CS 110
Computer Architecture

Summary

Instructors:
Sören Schwertfeger & Chundong Wang

https://robotics.shanghaitech.edu.cn/courses/ca/21s/

School of Information Science and Technology SIST

ShanghaiTech University

Slides based on UC Berkley's CS61C

Final Exam

- Date: Thursday, June 17th, 2021
- Time: 10:30am - 12:30pm
  - Be there latest 10:15 – **we start at 10:30 sharp**!
- Venue: check on egate which room you will be in!
  - 教学中心101
    教学中心201
- Closed book:
  - You can bring **three** A4 pages with notes (both sides; in **English**): Write your Chinese and **Pinyin** name on the top! **Handwritten** by you!
  - "*Written on iPAD/Surface/…, printed out*" NOT allowed!!!
  - You will be provided with the RISC-V "green sheet"
  - No other material allowed!

Final Exam

- Switch cell phones **off**!
  (not silent mode – off!)
  - Put them in your bags.
- Bags away from you. Nothing except paper, pen, 1 drink, 1 snack, your student ID card on the table!
- No other electronic devices are allowed!
  - No ear plugs, music, smartwatch…
- Anybody touching any electronic device will **FAIL** the course!
- Anybody found cheating (copy your neighbors answers, additional material, ...) will **FAIL** the course!
- Content: Everything!

Admin


CASIO
JF-120TM
TWO WAY POWER
12 DIGITS
F CUT 5/4
4 3 2 1 0 A
MRC
TAX−
TAX RATE
TAX+
RATE SET
%
GT
ON
AC
C
+/−
7
8
9
4
5
6
1
2
3
0
00
×
−
=
+

Admin

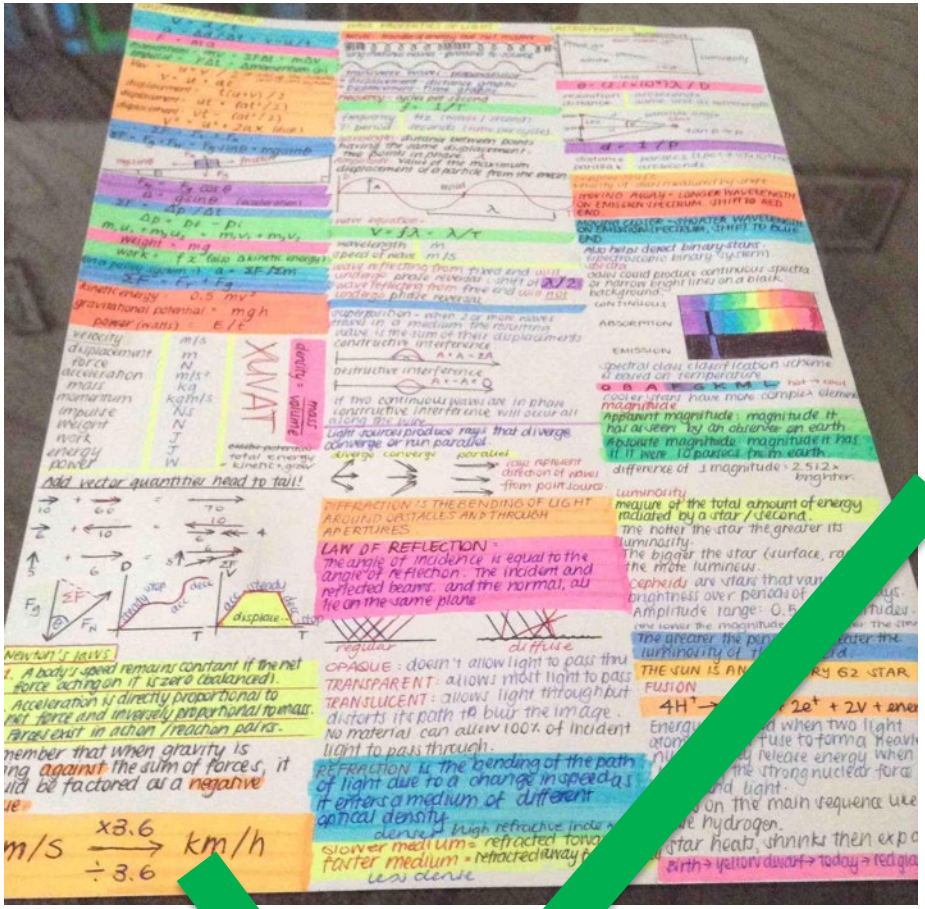
Admin

Admin

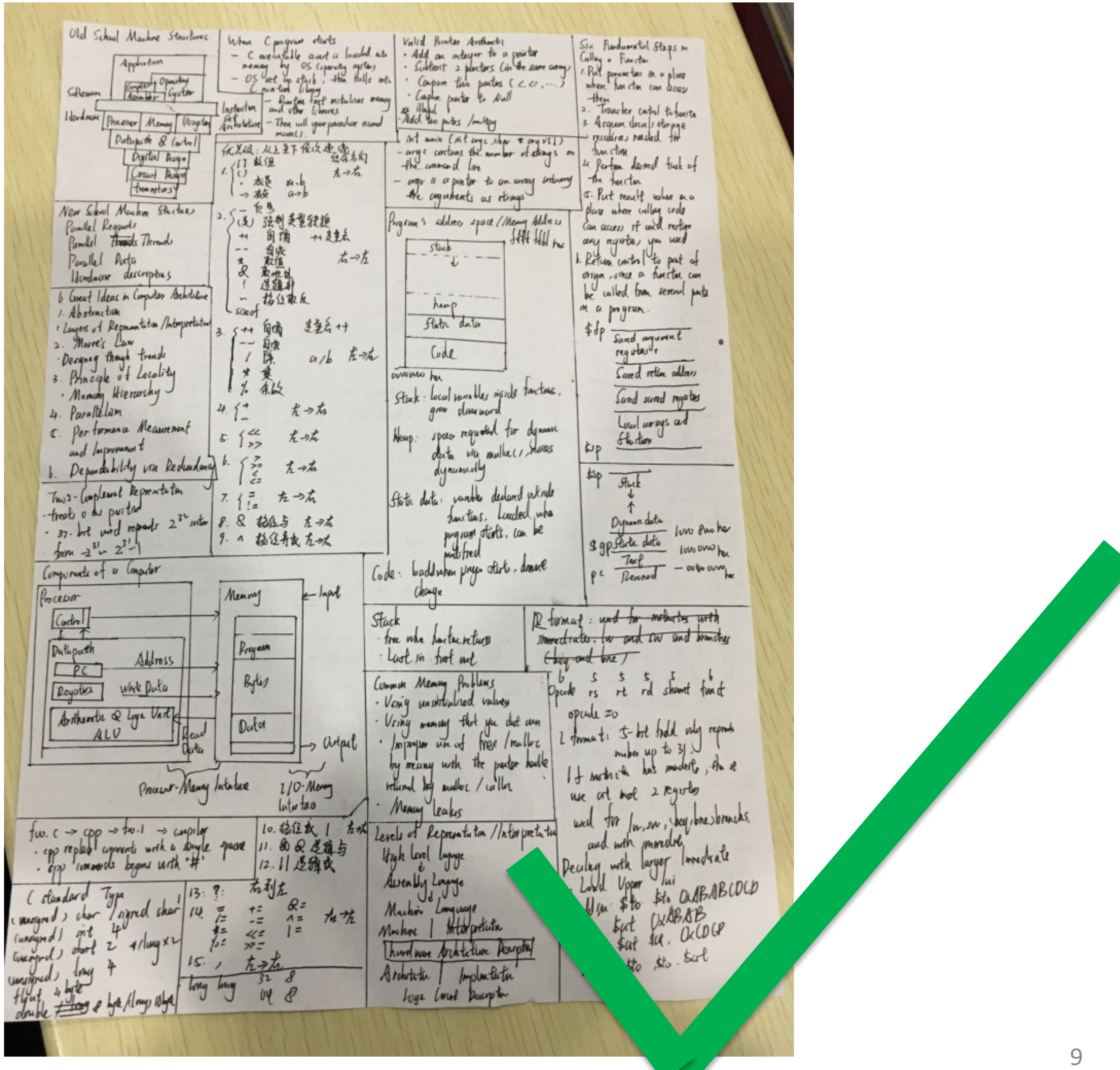

COMPUTER
ORGANIZATION
AND DESIGN
THE HA WARE/SOFTWA INTERFACE
RISC-V EDITION
MK
MORGAN KAUFMANN
DAVID A. PATTERSON
JOHN L. HENNESSY

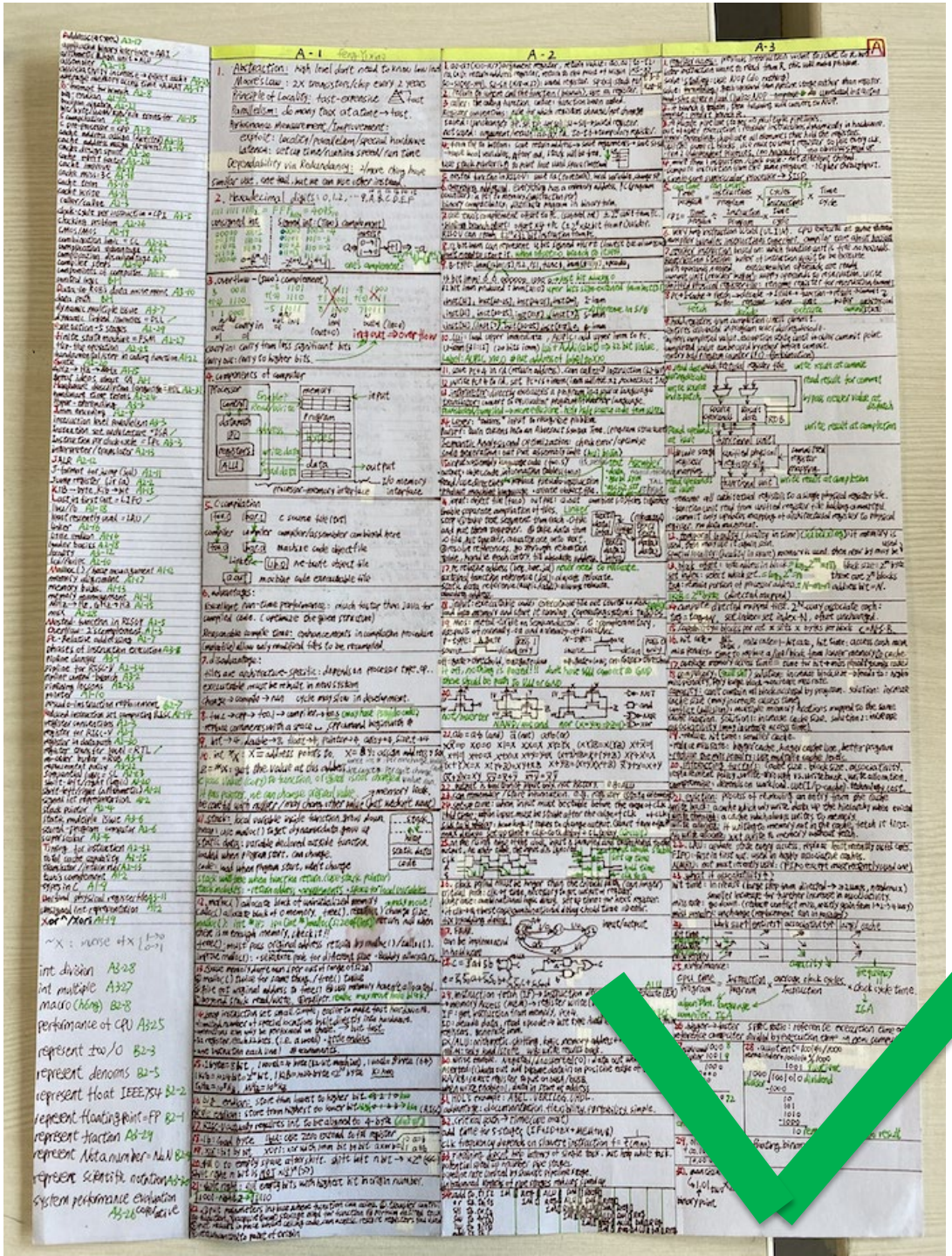
Admin


FUNCTIONS OF SEVERAL VARIABLES
LEVEL CURVES
FUNCTION OF n VARIABLES
PARTIAL DERIVATIVES
SECOND PARTIAL DERIVATIVES
CLAIRAUT'S THEOREM
THE CHAIN RULE
IMPLICIT DIFFERENTIATION
TANGENT PLANE TO A LEVEL SURFACE
NORMAL LINE TO A LEVEL SURFACE
MAXIMUM AND MINIMUM VALUES
FINDING ABSOLUTE MAX. AND MINS.
MAXIMIZING AND MINIMIZING
DOMAINS: Allowed (x,y), (x,y,z) RANGES: 2.5.W's
ε-δ DEFINITION OF CONTINUITY
EQUATIONS OF TANGENT PLANES TO SURFACES
TOTAL DIFFERENTIAL
INCREMENTS Δx, Δy, Δz DIFFERENTIALS, dx, dy, dz
THEOREM
DEPENDENCY DIAGRAMS
THE GRADIENT VECTOR
DIRECTIONAL DERIVATIVES


