CS 110
Computer Architecture

Summary

Instructors:

Chundong Wang & Siting Liu

https://toast-lab.sist.shanghaitech.edu.cn/courses/CS110@ShanghaiTech/Spring-2023/index.html

School of Information Science and Technology SIST

ShanghaiTech University

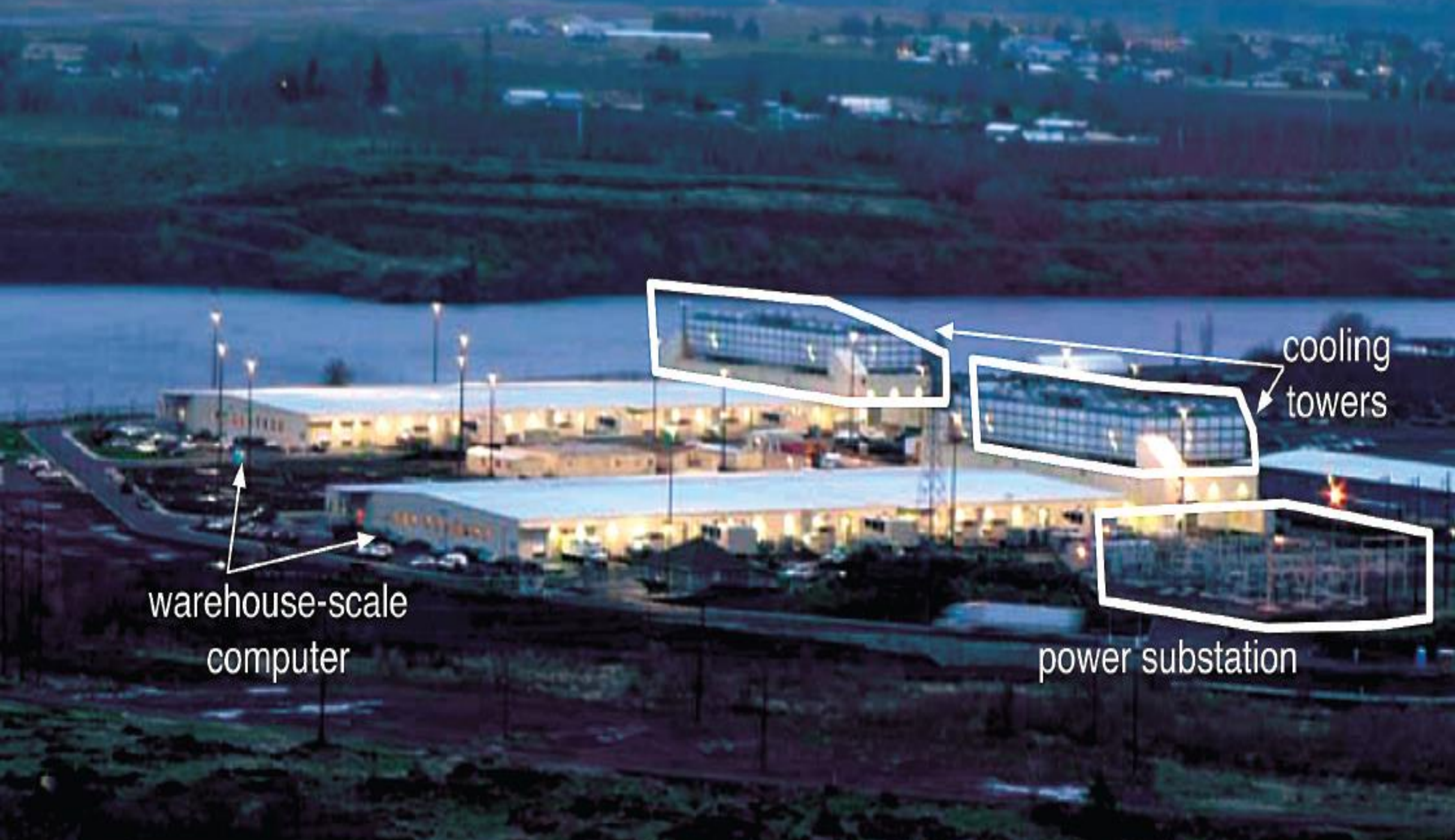
Slides based on UC Berkley's CS61C

Let us review CA now.

New School Computer Architecture (1/3)

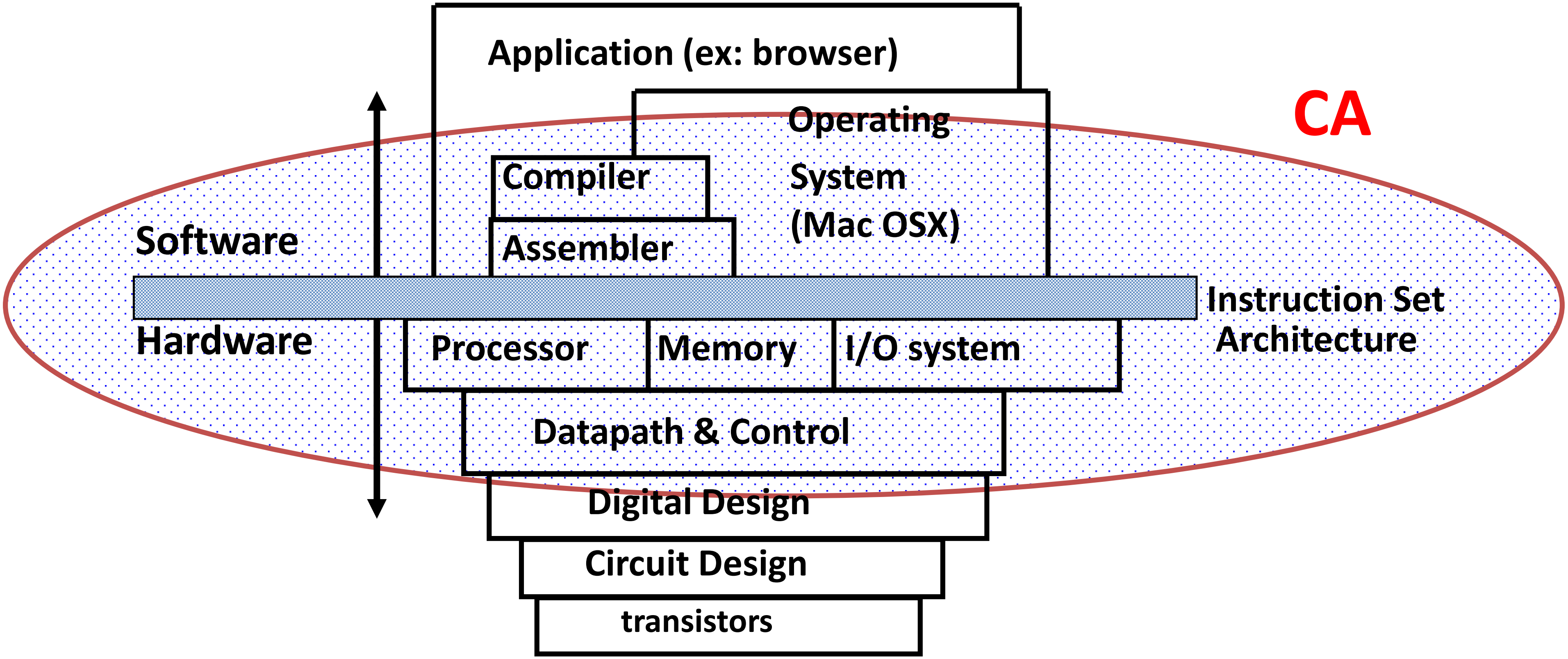
Personal
Mobile
Devices

New School Computer Architecture (2/3)



New School Computer Architecture (3/3)

Old Machine Structures



New-School Machine Structures (It’s a bit more complicated!)

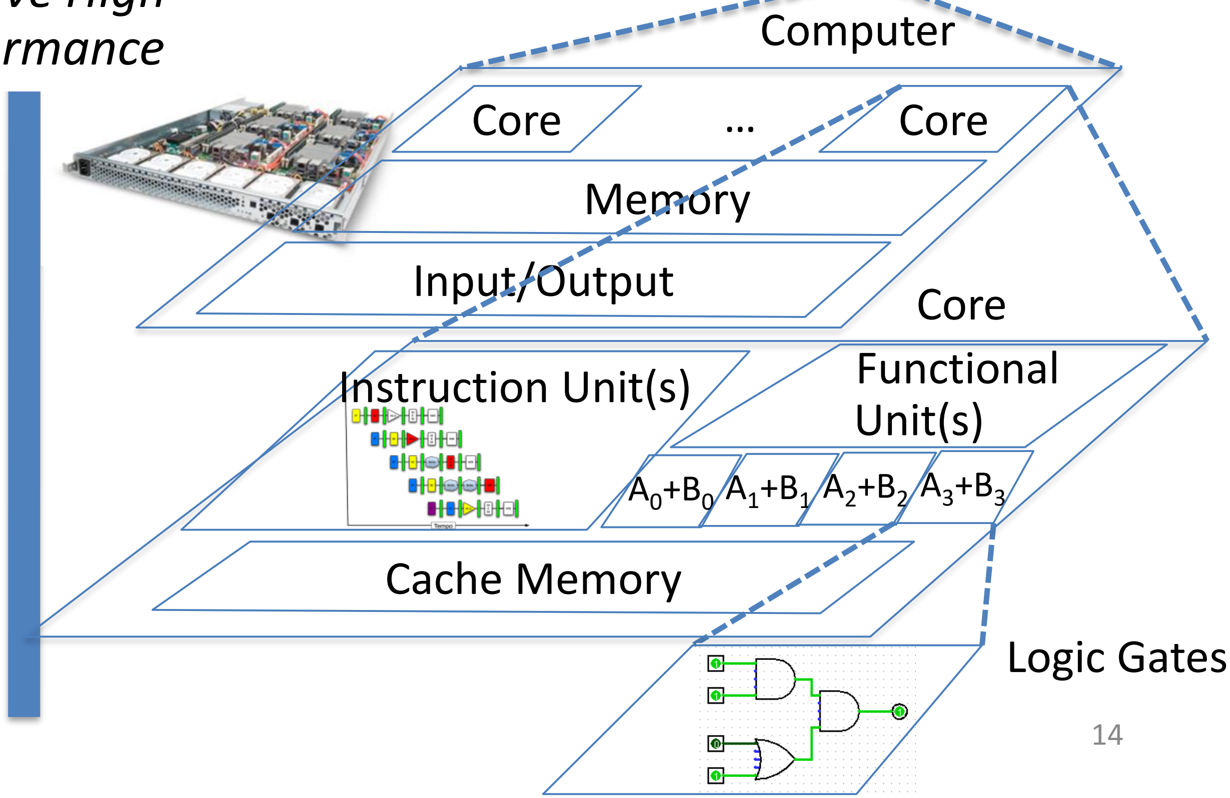
Software | *Hardware*

- Parallel Requests
  Assigned to computer
  e.g., Search “Avatar 2”
- Parallel Threads
  Assigned to core
  e.g., Lookup, Ads
- Parallel Instructions
  >1 instruction @ one time
  e.g., 5 pipelined instructions
- Parallel Data
  >1 data item @ one time
  e.g., Add of 4 pairs of words
- Hardware descriptions
  All gates functioning in parallel at same time
- Programming Languages

Leverage Parallelism & Achieve High Performance

Warehouse Scale Computer

Smart Phone



Great Ideas in Computer Architecture

1. Design for Moore's Law
2. Abstraction to Simplify Design eg. network
3. Make the Common Case Fast
4. Dependability via Redundancy
5. Memory Hierarchy
6. Performance via Parallelism/Pipelining/Prediction

Powers of Ten inspired CA Overview

Going Top-Down cover 3 Views

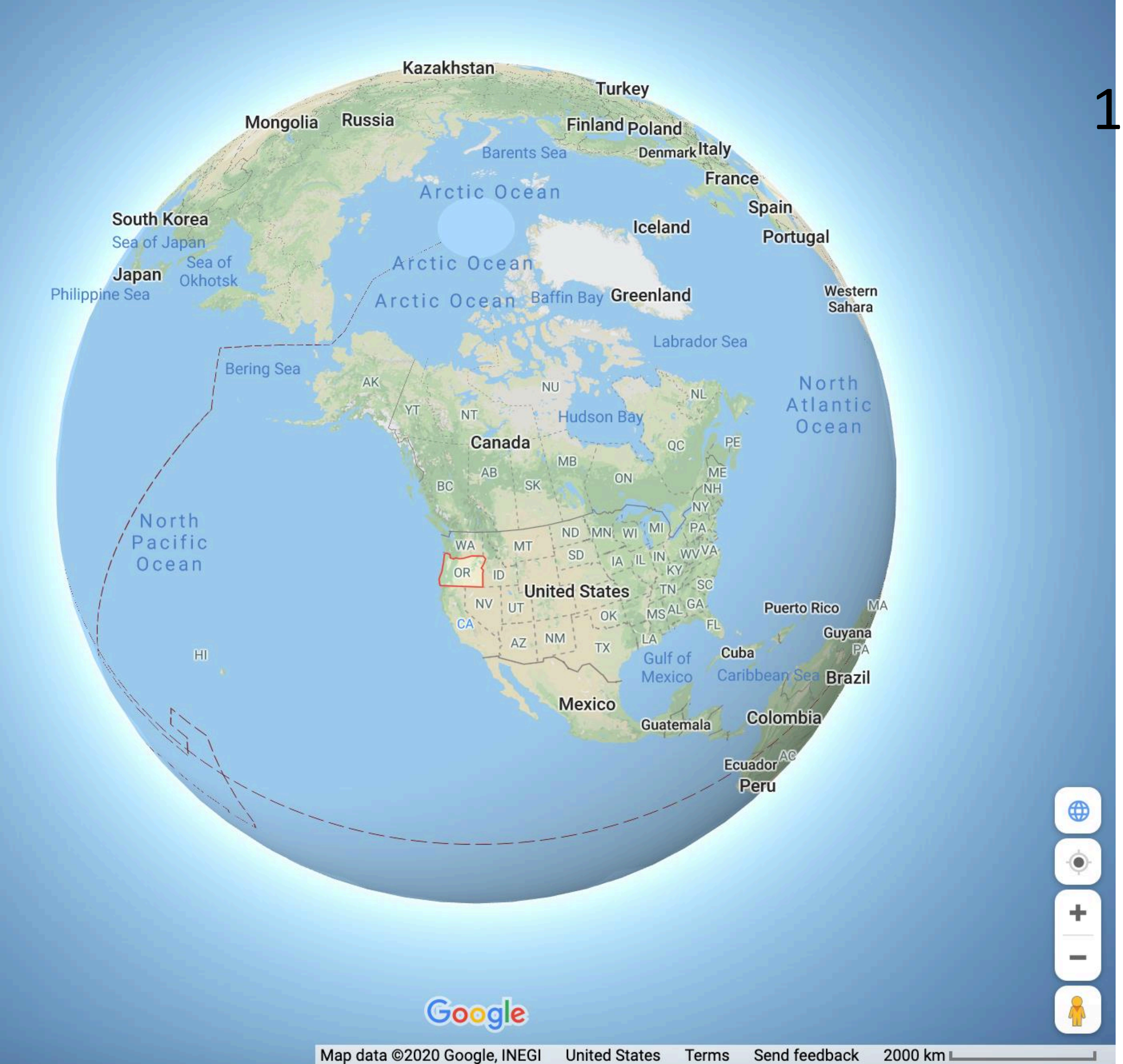
1. Architecture (when possible)
2. Physical Implementation of that architecture
3. Programming system for that architecture and implementation (when possible)

