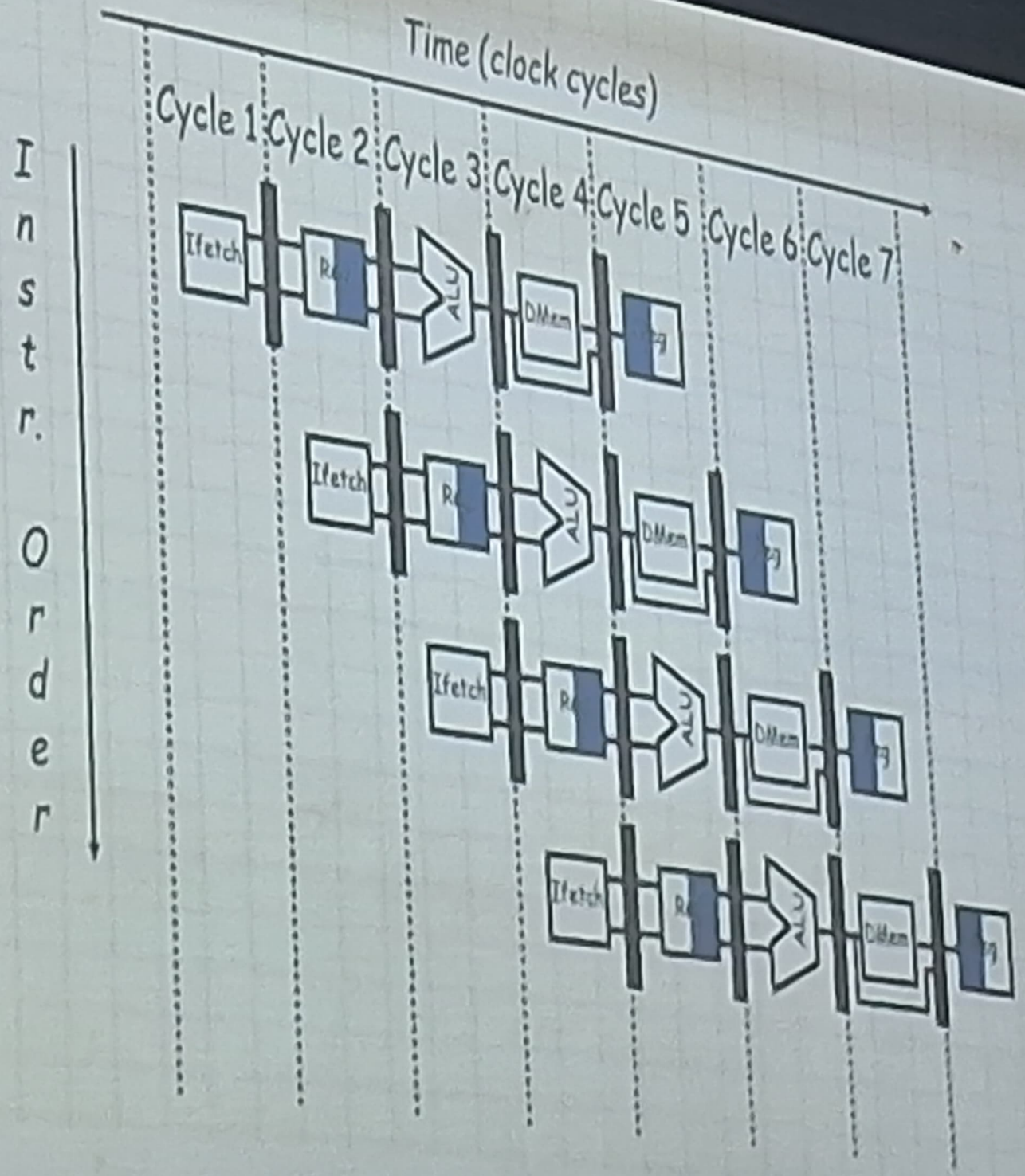


What's a Clock Cycle?



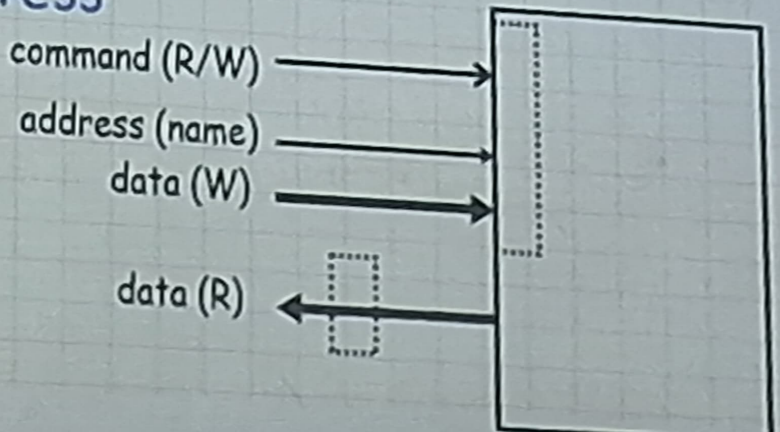
- 10 levels of gates

Pipelined Instruction Execution



The Memory Abstraction

- Association of <name, value> pairs
  - typically named as byte addresses
  - often values aligned on multiples of size
- Sequence of Reads and Writes
- Write binds a value to an address
- Read of addr returns most recently written value bound to that address