Add vector quantities head to tail!
LAW OF REFLECTION
REFRACTION
DIFFRACTION
Newton's laws
x3.6
km/h
÷3.6

Admin


Old School Machine Structures
Valid Pointer Arithmetic
Six Fundamental Steps in Calling a Function
New School Machine Structure
6 Great Ideas in Computer Architecture
Program's address space
Stack
heap
Static data
Code
Components of a Computer
Processor
Control
Datapath
Memory
Input
Output
Common Memory Problems
Memory leaks
Levels of Representation / Interpretation
High Level Language
Assembly Language
Machine Language



SIFT REFERENCE GUIDE (V.1.1) – CREATING TIMELINES WITH THE SIFT WORKSTATION

Admin

SANS COMPUTER FORENSICS and INCIDENT RESPONSE

THE PURPOSE OF THIS REFERENCE GUIDE IS TO WALK THROUGH THE PROCESS OF BOOTING THE SIFT WORKSTATION, CREATING A TIMELINE ("SUPER" OR "MICRO") AND REVIEWING IT.

puter-forensics11.sans.org/community/downloads

oad: SIFT Workstation VM Appliance

Do… FT Workstation Installation

2

2. BOOT SIFT VM

Login: *sansforensics*
Password: *forensics*

3

$ sudo su

3. ELEVATE PRIVS

4

4. CONNECT IMAGE TO SIFT

Plug hard drive to physical host and attach to SIFT VM

5

5. HARD DRIVE MOUNTING *(if you are using log2timeline-sift and Single DD you can skip to 7-A)*

SINGLE OR SPLIT IMAGE (2 options):

EWF/…

…nt_ewf.py image.E01 /mnt/ewf

#… …image.E01 /mnt/ewf/

Not Needed For 7-A

```
# mount -t ntfs -o ro,loop,show_sys_files,streams_interface=wind…
  offset=#### /mnt/ewf/<image> /mnt/windows_mount/
```

MOUNT TO MOUNT POINT

DD

SINGLE IM…

```
# mount -t ntfs -o …w_sys_files,streams_interface=windows,offset=#### image.dd /m…ows_mount/
```

SPLIT IMAGE (2 step proc…

```
# affuse image.001 /mnt/aff
# mount –t ntfs-3g –o loop,ro,show_…/mnt/aff/<image> /mnt/windows_…
```

6

HOW TO CALCULATE THE OFFSET FOR MOUNTING

1. Run mmls to query partition layout
 # mmls image.E01
2. Identify partition and byte offset
3. (Partition byte offset) x (bytes per sector) = **offset #### to use!**
Example: 63 X 512 = 32256

Note: If needed, repeat for each partition. Make new mount point:
***# mkdir** /mnt/windows_mount2/*

6. log2timeline default timezone is set to examiner local host. To change use -z [TIMEZONE] option. To list all available timezones:
 # log2timeline -z list

7-A & 7-B

8

7-A: AUTOMATED SUPER TIMELINE CREATION

log2timeline-sift -o –z [TIMEZONE] -p [PARTITION #] -i [IMAGE FILE]

DISK IMAGE (prompt for partition, mount, and run):

XP `# log2timeline-sift –z EST5EDT -i image`

WIN7 `# log2timeline-sift -win7 -z EST5EDT -i image`

FOR PARTITION (mount and run using all applicable pl…

XP `# log2timeline-sift –z EST5EDT -p 0 -i parti…`

WIN7 `# log2timeline-sift -win7 -z EST5EDT …tion`

OTHER USAGE EXAMPLES:

Display list of available plu…
 # log2timeline -f list
Run log2timeline use… use only specific plugins:
 # log2timeline-sift …refetch –z EST5EDT -i image.dd
Help (man page…
 # log2timeli…

…NUAL "MICRO" TIMELINE CREATION

…PTIONS] [-f FORMAT] [-z TIMEZONE] [-o OUTPUT MODULE] [-w …G_FILE/LOG_DIR [--] [FORMAT FILE OPTIONS]

…EM METADATA (using log2timeline or fls)

…stem data w/log2timeline from mounted file system:
 # log… …f mft -o mactime –r -z EST5EDT -w
mft.bo… …olume/
OR Extract M… …age using Sleuthkit:
 # fls -m "" -o offs… …dd > fls.body
Convert body file for… …format w/ mactime:
 # mactime –b fls.body –d

ARTIFACTS (run l2l on mounted file s… …h plugins recursively)

Extract artifacts w/ log2timeline and ru… …d file system:
 # log2timeline -f firefox3,chrome -o mactim… …EDT -w
web.body /mnt/volume/
Convert body file format to CSV format w/ mactime:
 # mactime –b log2timeline.body –d > log2timeline.csv

HELP? OPTIONS? USAGE?
log2timeline -help
Log2timeline-sift -help
L2t_process -help

OTHER log2timeline OUTPUT FORMATS
Note: CSV is Default Output
-BeeDocs - Mac OS X visualization tool
-CEF - Common Event Format - ArcSight
-CFTL - XML file- CyberForensics TimeLab visualization tool
-CSV - comma separated value file
-Mactime - Both older and newer version of the format supported for use by TSK's mactime
-SIMILE - XML file - SIMILE timeline visualization widget
-SQLite - SQLite database
-TLN - Tab Delimited File
-TLN - Format used by some of H Carvey tools, expressed as a ASCII output
-TLNX - Format used by some of H Carvey tools, expressed as a XML document

10

8. CSV F… …PUT (*/cases/timeline-output-folder*)

-dat… …e event, in the format of MM/DD/YYYY
-t… …of day, expressed in a 24h format, HH:MM:SS
-z: the timezone that was used to call the tool with.
-…: MACB meaning of the fields, comp w/ mactime format.
-…rce: Source short name (i.e. registry entries are REG)
-sourcetype: Desc of the source ("Internet Explorer" instead of WEBHIST)
-type: Timestamp type (i.e. "Last Accessed", "Last Written")
-user: Username associated with the entry, if one is available.
-host: Hostname associated with the entry, if one is available.
-short: Contains less text than the full description field.
-desc: where majority info is stored, the actual parsed desc of the entry.
-version: Version number of the timestamp object.
-filename: Filename with the full path that contained the entry
-inode: inode number of the file being parsed.
-notes: Some input modules insert additional information in the form of a note, which comes here. Or it can be used during the review.
-format: Input module name used to parse the file.
-extra: Additional information parsed is joined together and put here.

9

9. FILTER TIMELINE

Filter timeline with date range to include only:
l2t_process -b timeline.csv MM-DD-YYYY..MM-DD-YYYY > filtered.csv
Filter timeline with keyword list (one term per line in keywords.txt):
l2t_process -b timeline.csv -k keywords.txt > filtered.csv
What sources are in your timeline?
awk–F , '{print $6;}' timeline.csv| grep–v sourcetype|sort | uniq
Find all LNK files that reference E Drive
grep"Shortcut LNK" timeline.csv| grep"E:"
FiindMountPoints2 entries that reference E Drive
grep"MountPoints2 key" timeline.csv} | grep"E drive"
grepUSB timeline.csv| grep"SetupAPILog"

File System	M	A	C	B
Ext2/3	Modified	Accessed	Changed	N/A
FAT	Written	Accessed	N/A	Created
NTFS	File Modified	Accessed	MFT Modified	Created
UFS	Modified	Accessed	Changed	N/A

10. CONNECT TO SIFT

✓ …> SETTINGS -> OPTIONS -> Shared Fo…ers -> Always Enabled (Check)

✓ 2. SIFT Desktop > VMware-Shared-Drive

✓ Access from a Win Machine \\SIFTWORKSTATION

11

11. REVIEW TIMELINE

Review timelines using:
- Open, Soft, Filter with Excel
- Import into SPLUNK
- SIMILE
- Tapestry

log2timeline PARSING PLUGINS

apache2_error - Apache2 error log file
chrome - Chrome history file
encase_dirlisting - CSV file that is exported from encase
evt - Windows 2k/XP/2k3 Event Log
evtx - Windows Event Log File (EVTX)
exif - Metadata information from files using ExifTool
ff_bookmark - Firefox bookmark file
firefox2 - Firefox 2 browser history
firefox3 - Firefox 3 history file
ftk_dirlisting - CSV file that is exported from FTK Imager (dirlisting)
generic_linux - Generic Linux logs that start with MMM DD HH:MM:SS
iehistory - index.dat file containg IE history
iis - IIS W3C log file
isatxt - ISA text export log file
jp_ntfs_change - CSV output file from JP (NTFS Change log)
mactime - Body file in the mactime format
mcafee - Log file
mft - NTFS MFT file
mssql_errlog - ERRORLOG file produced by MS SQL server
ntuser - NTUSER.DAT registry file
opera - Opera's global history file
oxml - OpenXML document pcap
pcap - PCAP file
pdf - Available PDF document metadata
prefetch - Prefetch directory
recycler - Recycle bin directory
restore 0.9 - Restore point directory
safari - Safari History.plist file
sam - SAM registry file
security - SECURITY registry file
setupapi - SetupAPI log file in Windows XP
skype_sql - Skype database
software - SOFTWARE registry file
sol - .sol (LSO) or a Flash cookie file
squid - Squid access log (http_emulate off)
syslog - Linux Syslog log file
system - SYSTEM registry file
tln - Body file in the TLN format
volatility - Volatility output files (psscan2, sockscan2, …)
win_link - Windows shortcut file (or a link file)
wmiprov - wmiprov log file
xpfirewall - XP Firewall log

List plugins # log2timeline -f list
…HELP EXPAND THIS LIST. BUILD PLUGINS!!!

BY DAVID NIDES (12/16/2011)
TWITTER: @DAVNADS
BLOG: DAVNADS.BLOGSPOT.COM
EMAIL: DNIDES@KPMG.COM

CREDITS TO: ED GOINGS, ROB LE… KRISTINN GUDJONSSON, KPMG &…!!
QUESTIONS/FEEDBACK–CONTACT US!