- See http://www.powersof10.com/film

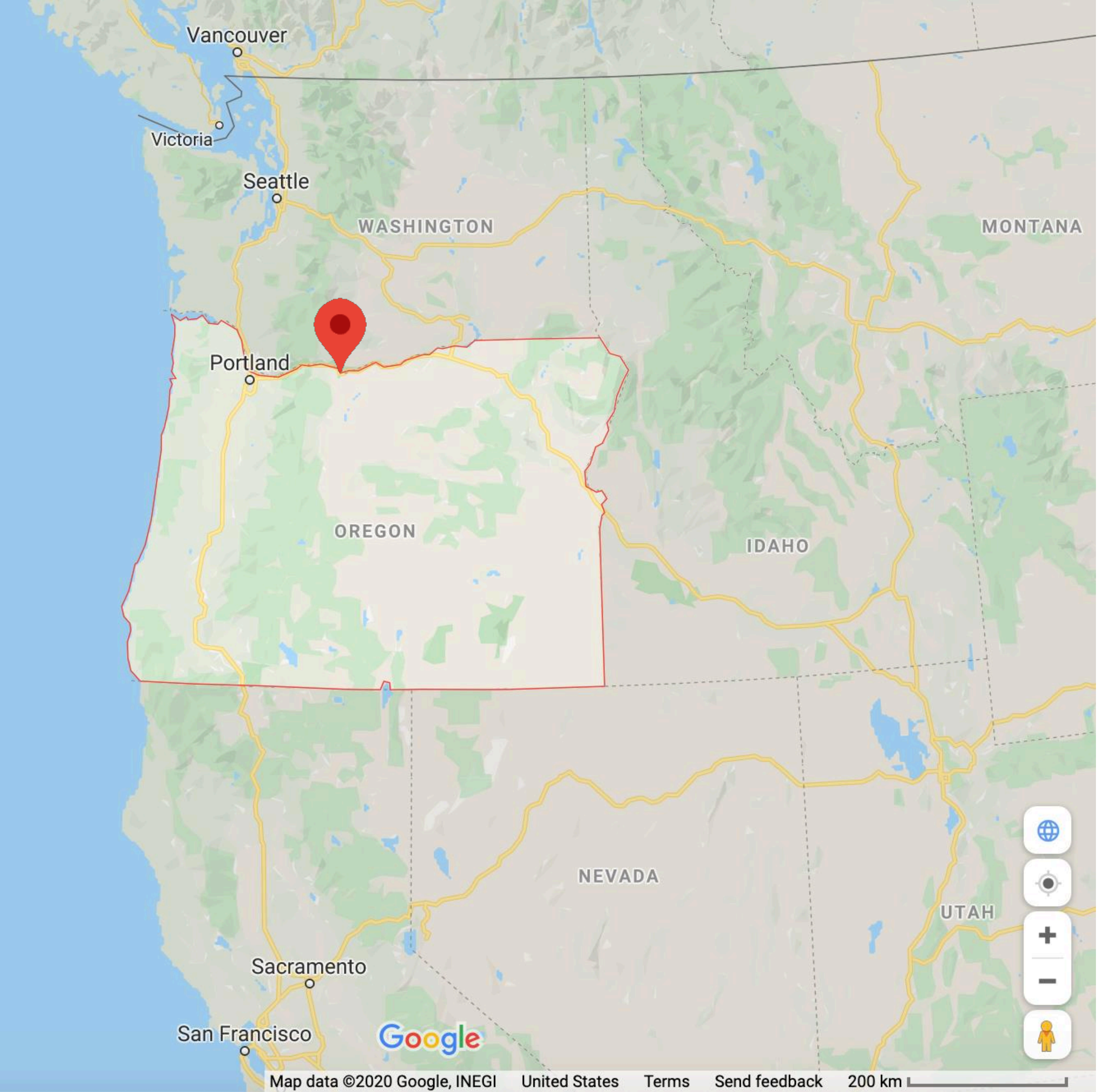
Earth

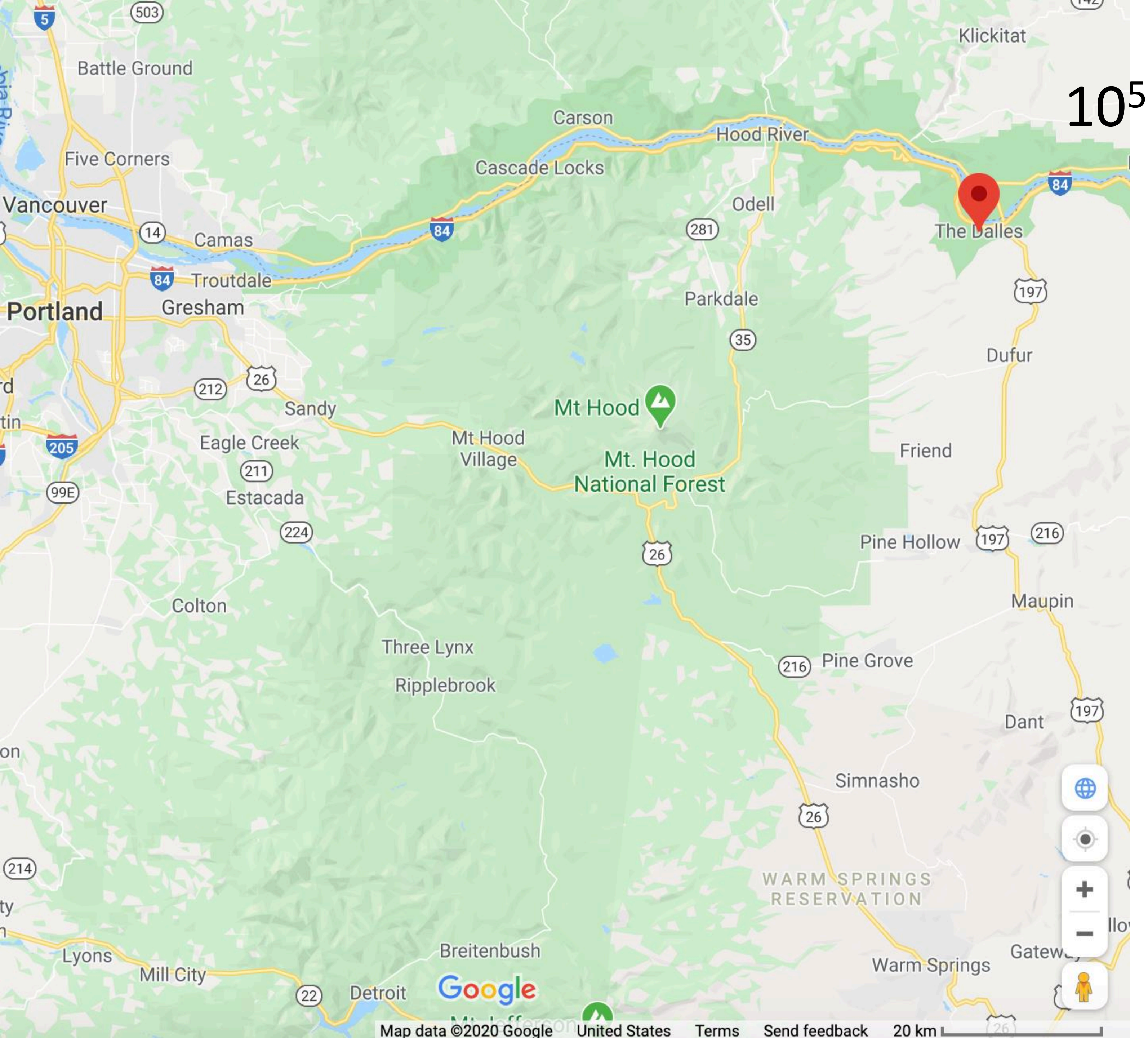
10^7 meters

10^7 meters


Kazakhstan
Turkey
Mongolia
Russia
Finland
Poland
Barents Sea
Denmark
Italy
France
Arctic Ocean
Spain
South Korea
Sea of Japan
Iceland
Portugal
Sea of Okhotsk
Japan
Arctic Ocean
Philippine Sea
Arctic Ocean
Baffin Bay
Greenland
Western Sahara
Labrador Sea
Bering Sea
AK
NU
NL
North Atlantic Ocean
YT
NT
Hudson Bay
Canada
QC
PE
MB
AB
ON
ME
BC
SK
NH
NY
North Pacific Ocean
ND
MN
WI
MI
PA
WA
MT
SD
WV
VA
OR
IA
IL
IN
KY
ID
United States
TN
SC
NV
UT
GA
Puerto Rico
MA
CA
OK
MS
AL
FL
Guyana
HI
AZ
NM
TX
LA
PA
Gulf of Mexico
Cuba
Caribbean Sea
Brazil
Mexico
Guatemala
Colombia
Ecuador
AC
Peru
Google
Map data ©2020 Google, INEGI
United States
Terms
Send feedback
2000 km

10^6 meters

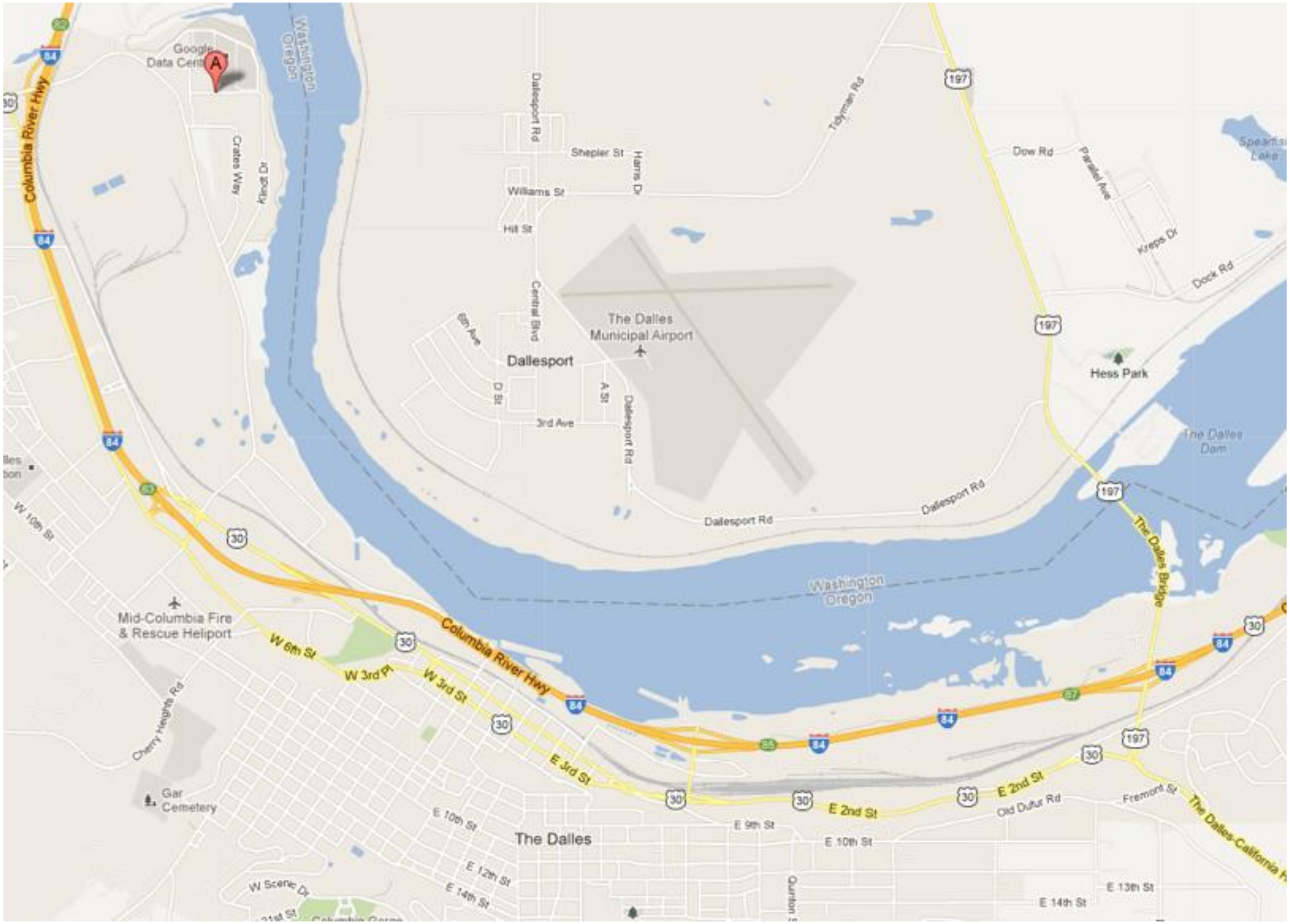

Vancouver
Victoria
Seattle
WASHINGTON
MONTANA
Portland
OREGON
IDAHO
NEVADA
UTAH
Sacramento
San Francisco
Google
Map data ©2020 Google, INEGI
United States
Terms
Send feedback
200 km