Levels of the Memory Hierarchy

Capacity
Access Time
Cost

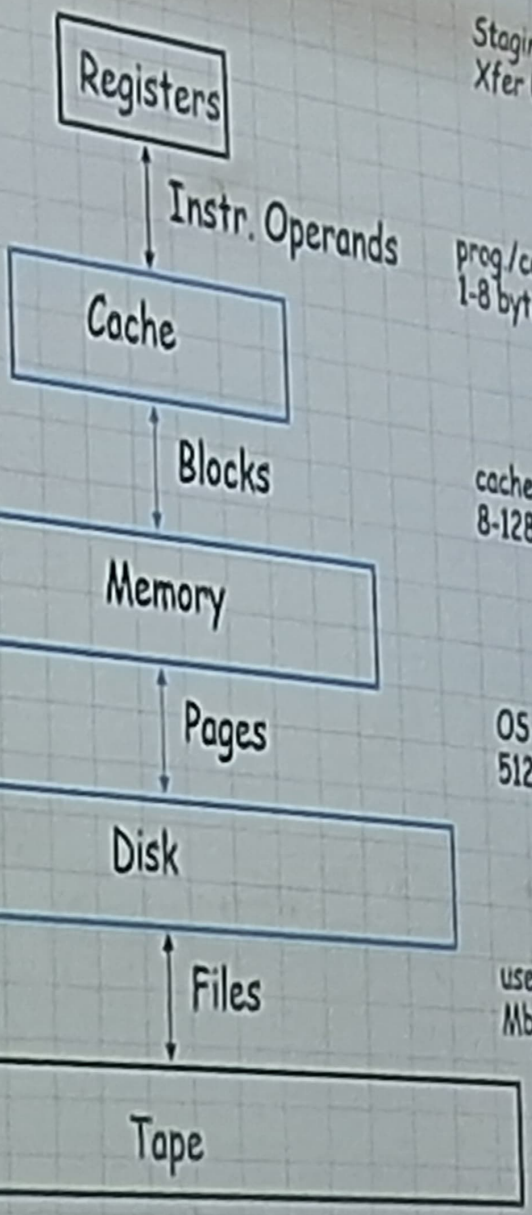
CPU Registers
100s Bytes
<< 1s ns

Cache
10s-100s K Bytes
~1 ns
$1s/ MByte

Main Memory
M Bytes
100ns- 300ns
$< 1/ MByte

Disk
10s G Bytes, 10 ms
(10,000,000 ns)
$0.001/ MByte

Tape
infinite
sec-min
$0.0014/ MByte



Staging
Xfer Unit

prog./compiler
1-8 bytes

cache cntl
8-128 bytes

OS
512-4K bytes

user/operator
Mbytes

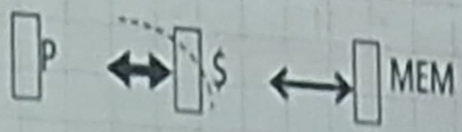
faster

Larger

Lower Level

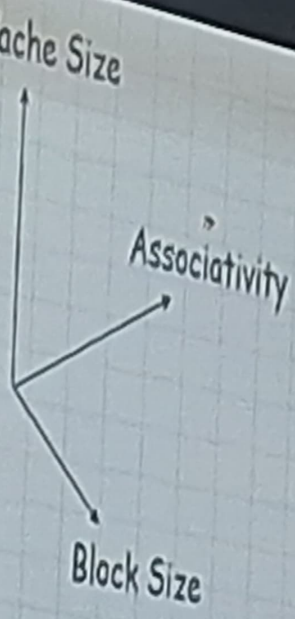
The Principle of Locality

- The Principle of Locality:
  - Program access a relatively small portion of the address space at any instant of time.
- Two Different Types of Locality:
  - Temporal Locality (Locality in Time): If an item is referenced, it will tend to be referenced again soon (e.g., loops, reuse)
  - Spatial Locality (Locality in Space): If an item is referenced, items whose addresses are close by tend to be referenced soon (e.g., straightline code, array access)
- Last 30 years, HW relied on locality for speed

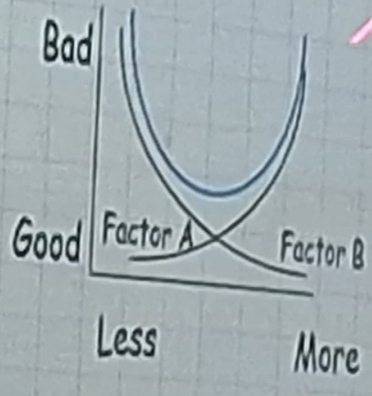


The Cache Design Space

- Several interacting dimensions
  - cache size
  - block size
  - associativity
  - replacement policy
  - write-through vs write-back

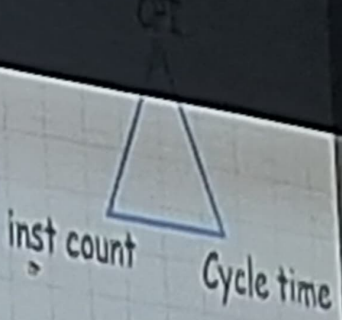


- The optimal choice is a compromise
  - depends on access characteristics
    - workload
    - use (I-cache, D-cache, TLB)
  - depends on technology / cost
- Simplicity often wins



Processor performance equation

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



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