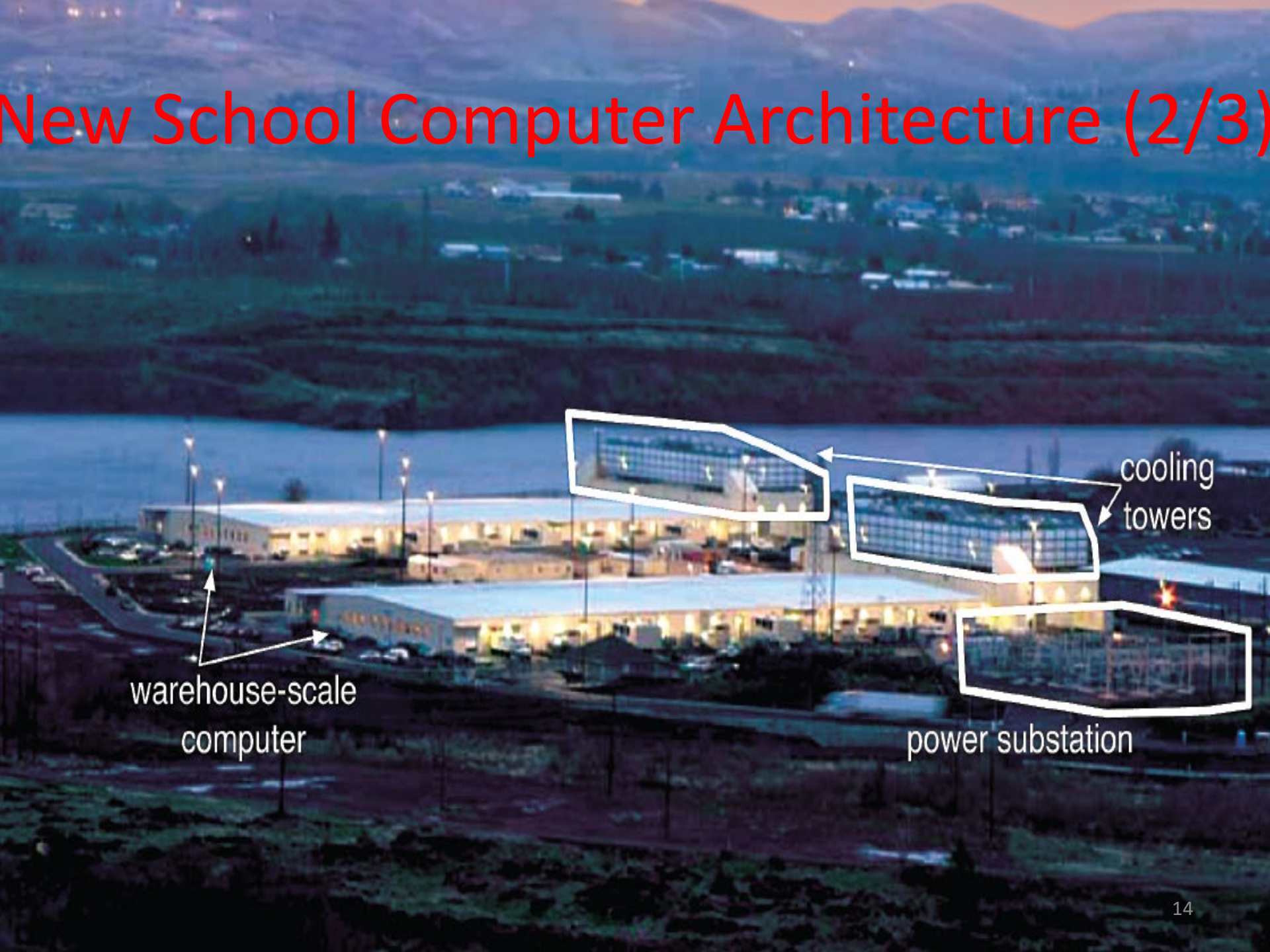
KEY
Red text – image/source
Blue text – mount point
Purple text – output file
Green text – log2timeline plugins
Brown text - TimeZone

Let us review CA now.

New School Computer Architecture (1/3)

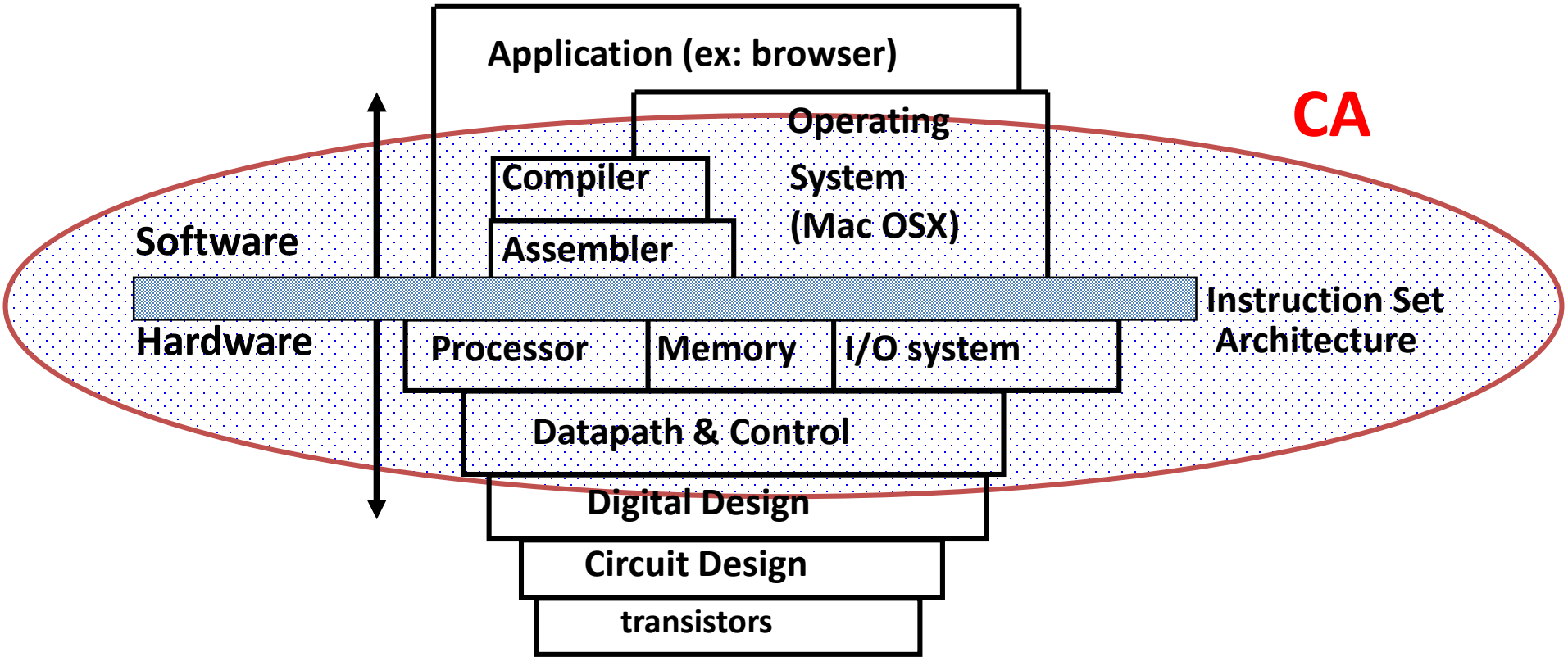
Personal Mobile Devices

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 "Katz"
- 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
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)