Battle Ground
Five Corners
Vancouver
Camas
Troutdale
Portland
Gresham
Carson
Cascade Locks
Hood River
Odell
Klickitat
The Dalles
Parkdale
Dufur
Sandy
Eagle Creek
Estacada
Mt Hood Village
Mt Hood
Mt. Hood National Forest
Friend
Pine Hollow
Maupin
Colton
Three Lynx
Ripplebrook
Pine Grove
Dant
Simnasho
WARM SPRINGS RESERVATION
Warm Springs
Breitenbush
Lyons
Mill City
Detroit
Google
Map data ©2020 Google
United States
Terms
Send feedback
20 km

10^5 meters

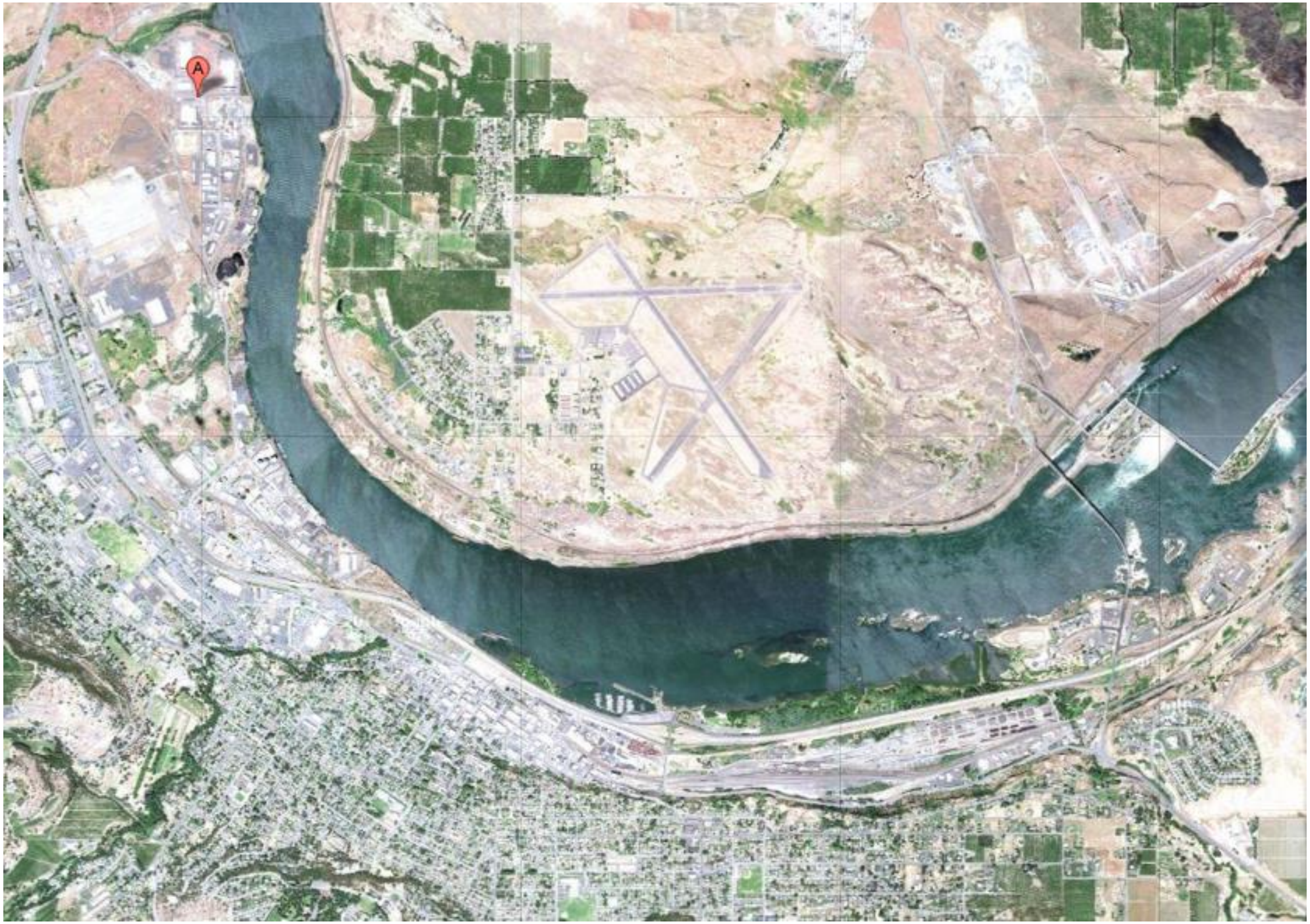
The Dalles, Oregon

10^4 meters



The Dalles, Oregon

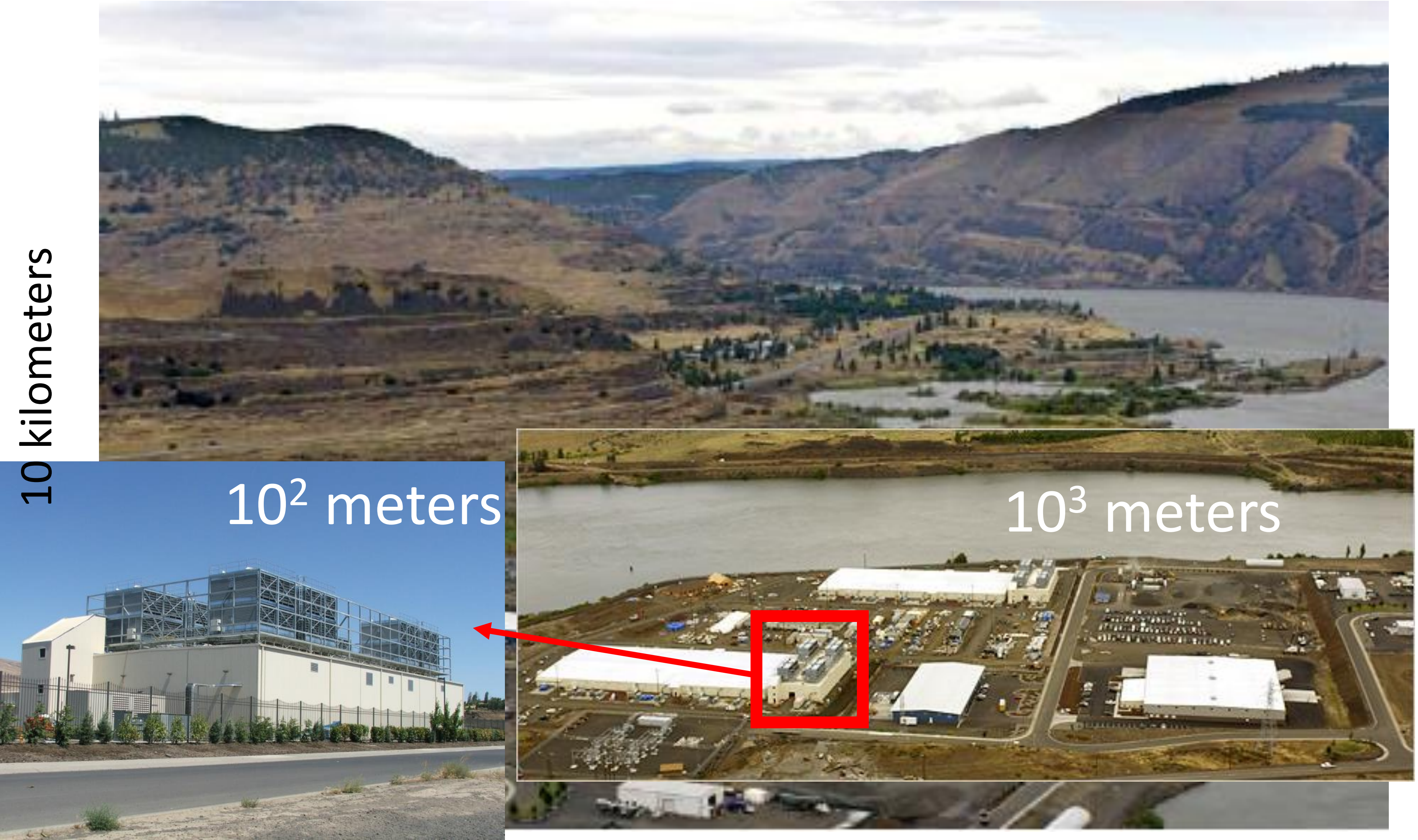
10^4 meters



Google's Oregon WSC 10^3 meters

10^4 meters

Google's Oregon WSC



Google Warehouse

- 90 meters by 75 meters, 10 Megawatts
- Contains 40,000 servers, 190,000 disks
- Power Utilization Effectiveness: 1.23
  - 85% of 0.23 overhead goes to cooling losses
  - 15% of 0.23 overhead goes to power losses
- Contains 45, 40-foot long containers
  - 8 feet × 9.5 feet × 40 feet
- 30 stacked as double layer, 15 as single layer

Containers in WSCs

10^2 meters

100 meters

Google Container

10^1 meters

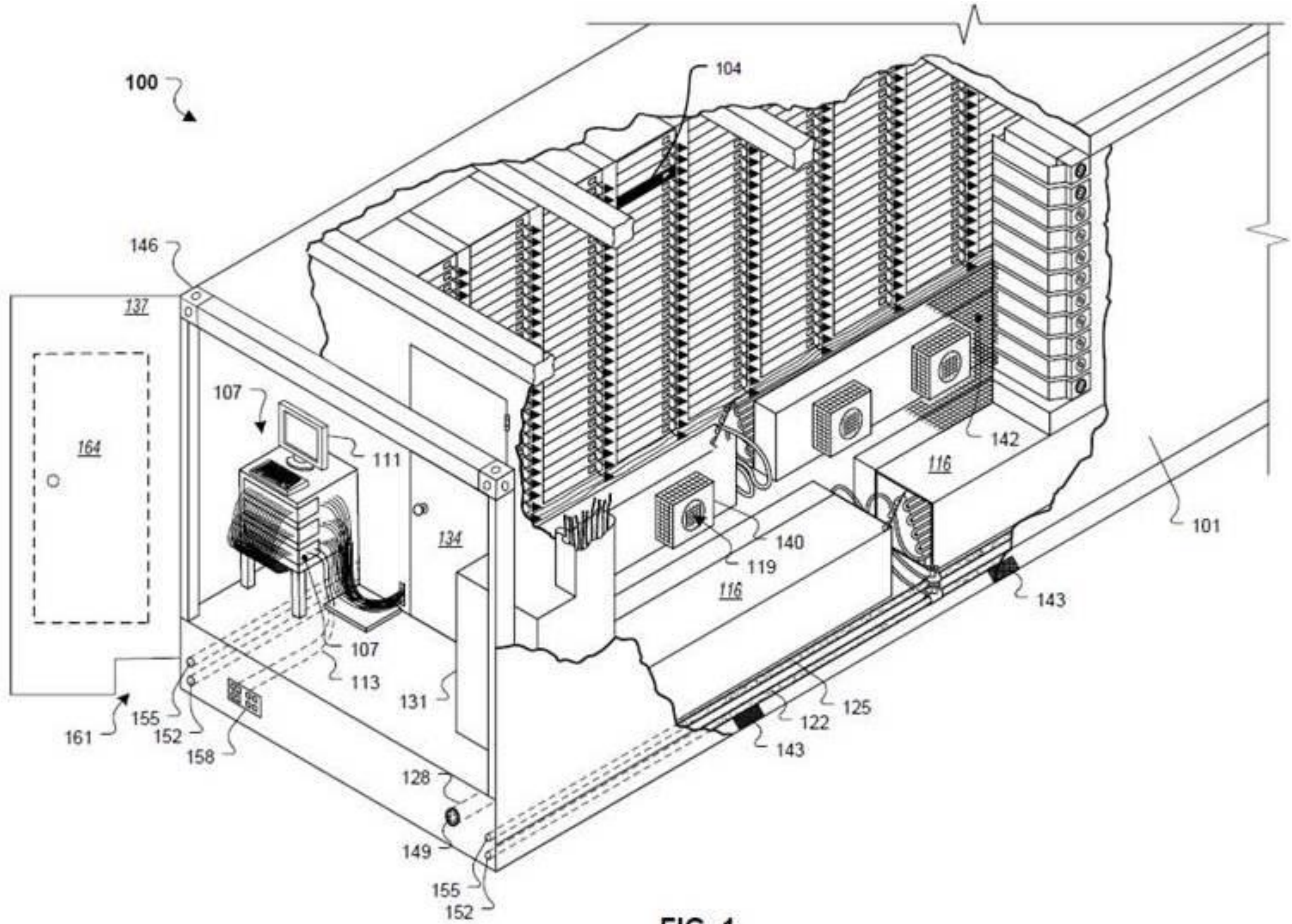
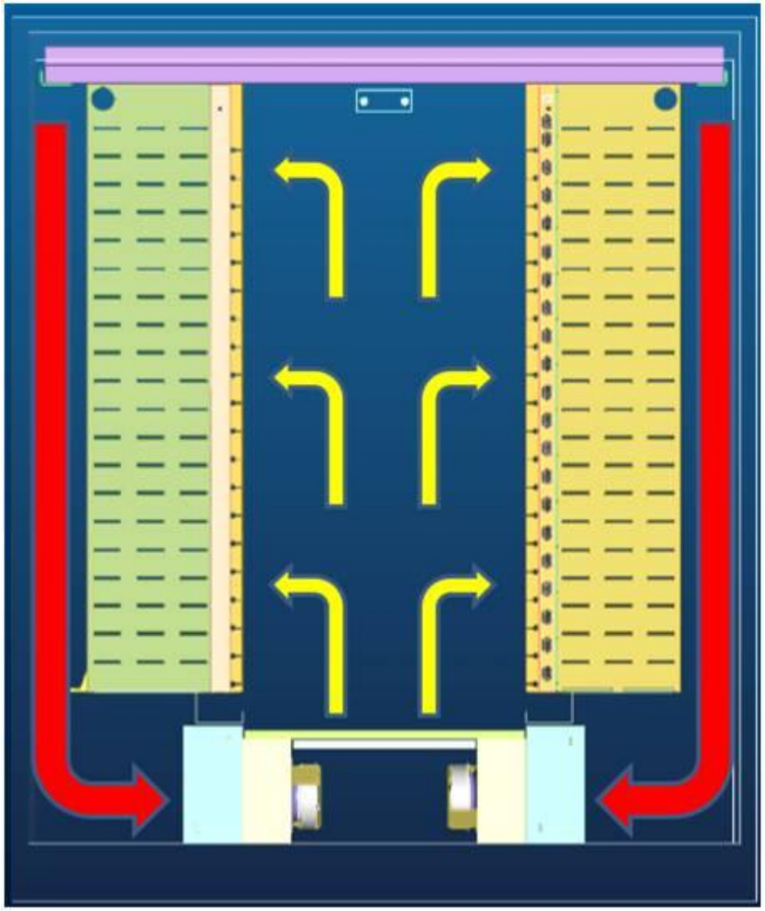


FIG. 1

10^0 meters

Google Container



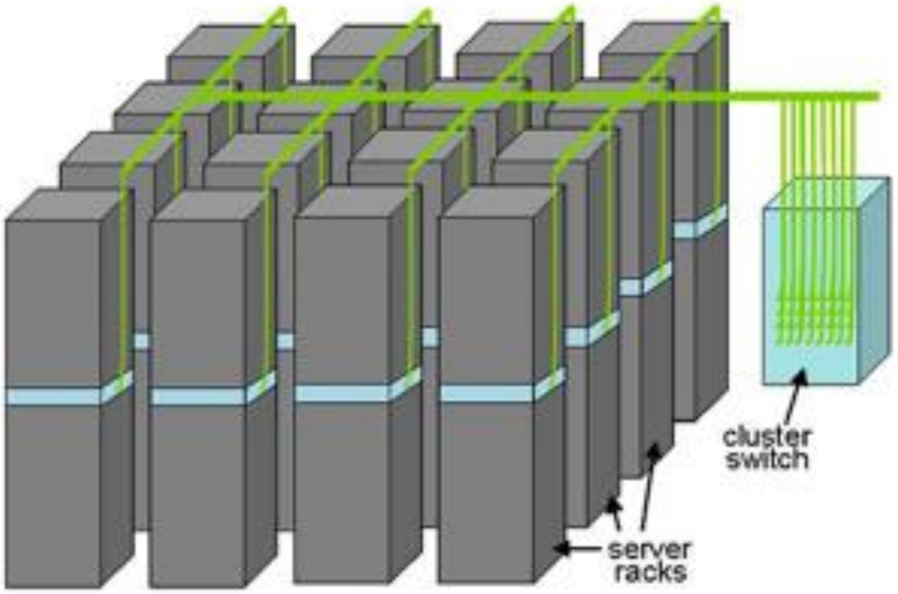
10 meters

- 2 long rows, each with 29 racks
- Cooling below raised floor
- Hot air returned behind racks

Equipment Inside a Container

Server (in rack format):

7 foot Rack: servers + Ethernet local area network switch in middle ("rack switch")



Array (aka cluster): server racks + larger local area network switch ("array switch") 10X faster => cost 100X: cost $f(N^2)$

Great Ideas in Computer Architecture

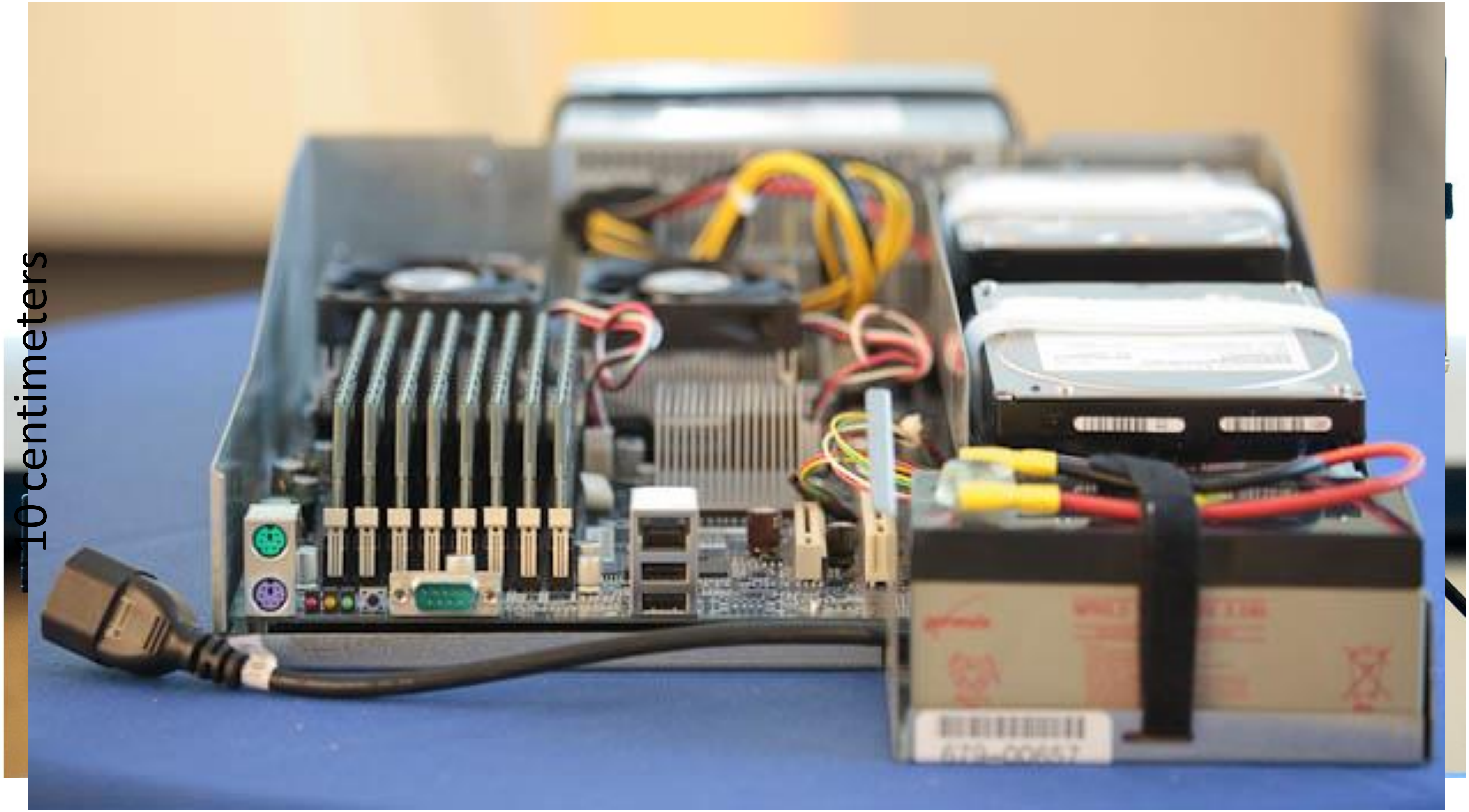
1. *Design for Moore's Law*

   -- WSC, Container, Rack
2. Abstraction to Simplify Design
3. Make the Common Case Fast
4. *Dependability via Redundancy*

   -- Multiple WSCs, Multiple Racks, Multiple Switches
5. Memory Hierarchy
6. *Performance via Parallelism/Pipelining/Prediction*

   -- Task level Parallelism, Data Level Parallelism

Google Server Internals 10^{-1} meters



10 centimeters

Facebook Datacenter

Software: Often uses MapReduce

- Simple data-parallel ***programming model*** and ***implementation*** for processing large datasets
- Users specify the computation in terms of
  - a ***map*** function, and
  - a ***reduce*** function

多台机器

- Underlying runtime system
  - Automatically ***parallelize*** the computation across large scale clusters of machines
  - ***Handles*** machine ***failure***
  - ***Schedule*** inter-machine communication to make efficient use of the networks

Programming Multicore Microprocessor: OpenMP

```
#include <omp.h>
#include <stdio.h>
static long num_steps = 100000;
int value[num_steps];
int reduce()
{
  int i;
  int sum = 0;
#pragma omp parallel for private(x) reduction(+:sum)
  for (i=1; i<= num_steps; i++){
       sum = sum + value[i];
  }
}
```

Great Ideas in Computer Architecture

1. *Design for Moore's Law*

   -- More transistors = Multicore + SIMD
2. Abstraction to Simplify Design
3. Make the Common Case Fast
4. Dependability via Redundancy
5. *Memory Hierarchy*

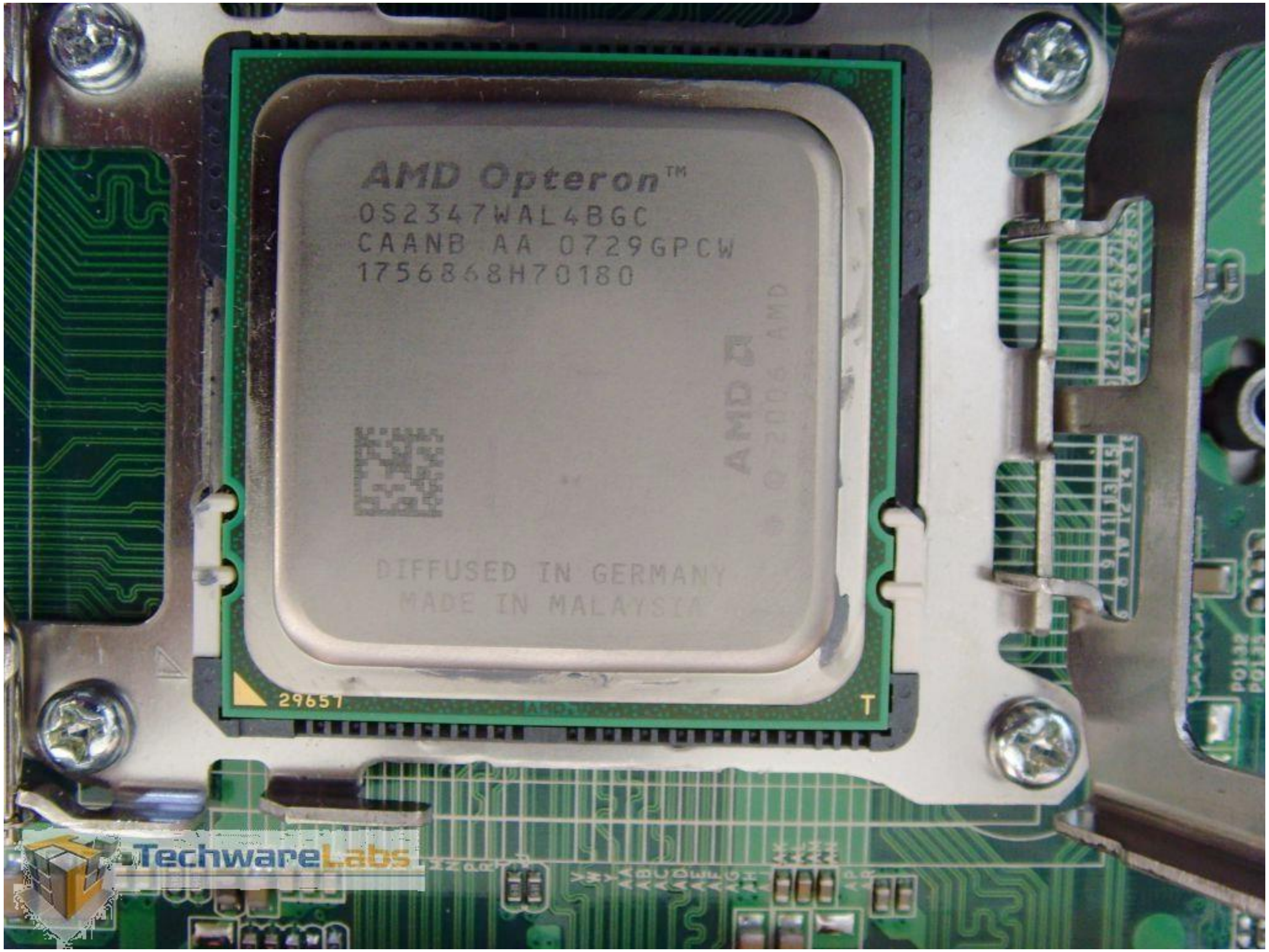
   -- More transistors = Cache Memories
6. *Performance via Parallelism/Pipelining/ Prediction*