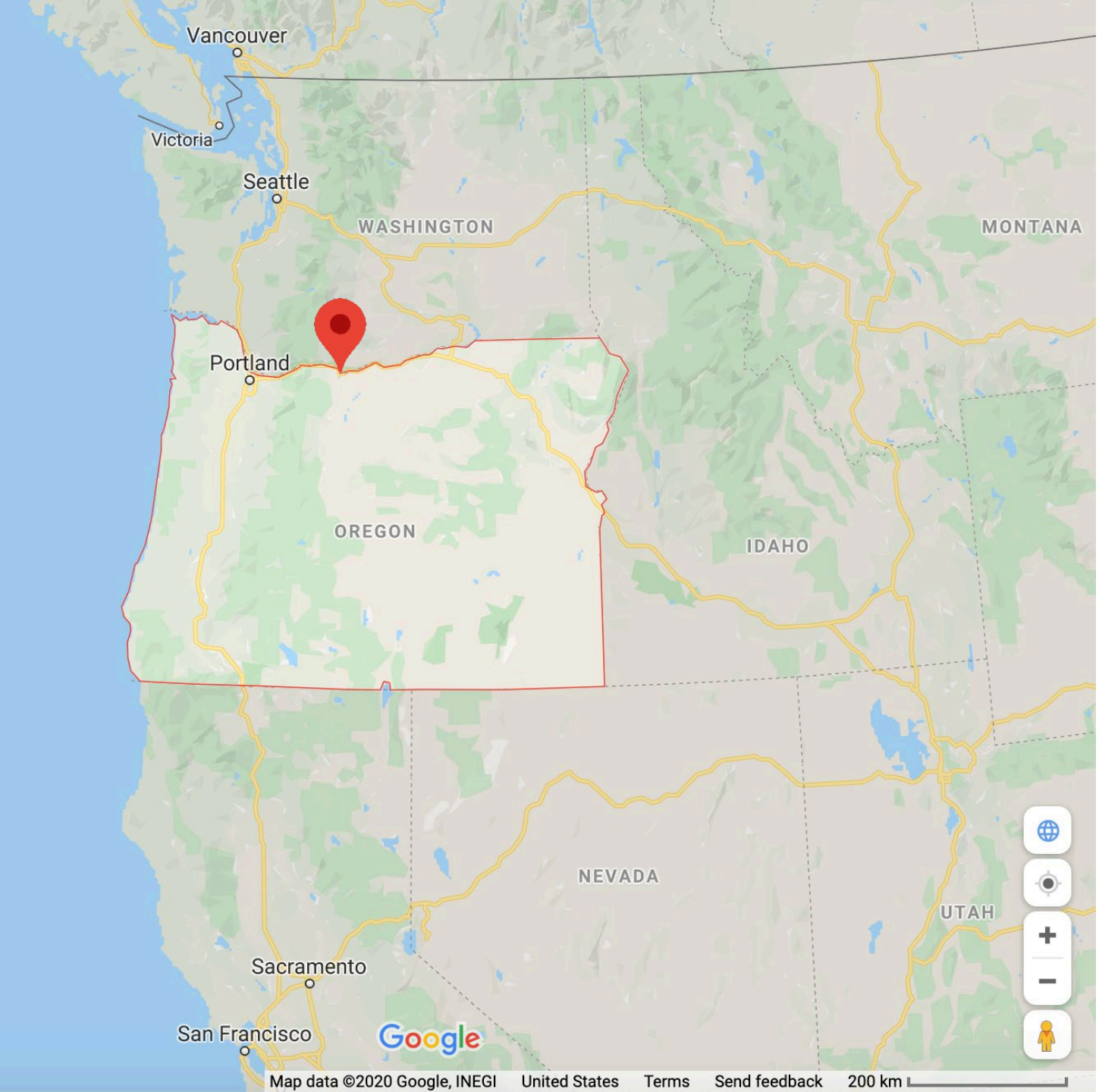
- See http://www.powersof10.com/film

Earth

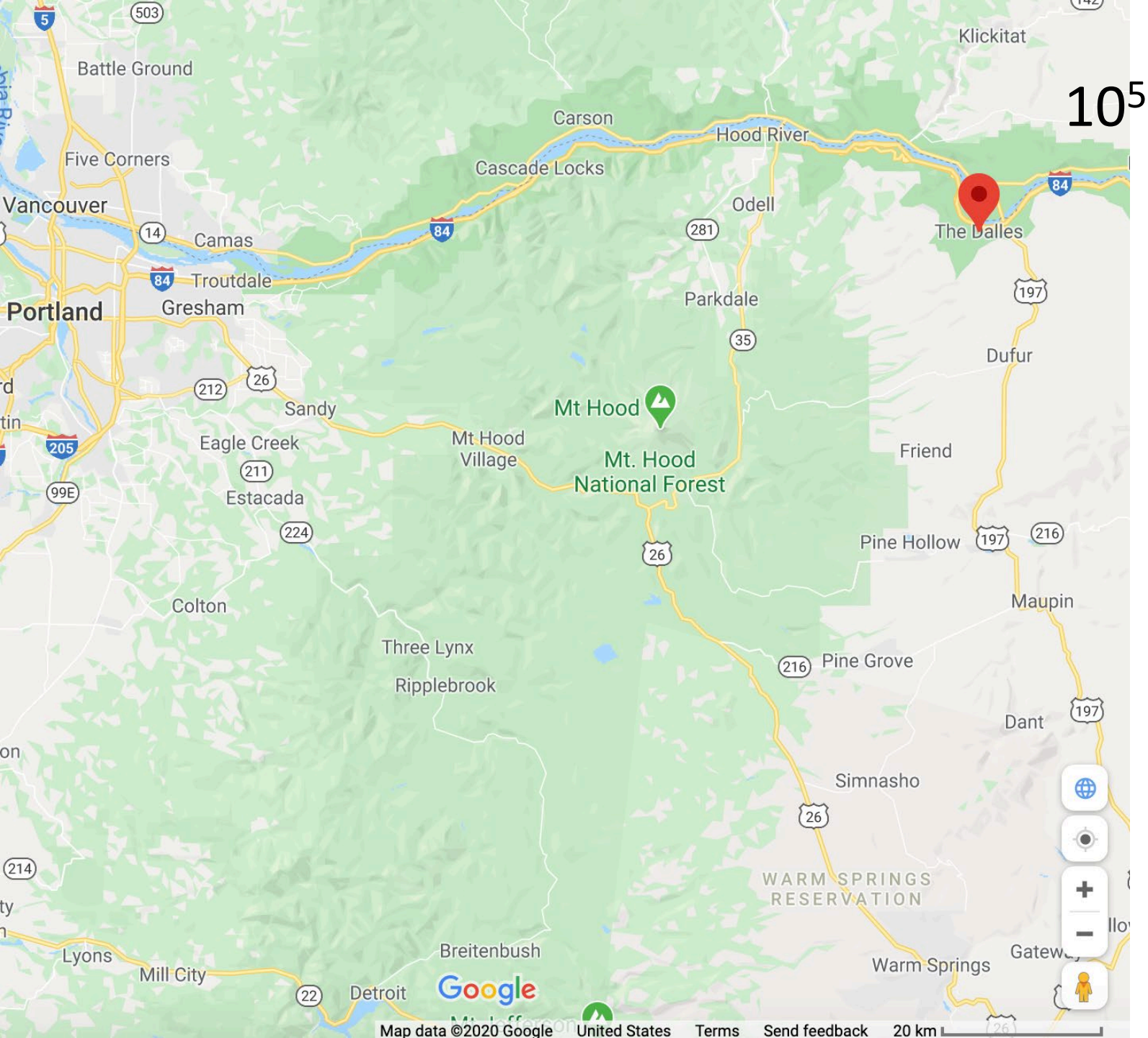
10^7 meters


Kazakhstan
Turkey
Mongolia
Russia
Finland
Poland
Barents Sea
Denmark
Italy
France
Arctic Ocean
Spain
South Korea
Iceland
Sea of Japan
Portugal
Sea of Okhotsk
Japan
Arctic Ocean
Philippine Sea
Arctic Ocean
Baffin Bay
Greenland
Western Sahara
Labrador Sea
Bering Sea
North Atlantic Ocean
AK
NU
NL
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
IA
IL
IN
WV
VA
OR
ID
KY
United States
TN
SC
NV
UT
OK
MS
AL
GA
Puerto Rico
MA
CA
FL
Guyana
AZ
NM
TX
LA
PA
HI
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

10^7 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

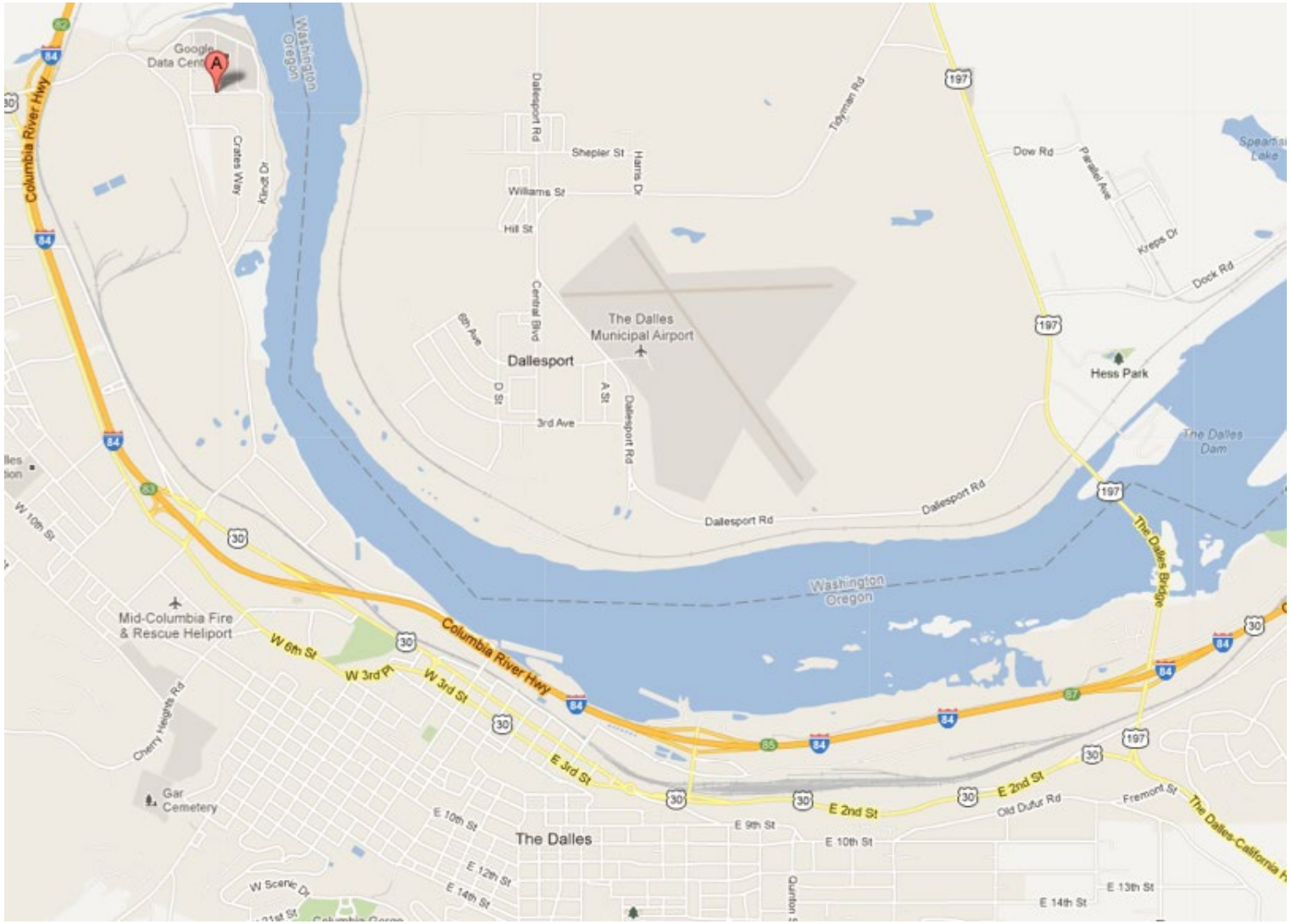
10^6 meters


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

10^5 meters

The Dalles, Oregon

10^4 meters

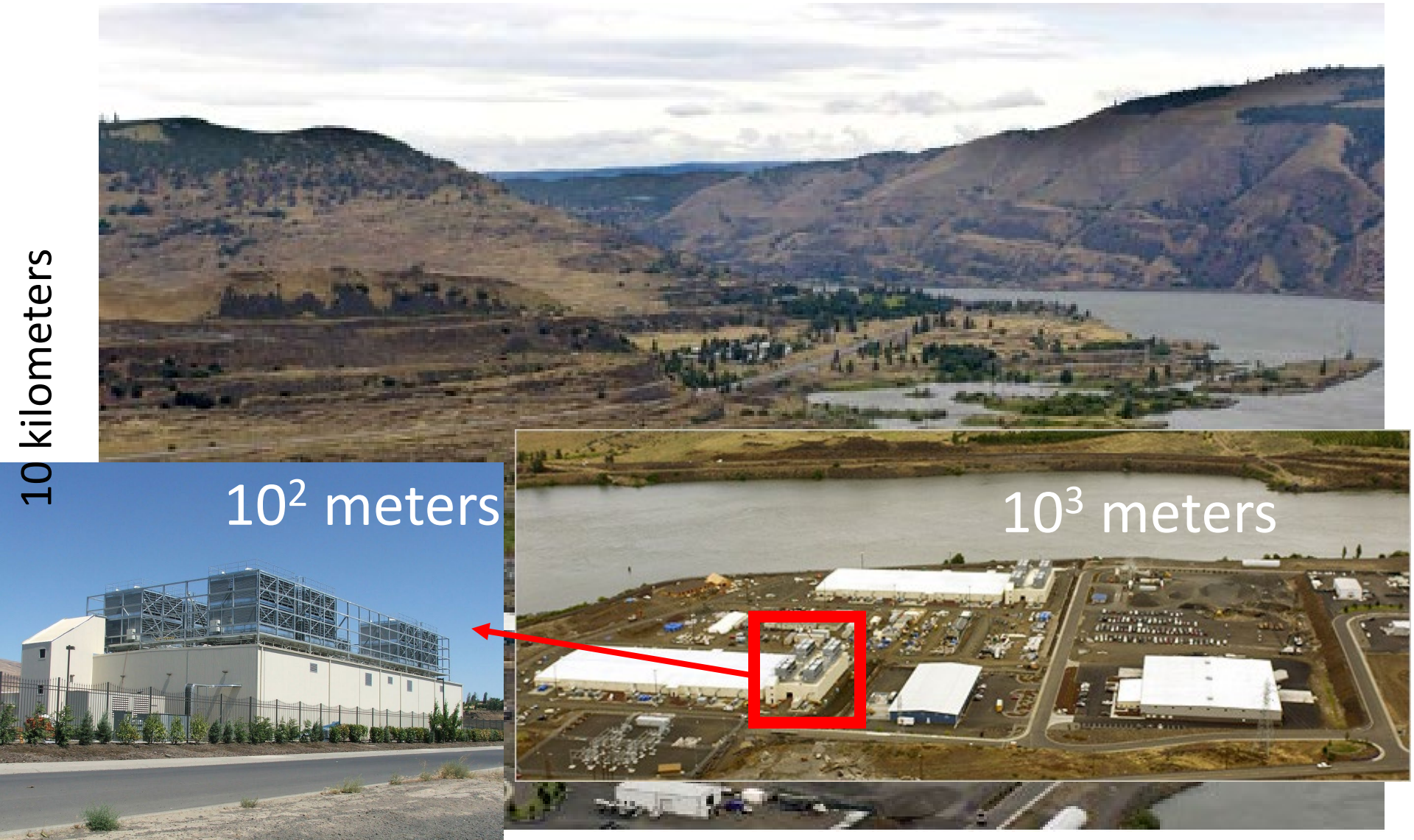


The Dalles, Oregon

10^4 meters

Google's Oregon WSC 10^3 meters

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

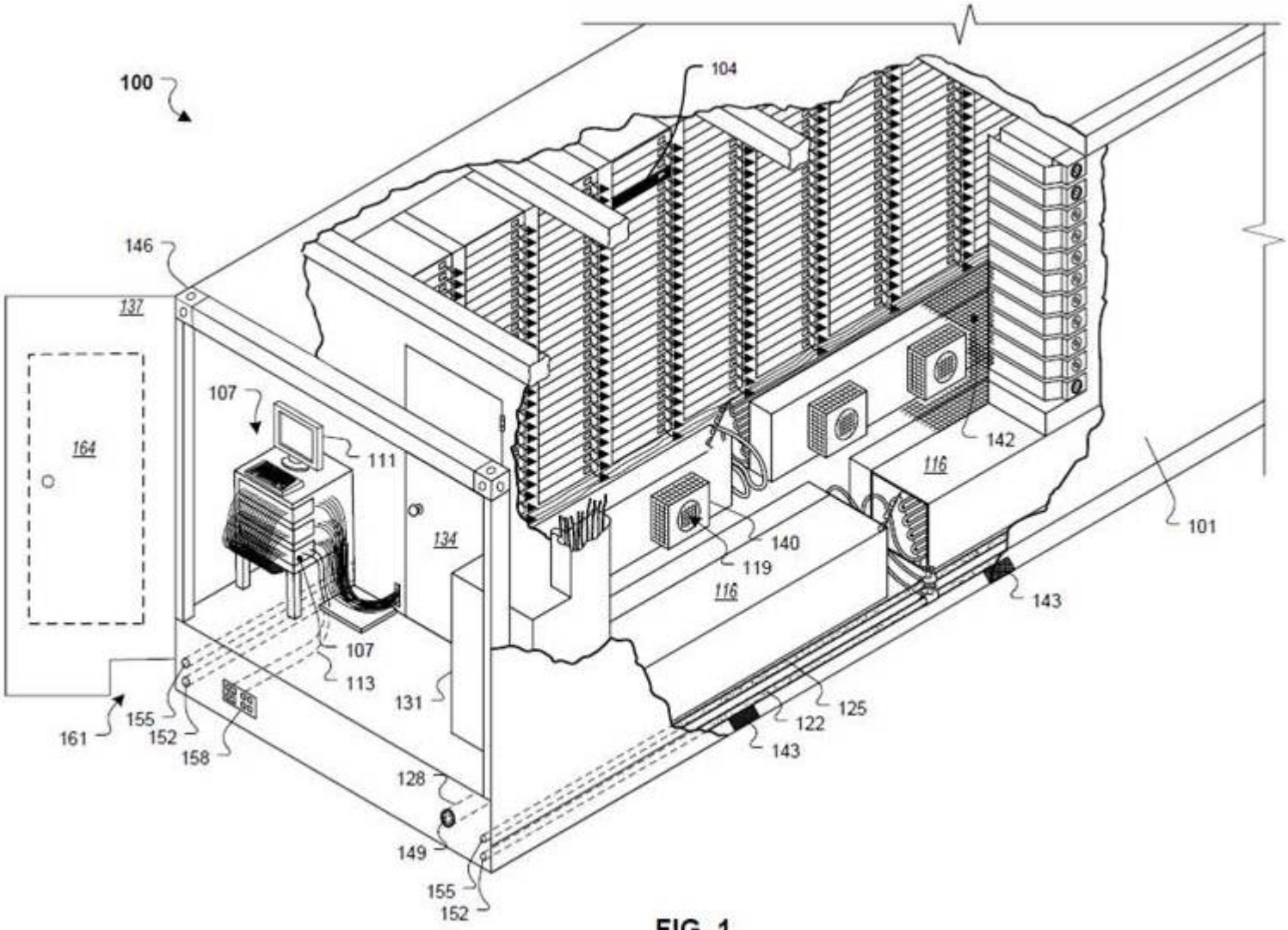
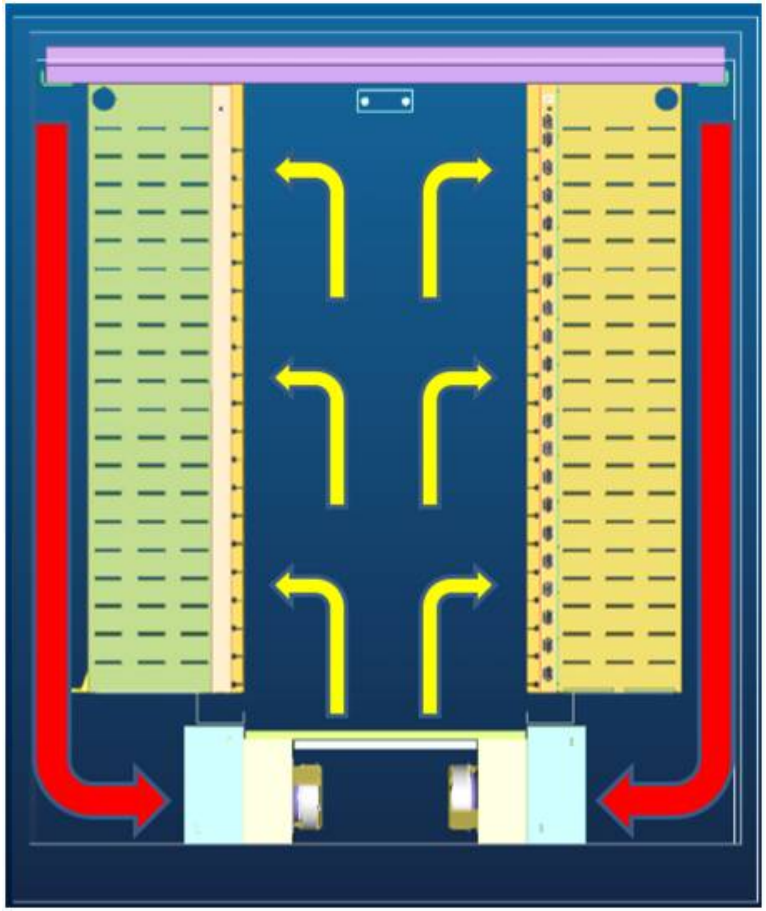


FIG. 1

Google Container

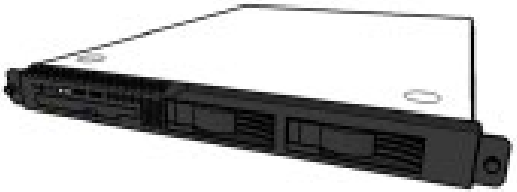
10^0 meters



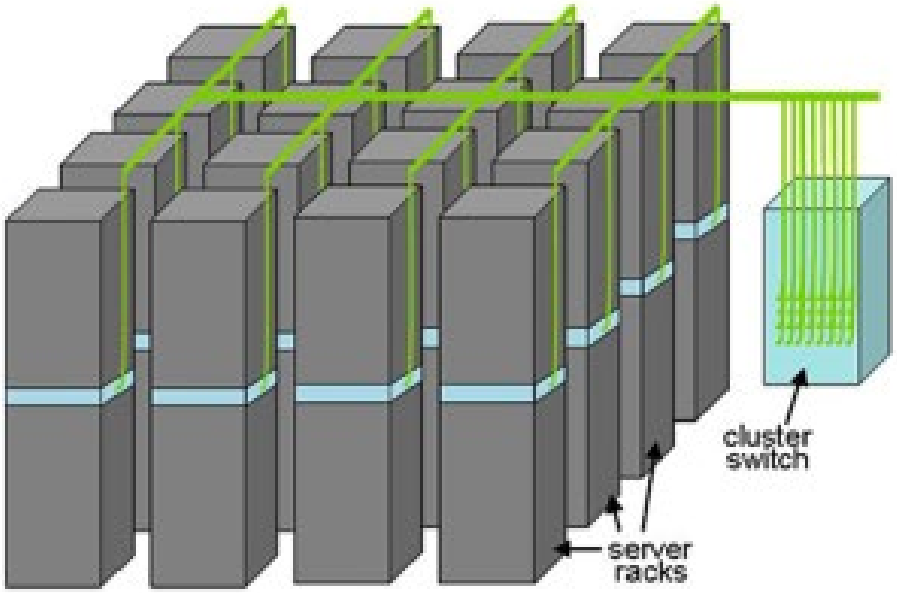
10 meters

- 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")

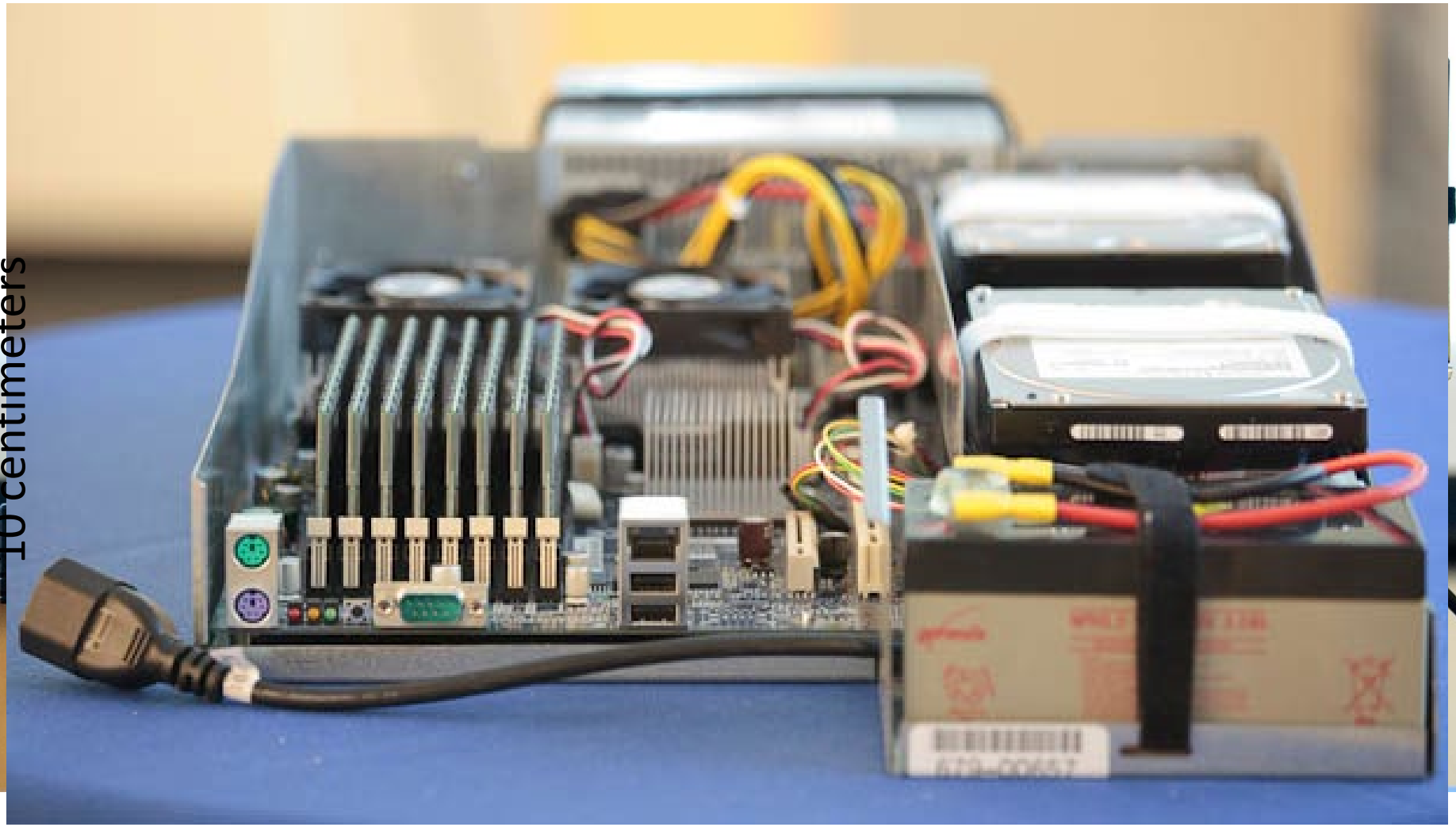
Array (aka cluster): server racks + larger local area network switch ("array switch") 10X faster => cost 100X: cost $f(N^2)$

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^{0} centimeters

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
- 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