   -- Thread-level Parallelism

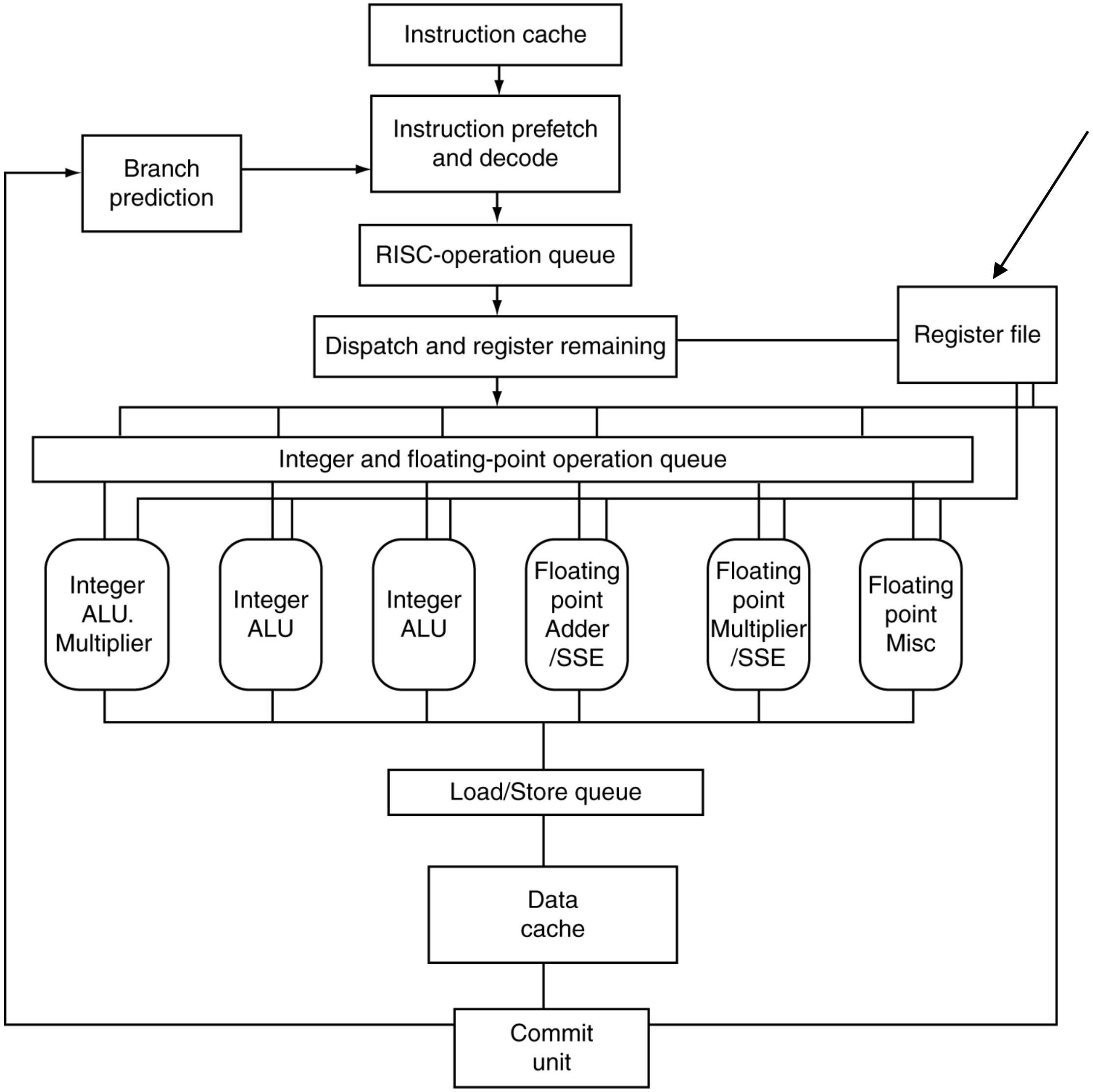
10^{-2} meters

AMD Opteron Microprocessor

centimeters



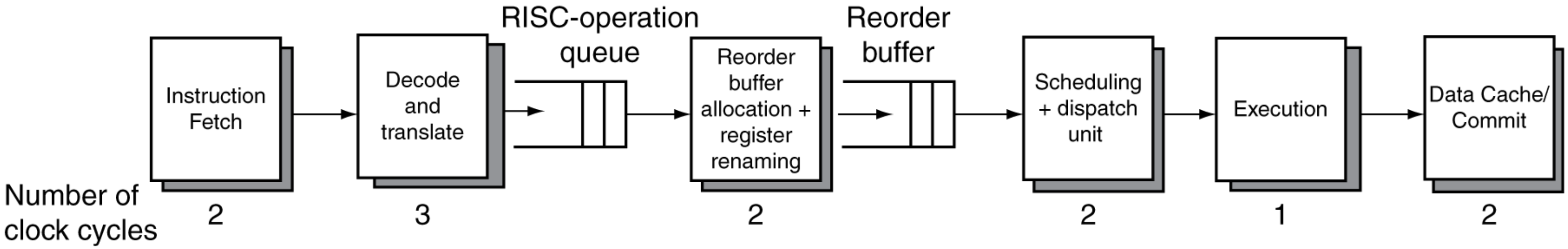
AMD Opteron Microarchitecture



72 physical registers

AMD Opteron Pipeline Flow

- For integer operations

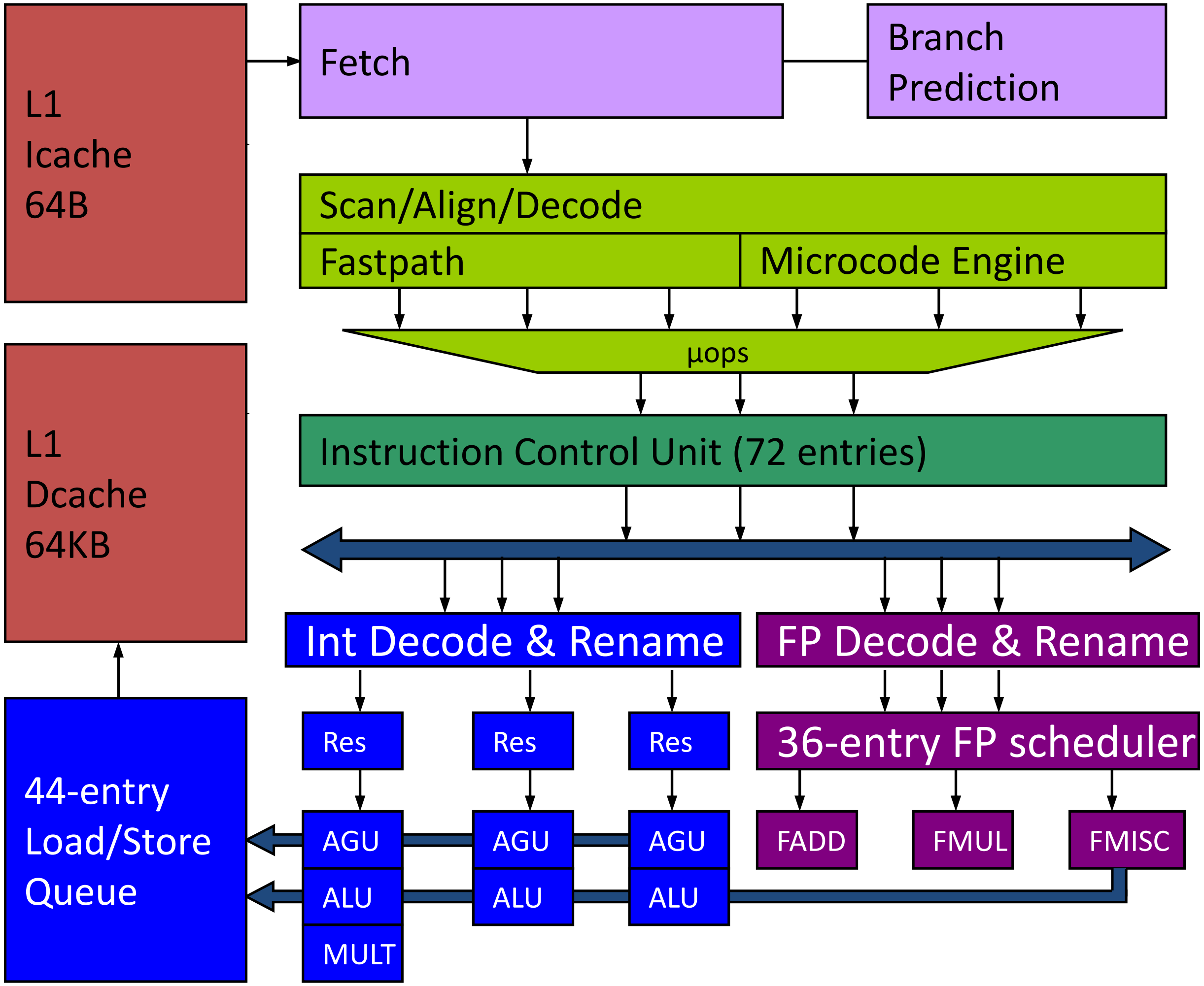


12级流水

12 stages (Floating Point is 17 stages)

Up to 106 RISC-ops in progress

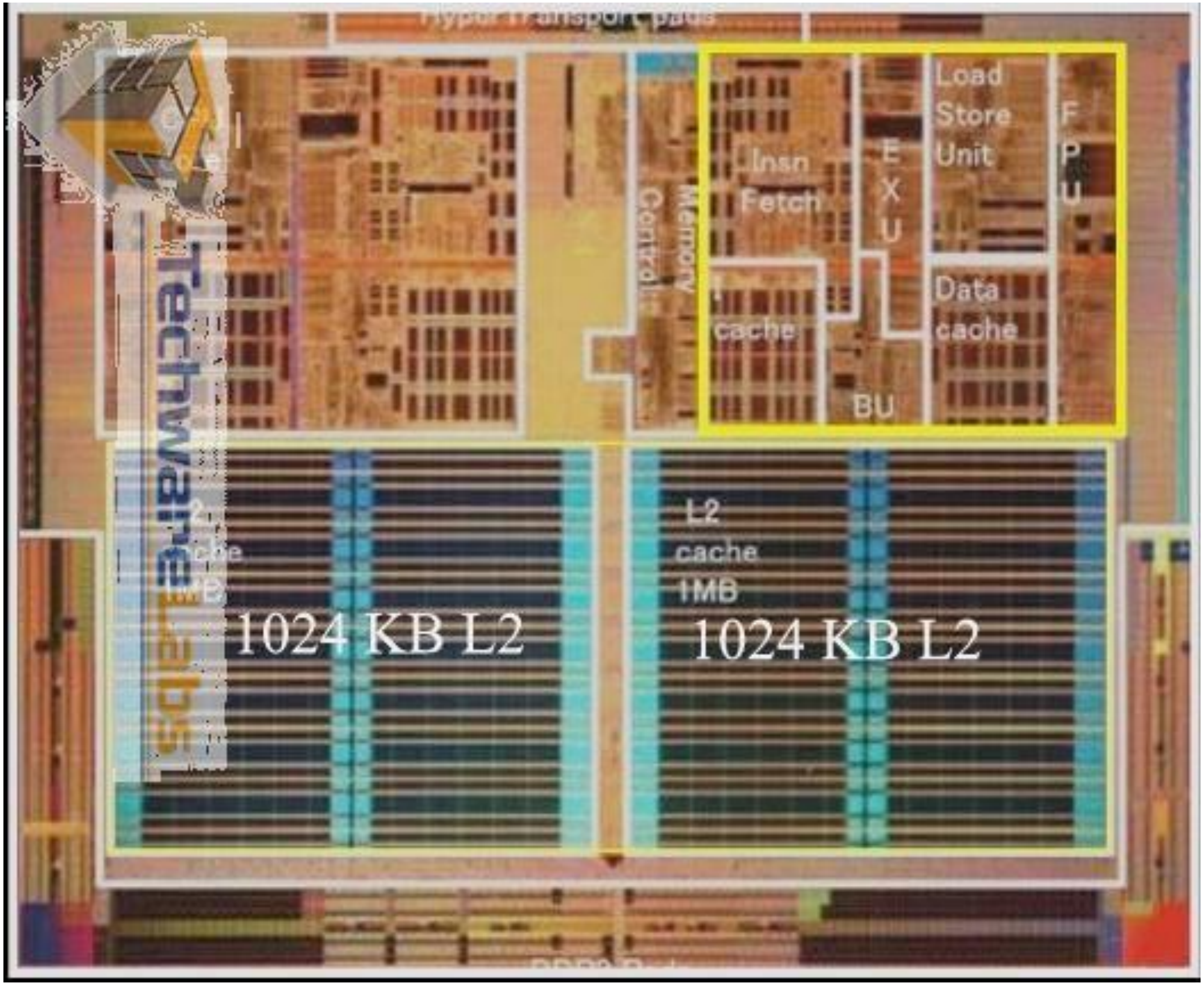
AMD Opteron Block Diagram



10^{-2} meters

AMD Opteron Microprocessor

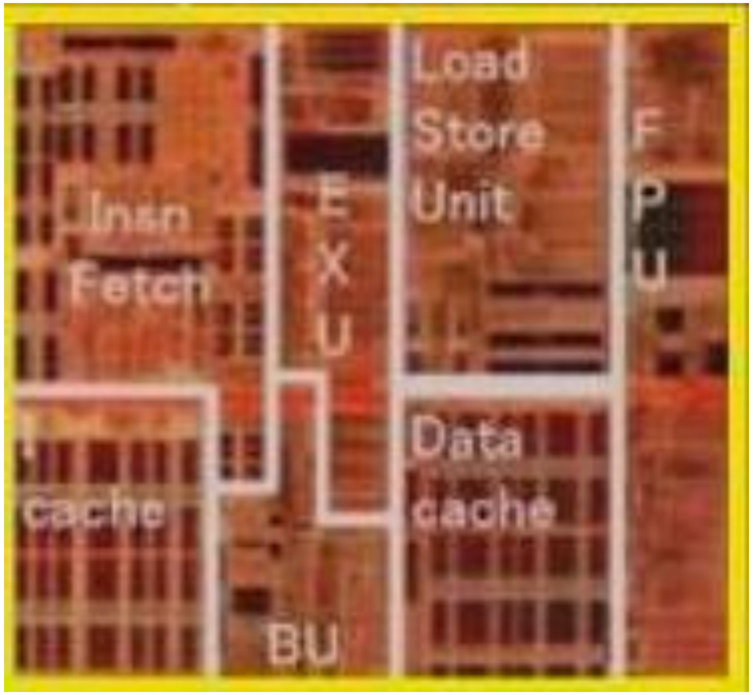
centimeters



10^{-3} meters

AMD Opteron Core

millimeters



Zoom into a Microchip

Zoom into a Microchip

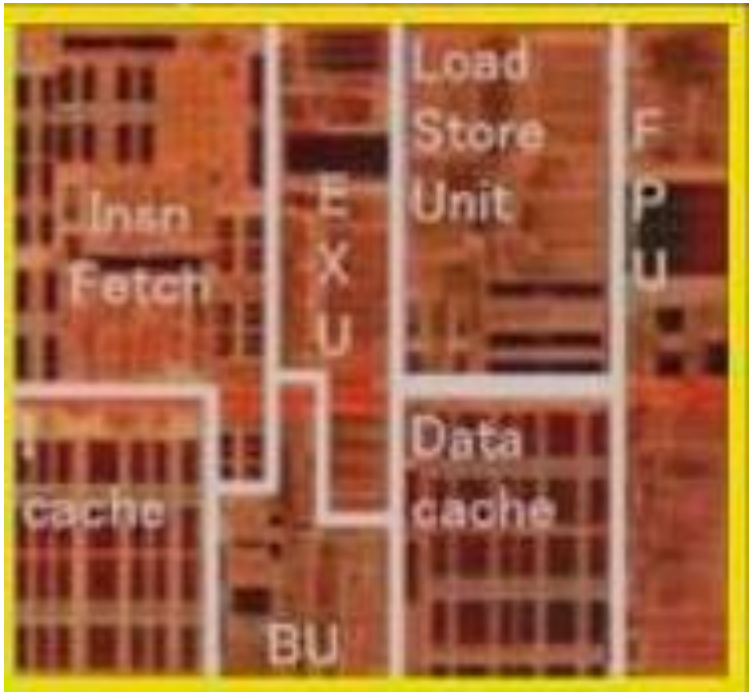
From a Digital Camera
To a Scanning Electron
Microscope