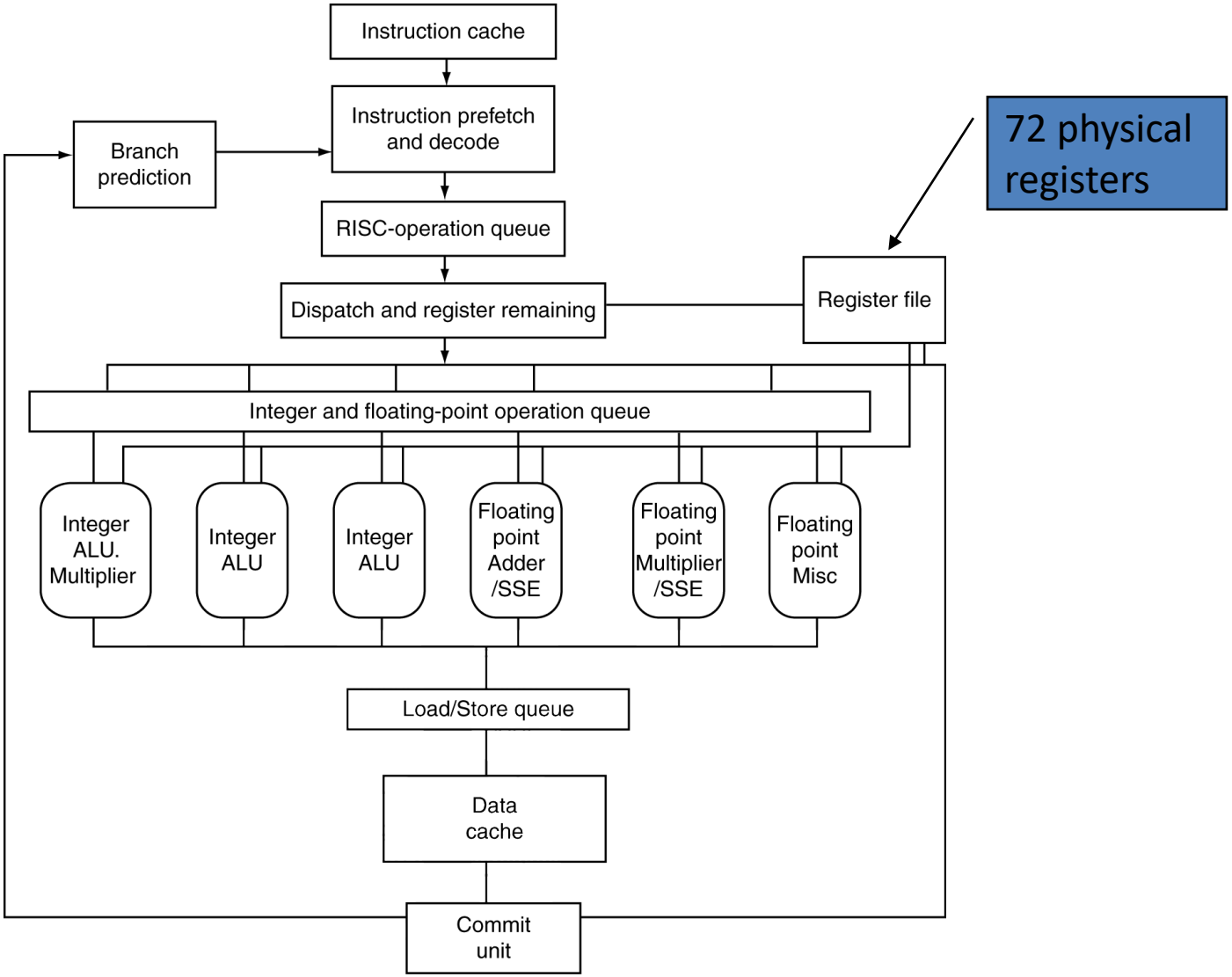
1. *Design for Moore's Law*
   -- More transistors = Multicore + SIMD
2. Abstraction to Simplify Design
3. Make the Common Case Fast
4. Dependability via Redundancy
5. *Memory Hierarchy*
   -- More transistors = Cache Memories
6. *Performance via Parallelism/Pipelining/ Prediction*
   -- Thread-level Parallelism