Produced by
NISE Net

10^{-3} meters

AMD Opteron Core

millimeters



Programming One Core: C with Intrinsics

```
void mmult(int n, float *A, float *B, float *C)
{
 for ( int i = 0; i < n; i+=4 )
  for ( int j = 0; j < n; j++ )
  {
   __m128 c0 = _mm_load_ps(C+i+j*n);
   for( int k = 0; k < n; k++ )
    c0 = _mm_add_ps(c0,
     _mm_mul_ps(_mm_load_ps(A+i+k*n),
             _mm_load1_ps(B+k+j*n)));
    _mm_store_ps(C+i+j*n, c0);
  }
}
```

p : pack

s : single 单精度

d : double 双精度

C[i][j]=A[i][k]*B[k][j]

What are *p* and *s* for?

Inner loop from gcc –O -S

Assembly snippet from innermost loop:

```
movaps  (%rax), %xmm9
mulps   %xmm0, %xmm9
addps   %xmm9, %xmm8
movaps  16(%rax), %xmm9
mulps   %xmm0, %xmm9
addps   %xmm9, %xmm7
movaps  32(%rax), %xmm9
mulps   %xmm0, %xmm9
addps   %xmm9, %xmm6
movaps  48(%rax), %xmm9
mulps   %xmm0, %xmm9
addps   %xmm9, %xmm5
```

Great Ideas in Computer Architecture

1. Design for Moore's Law
2. *Abstraction to Simplify Design*

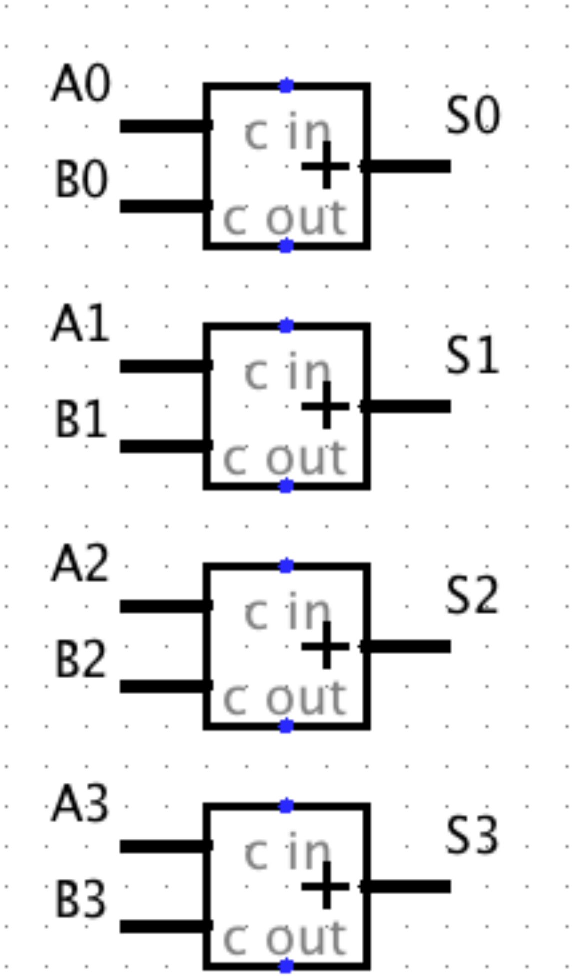
   -- Instruction Set Architecture, Micro-operations
3. Make the Common Case Fast
4. Dependability via Redundancy
5. Memory Hierarchy
6. *Performance via Parallelism/Pipelining/Prediction*

   -- Instruction-level Parallelism (superscalar, pipelining)

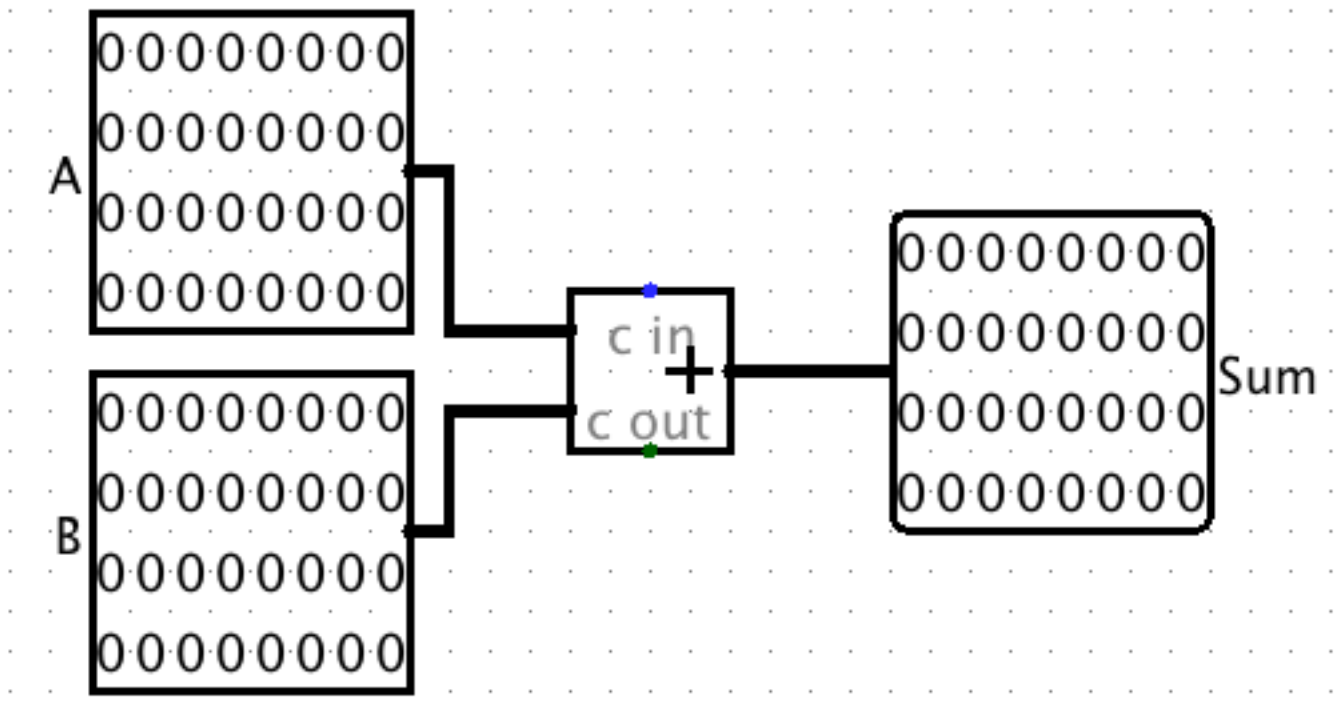
   -- Data-level Parallelism

SIMD Adder

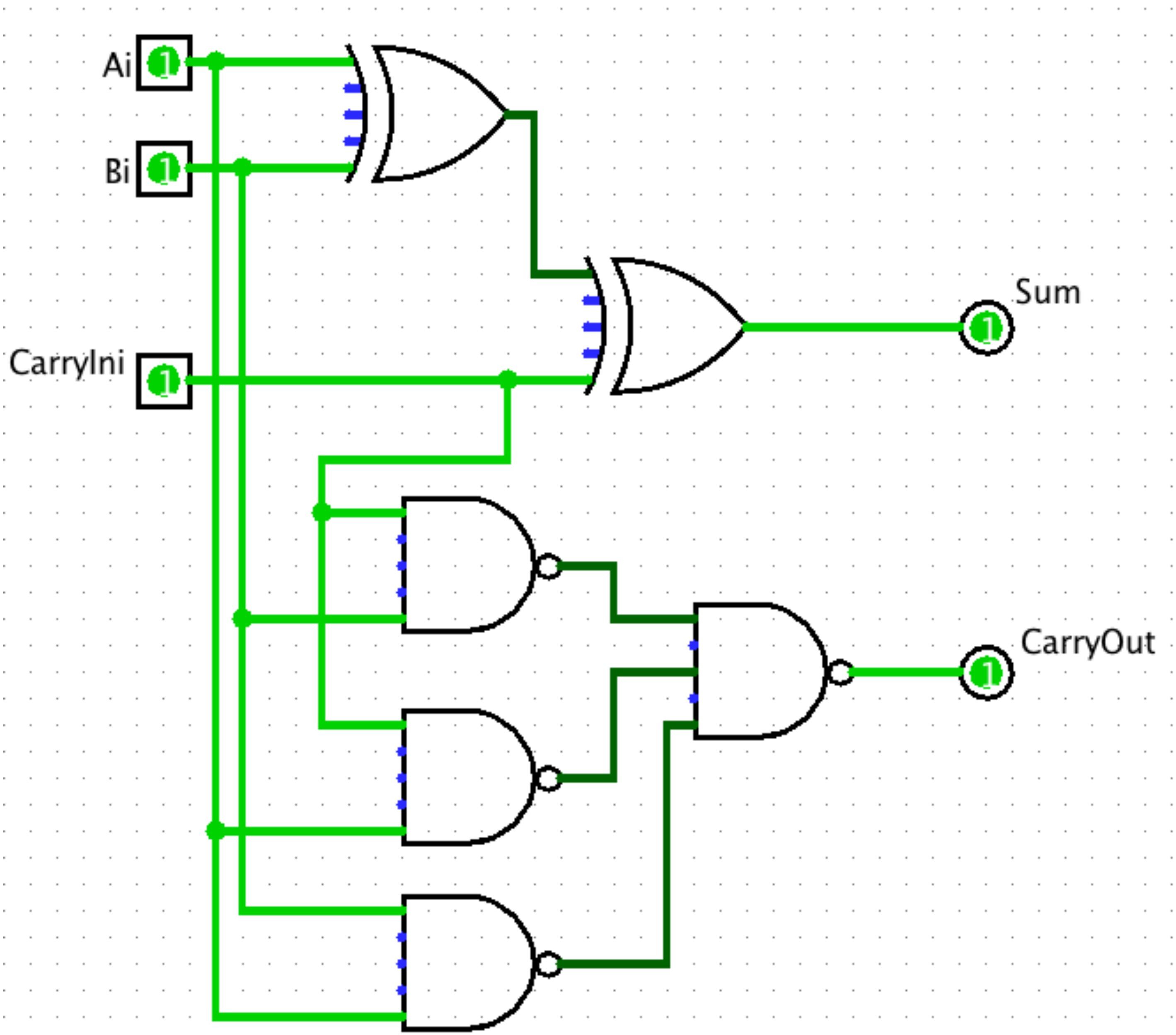
- Four 32-bit adders that operate in parallel
  - – Data Level Parallelism



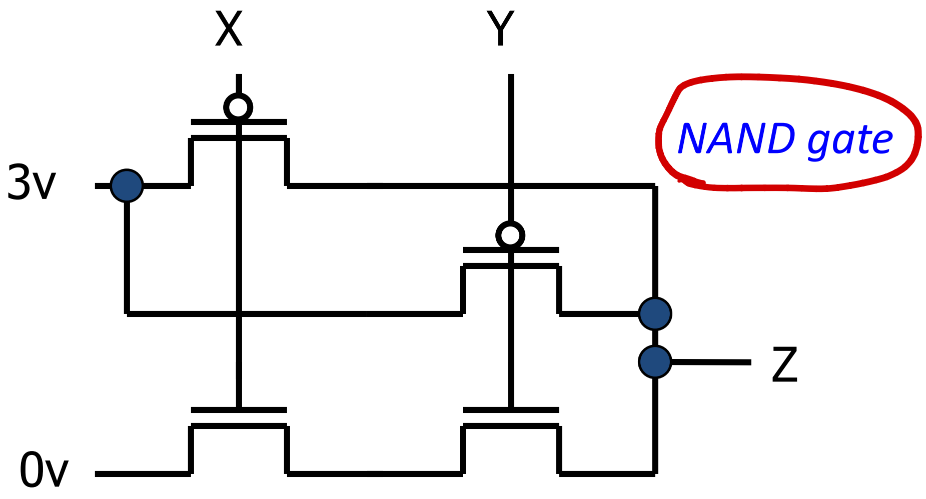
One 32-bit Adder



1 bit of 32-bit Adder



Complementary MOS Transistors (NMOS and PMOS) of NAND Gate

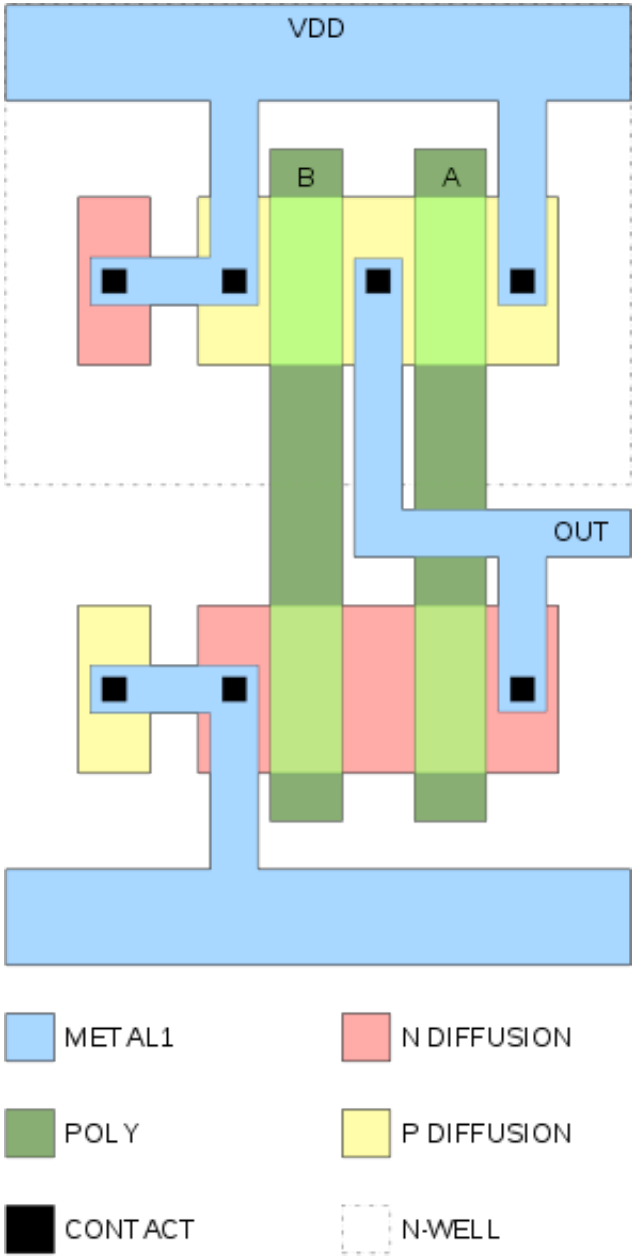


x	y	z
0 volts	0 volts	3 volts
0 volts	3 volts	3 volts
3 volts	0 volts	3 volts
3 volts	3 volts	0 volts

10^{-7} meters

Physical Layout of NAND Gate

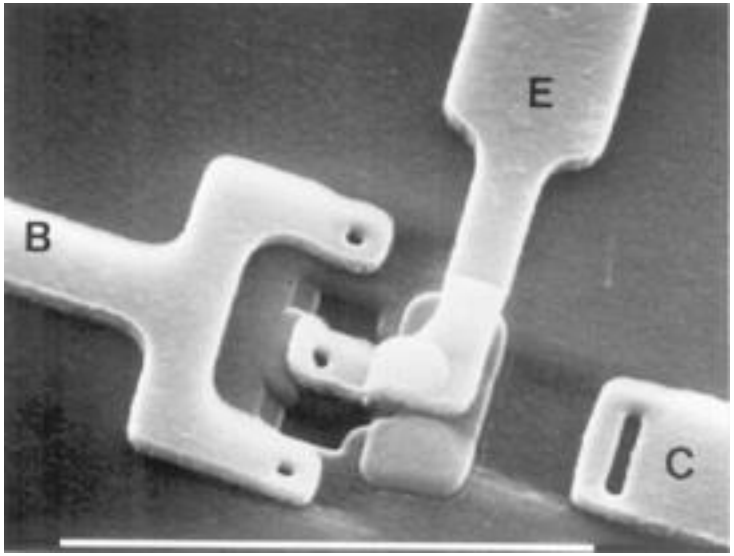
100 nanometers



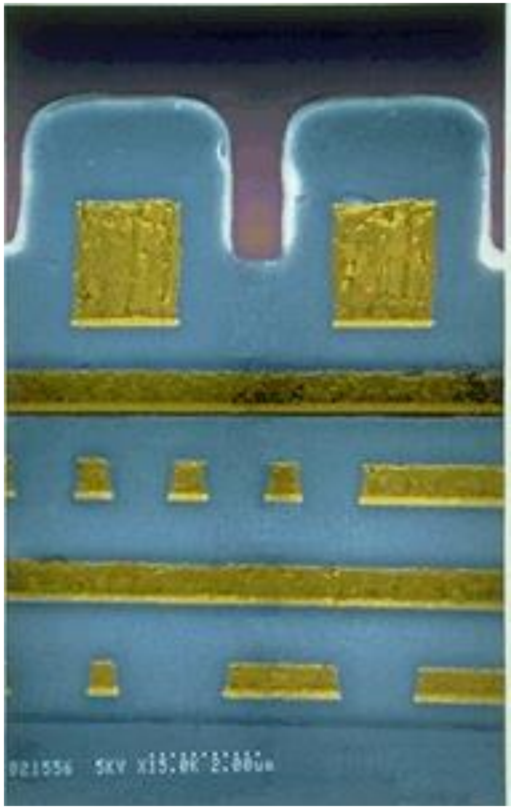
10^{-7} meters

Scanning Electron Microscope

100 nanometers



Top View

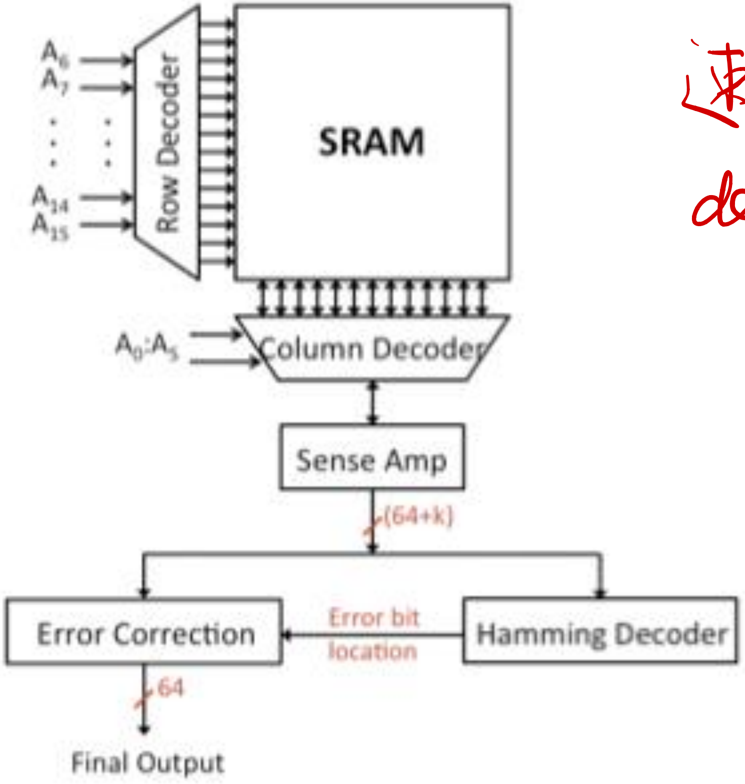


Cross Section

How to make a CMOS chip?