10^{-2} meters

AMD Opteron Microprocessor

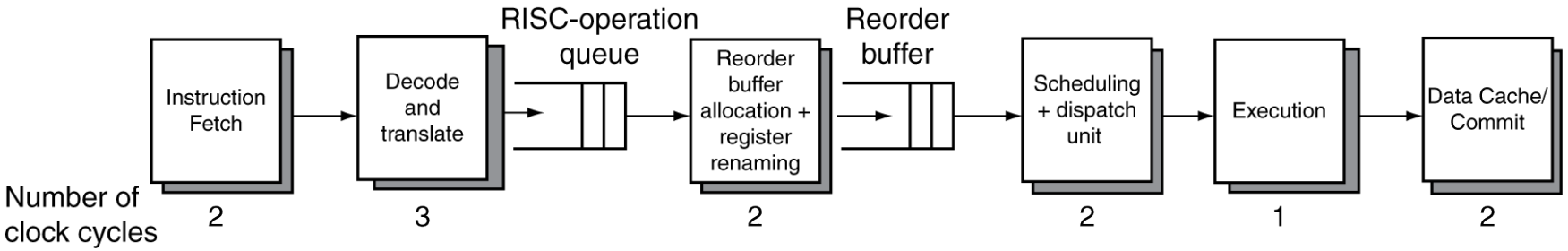
centimeters

AMD Opteron Microarchitecture



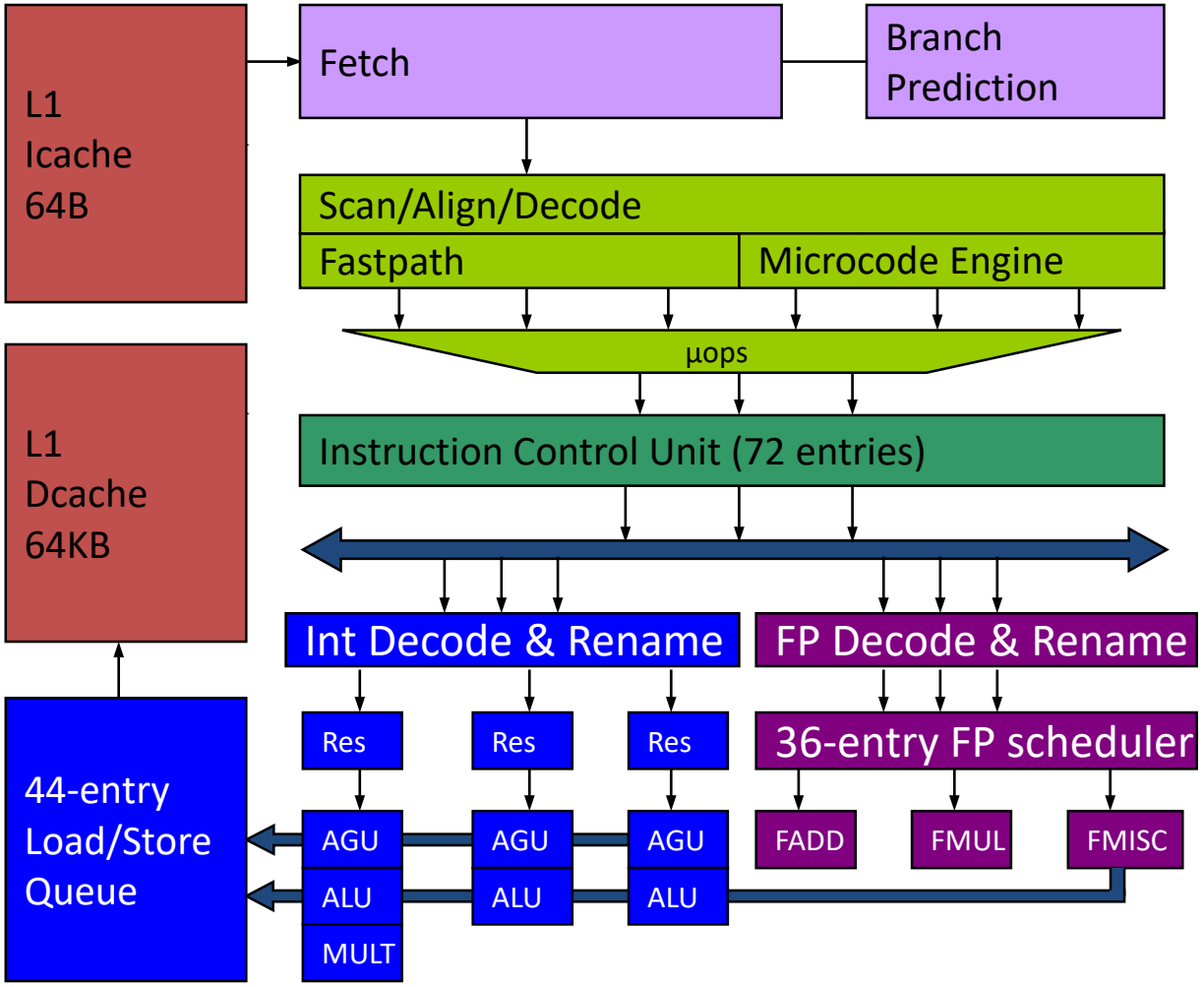
AMD Opteron Pipeline Flow

- For integer operations



- 12 stages (Floating Point is 17 stages)
- Up to 106 RISC-ops in progress

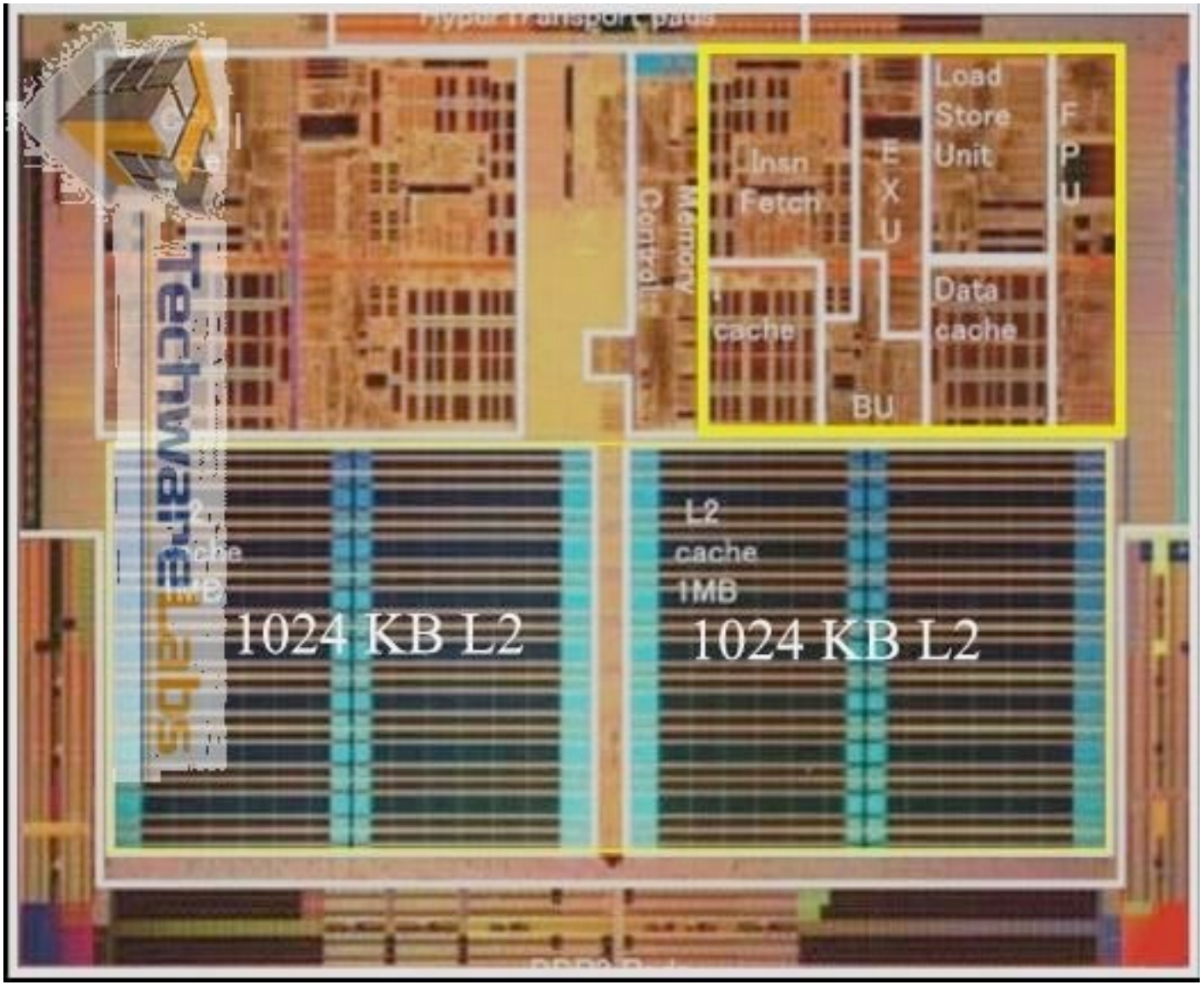
AMD Opteron Block Diagram



10^{-2} meters

AMD Opteron Microprocessor

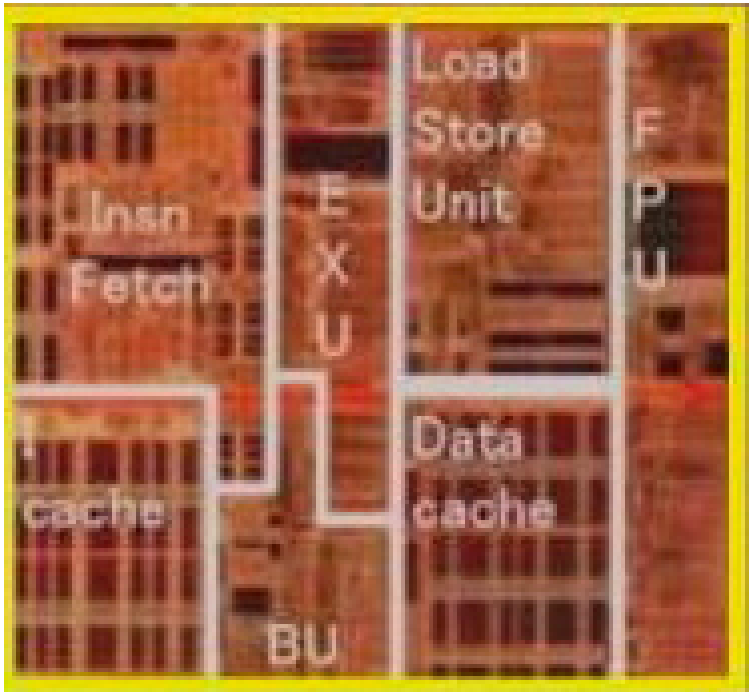
centimeters



10^{-3} meters

AMD Opteron Core

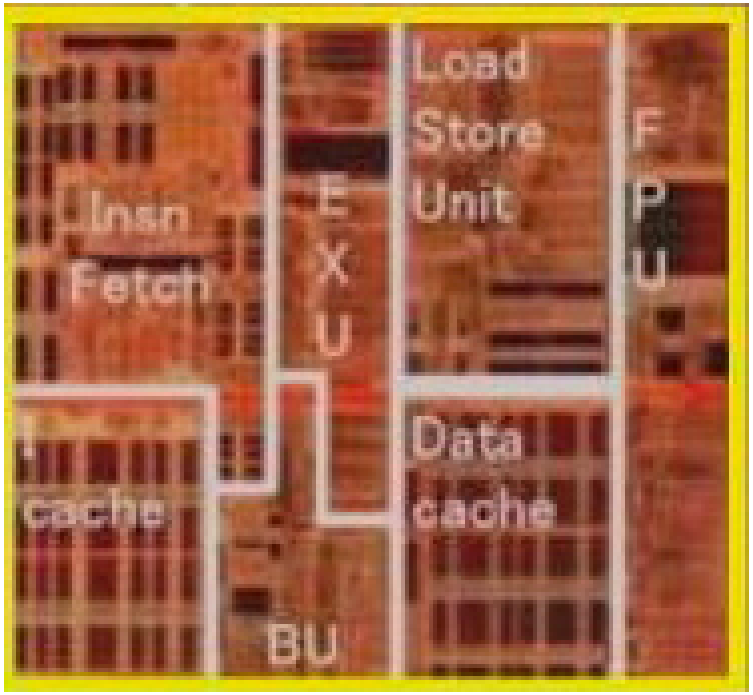
millimeters



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);
    }
}
```

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*

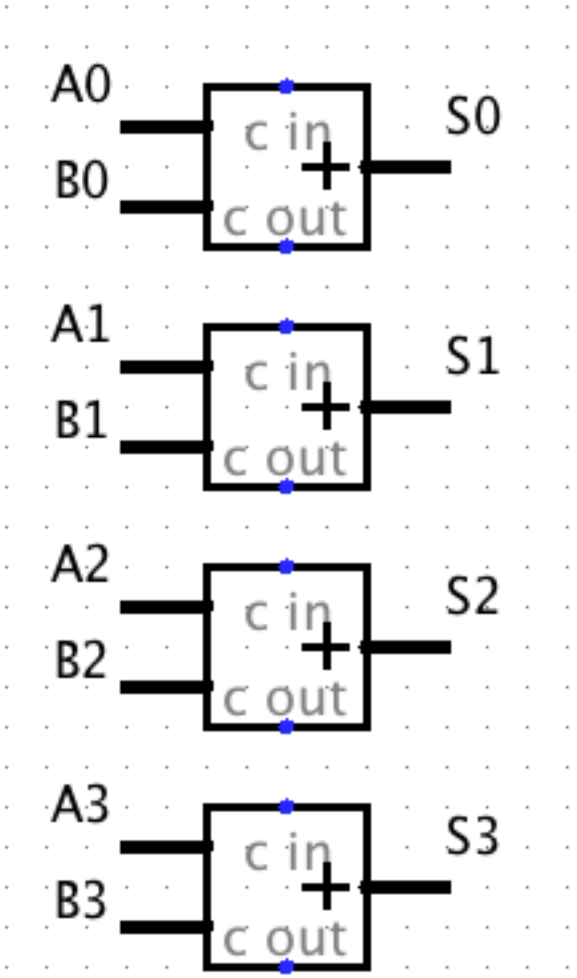
   -- Instruction Set Architecture, Micro-operations
3. Make the Common Case Fast
4. Dependability via Redundancy
5. Memory Hierarchy