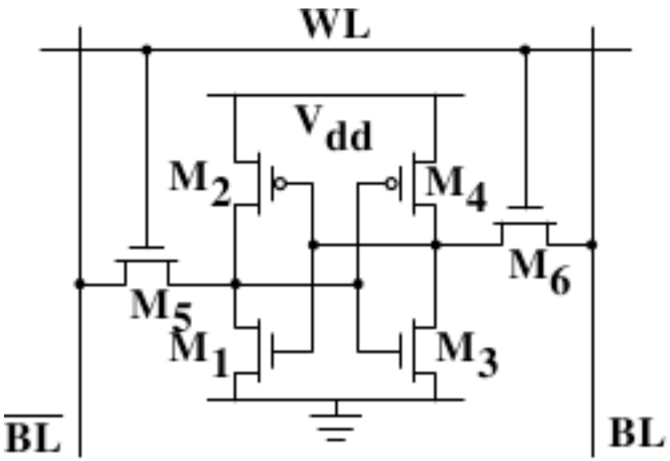
10^{-6} meters

Block Diagram of Static RAM



SRAM
速度快
density大

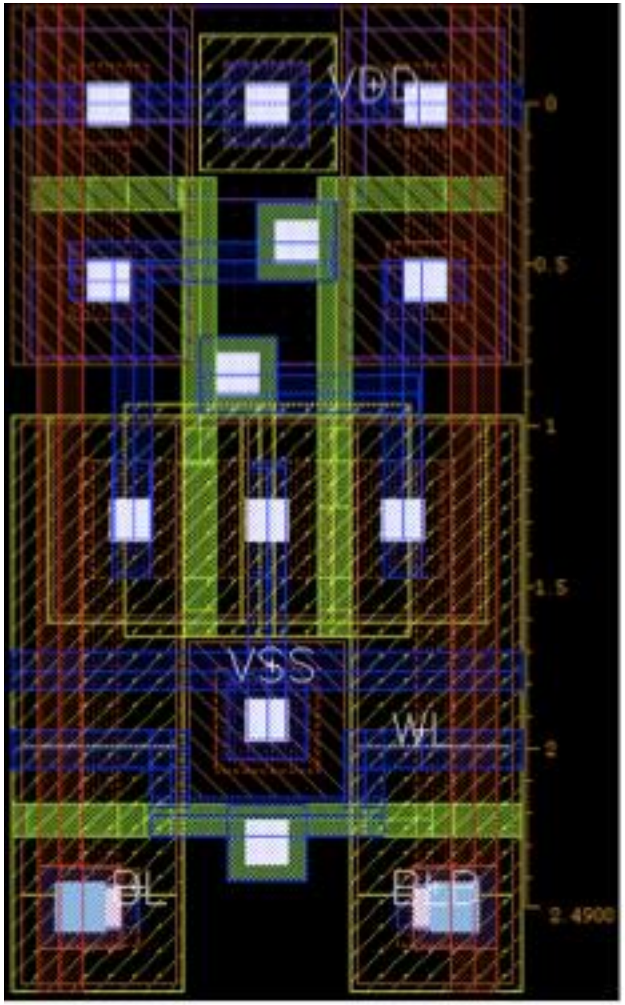
1 Bit SRAM in 6 Transistors



10^{-7} meters

Physical Layout of SRAM Bit

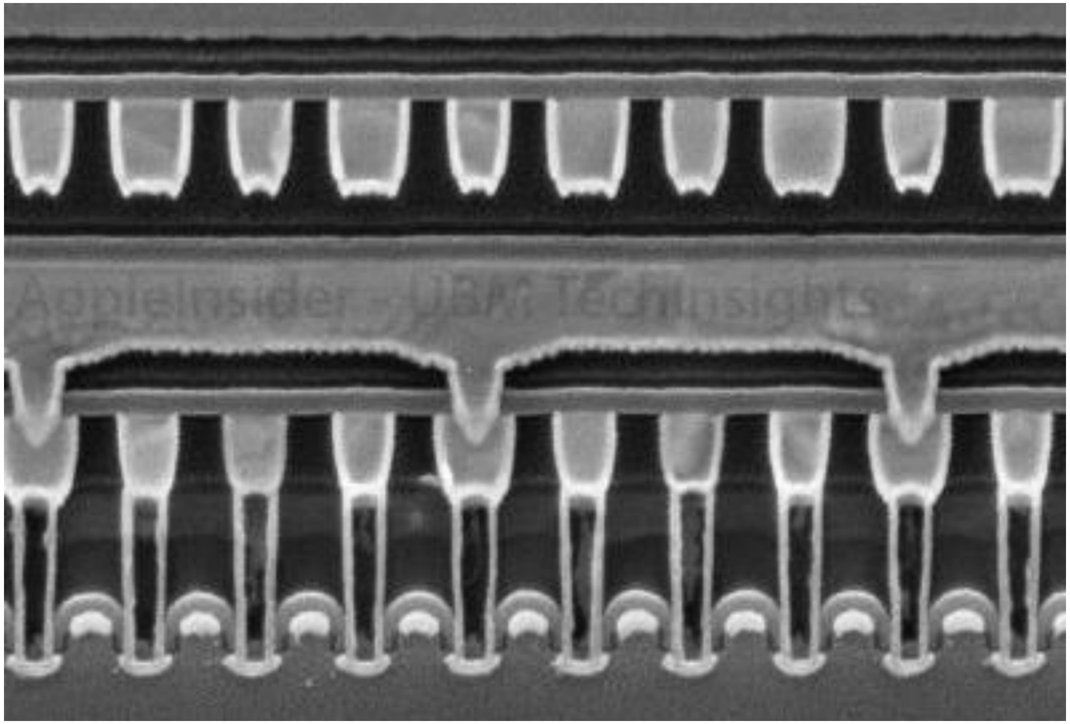
100 nanometers



10^{-7} meters

SRAM Cross Section

100 nanometers



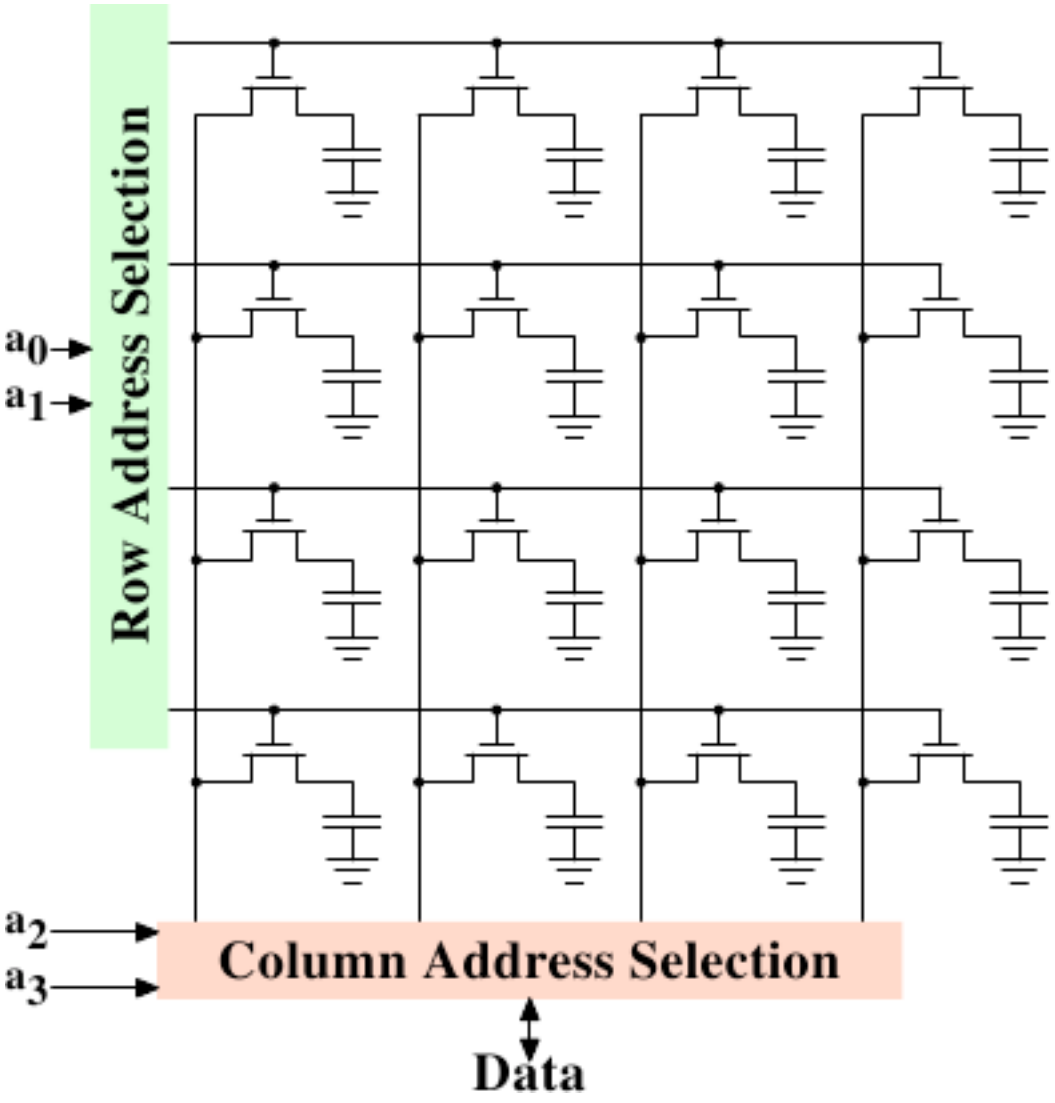
DIMM Module

- DDR = Double Data Rate
  - Transfers bits on Falling AND Rising Clock Edge
- Has Single Error Correcting, Double Error Detecting Redundancy (SEC/DED)
  - 72 bits to store 64 bits of data
  - Uses “Chip kill” organization so that if single DRAM chip fails can still detect failure
- Average server has 22,000 correctable errors and 1 uncorrectable error per year

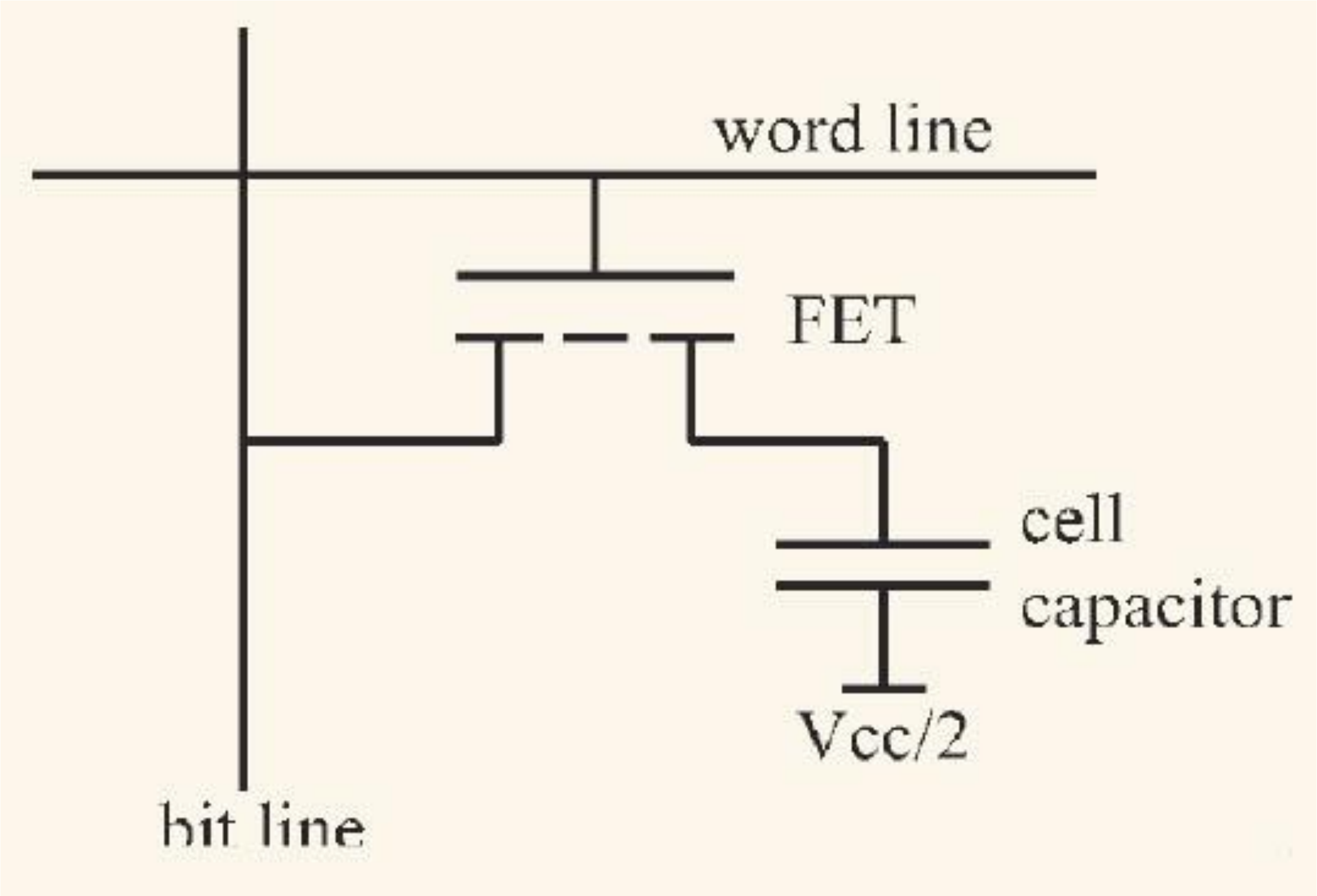
10^{-6} meters

DRAM Bits

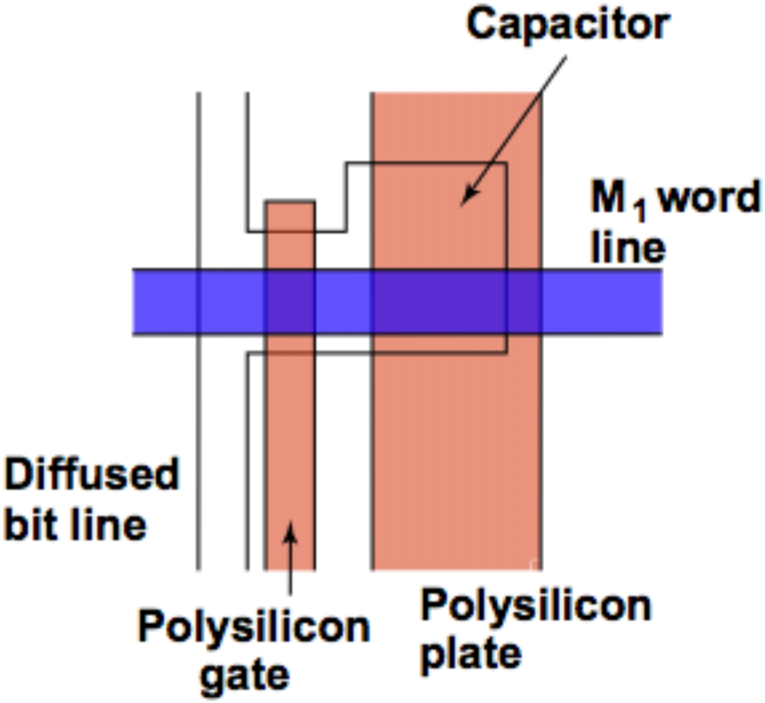
1 micron



DRAM Cell in Transistors



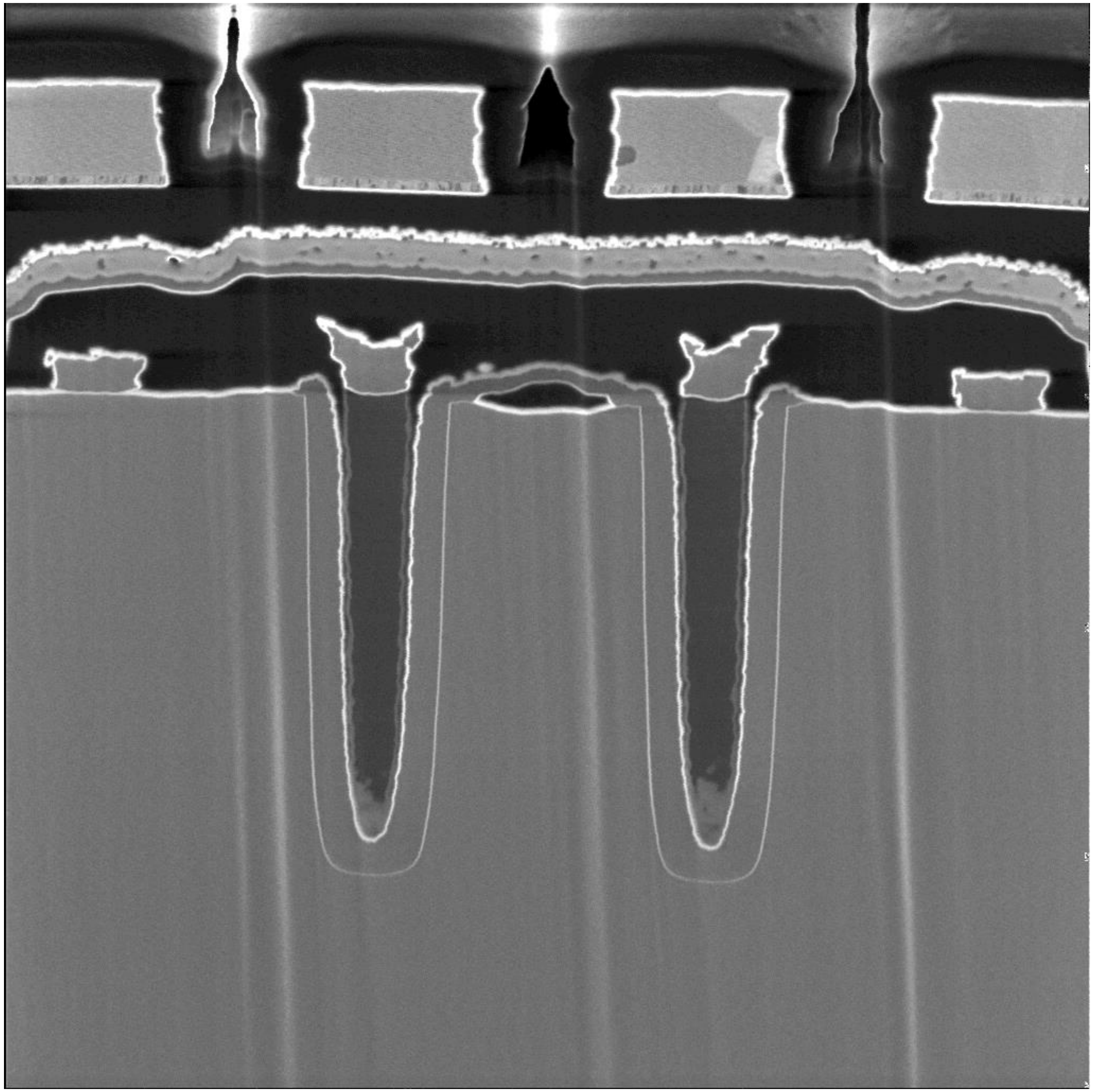
Physical Layout of DRAM Bit



10^{-7} meters

Cross Section of DRAM Bits

100 nanometers



AMD Opteron Dependability

- L1 cache data is SEC/DED protected
- L2 cache and tags are SEC/DED protected
- DRAM is SEC/DED protected with chipkill
- On-chip and off-chip ECC protected arrays include autonomous, background hardware scrubbers
- Remaining arrays are parity protected
  - Instruction cache, tags and TLBs
  - Data tags and TLBs
  - Generally read only data that can be recovered from lower levels

Programming Memory Hierarchy: Cache Blocked Algorithm

- The blocked version of the i-j-k algorithm is written simply as (A,B,C are submatricies of a, b, c)

```
for (i=0;i<N/r;i++)
  for (j=0;j<N/r;j++)
    for (k=0;k<N/r;k++)
      C[i][j] += A[i][k]*B[k][j]
```

  - r = block (sub-matrix) size (Assume r divides N)
  - X[i][j] = a sub-matrix of X, defined by block row i and block column j

Great Ideas in Computer Architecture

1. *Design for Moore's Law*

   -- Higher capacities caches and DRAM

2. Abstraction to Simplify Design
3. Make the Common Case Fast
4. *Dependability via Redundancy*

   -- Parity, SEC/DEC

5. *Memory Hierarchy*

   -- Caches, TLBs

6. *Performance via Parallelism/Pipelining/Prediction*

   -- Data-level Parallelism

Course Summary

- As the field changes, Computer Architecture courses change, too!
- It is still about the software-hardware interface
  - Programming for performance!
  - Parallelism: Task-, Thread-, Instruction-, and Data-MapReduce, OpenMP, C, SSE Intrinsics
  - Understanding the memory hierarchy and its impact on application performance