6. *Performance via Parallelism/Pipelining/Prediction*

   -- Instruction-level Parallelism (superscalar, pipelining)

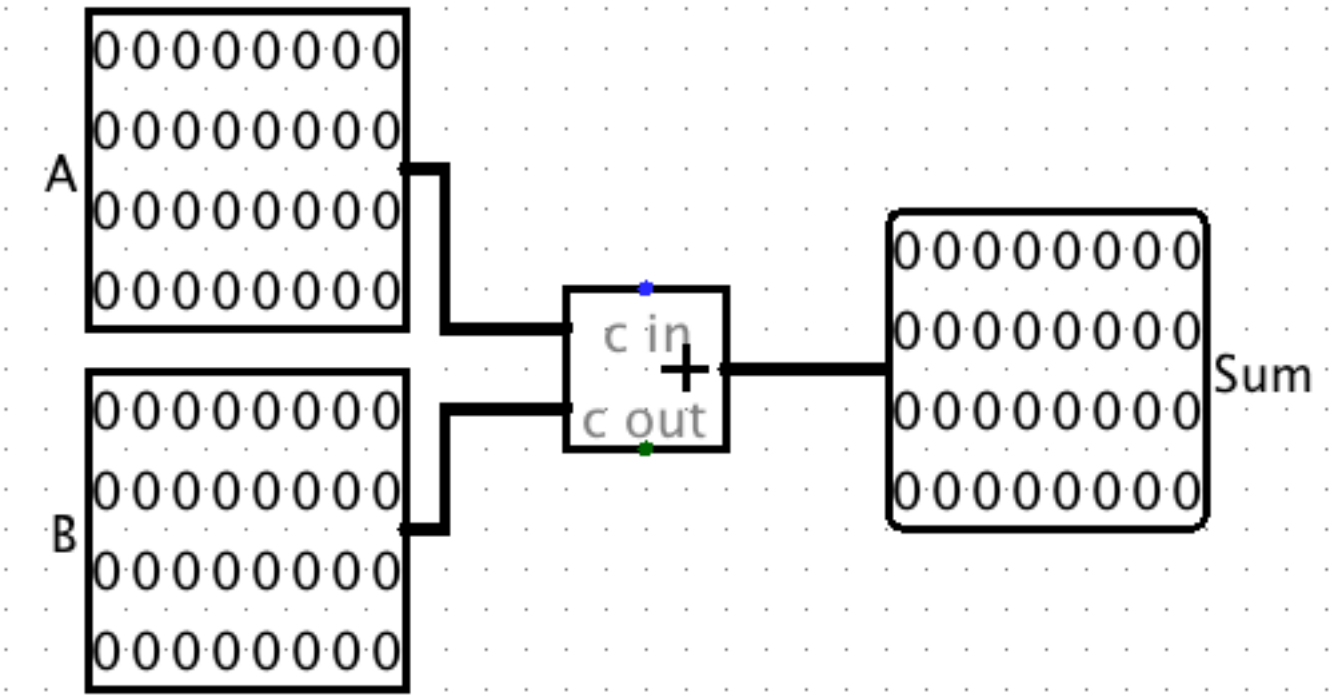
   -- Data-level Parallelism

SIMD Adder

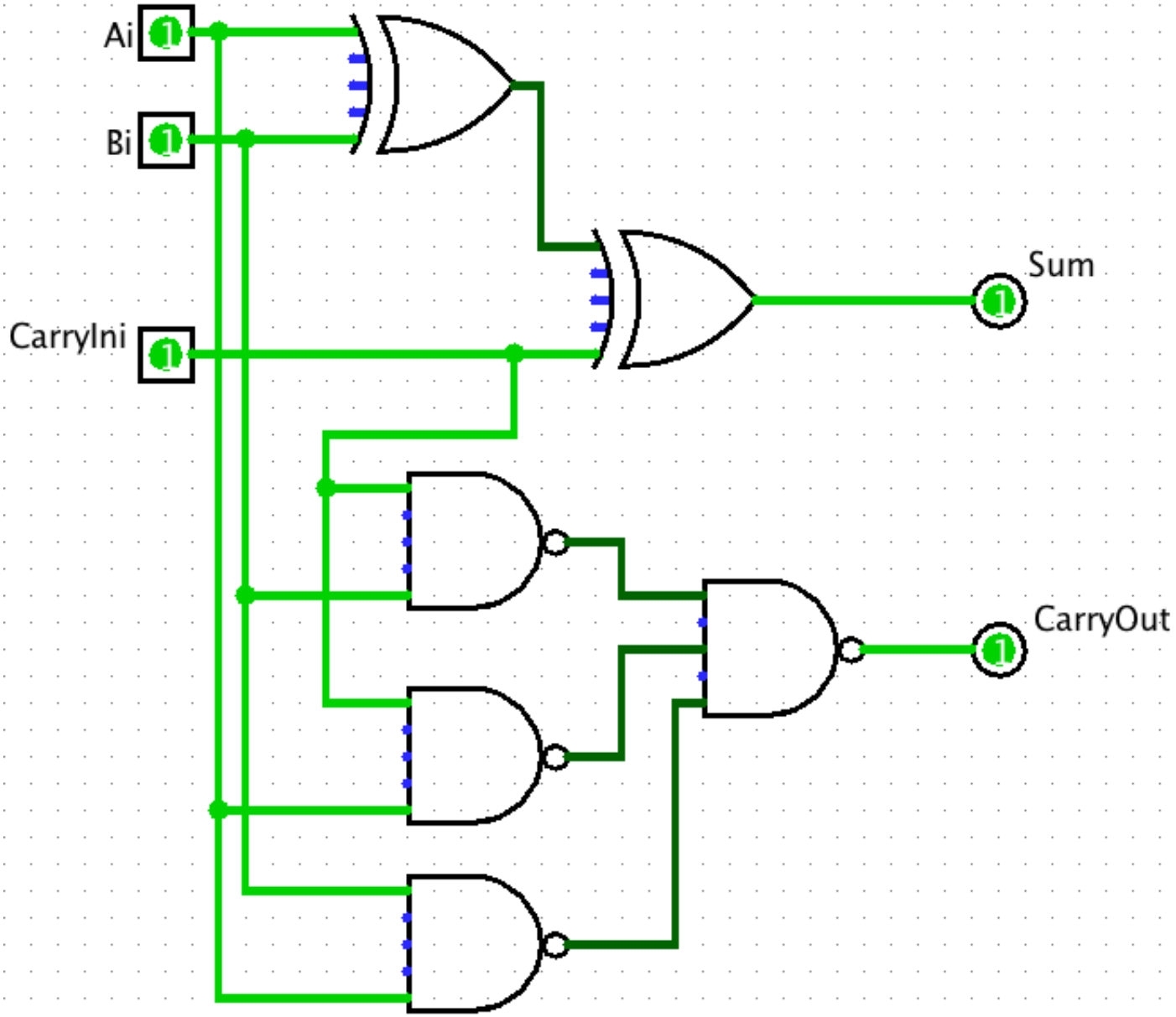
- Four 32-bit adders that operate in parallel
  - Data Level Parallelism



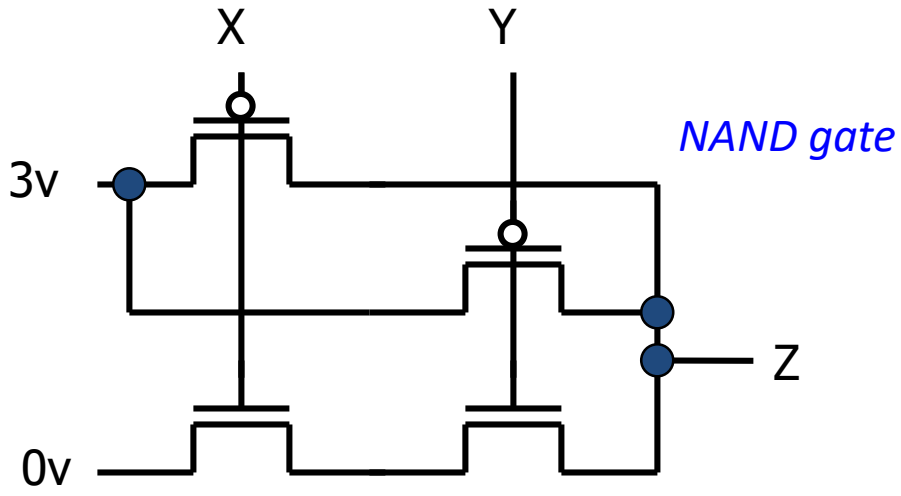
One 32-bit Adder



1 bit of 32-bit Adder



Complementary MOS Transistors (NMOS and PMOS) of NAND Gate



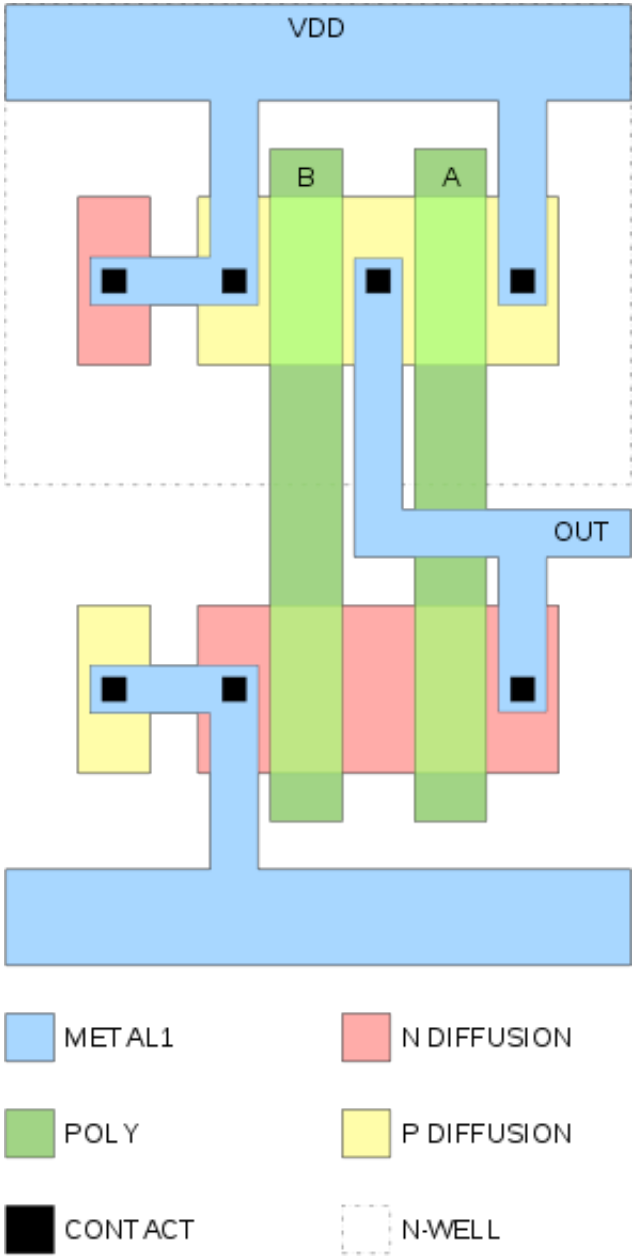
NAND gate

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

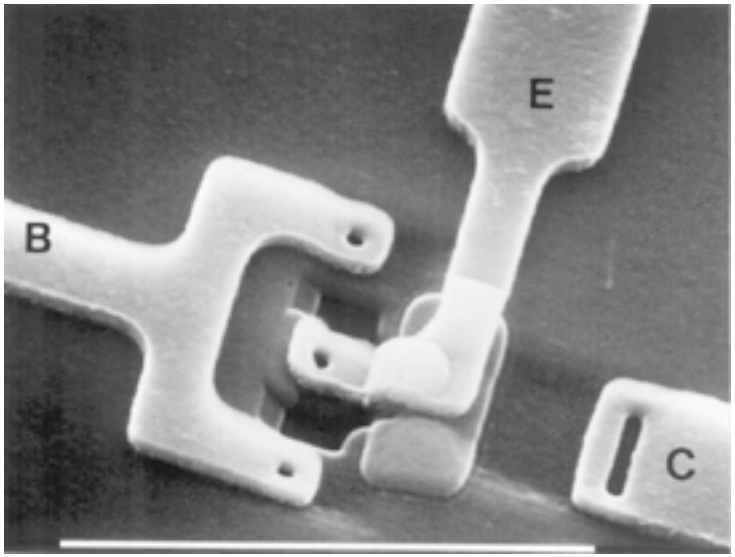
100 nanometers



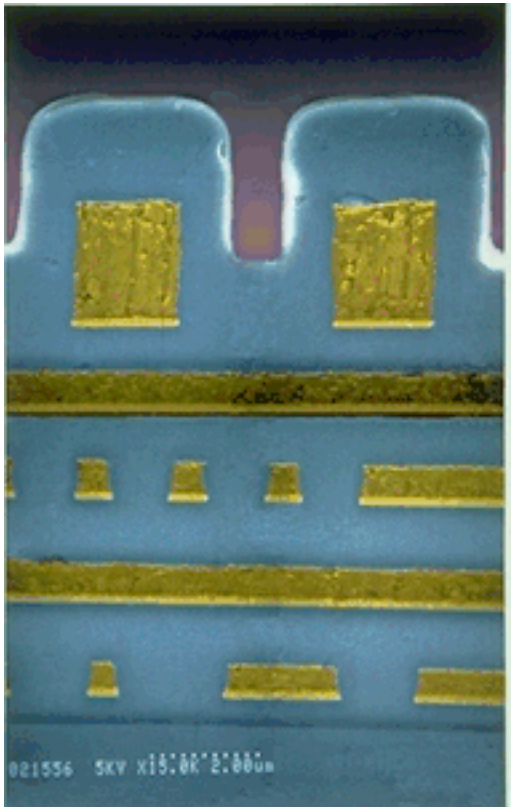
10^{-7} meters

Scanning Electron Microscope

100 nanometers



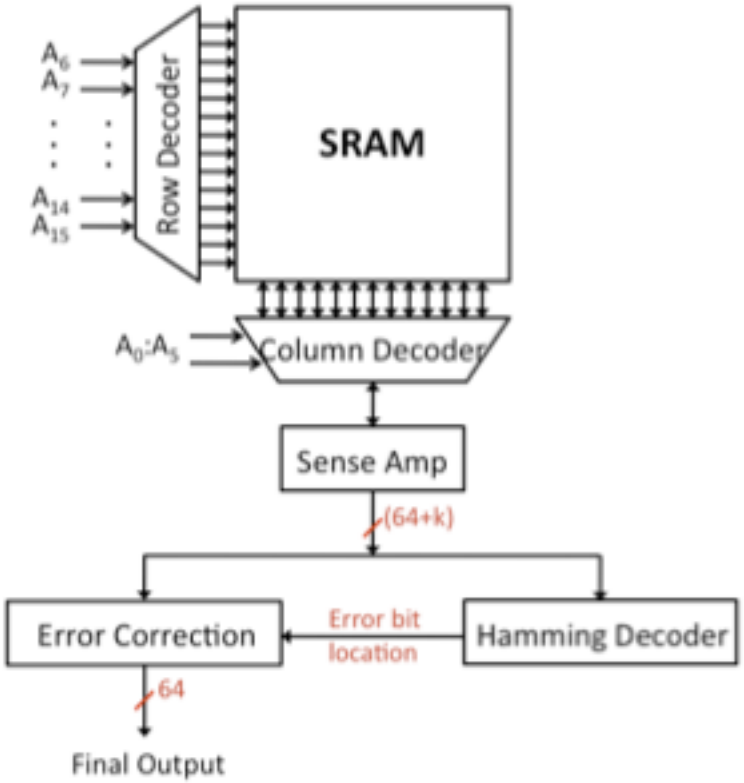
Top View



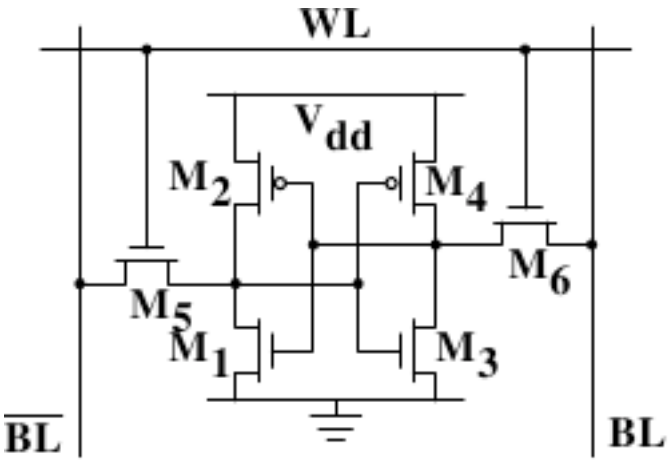
Cross Section

10^{-6} meters

Block Diagram of Static RAM



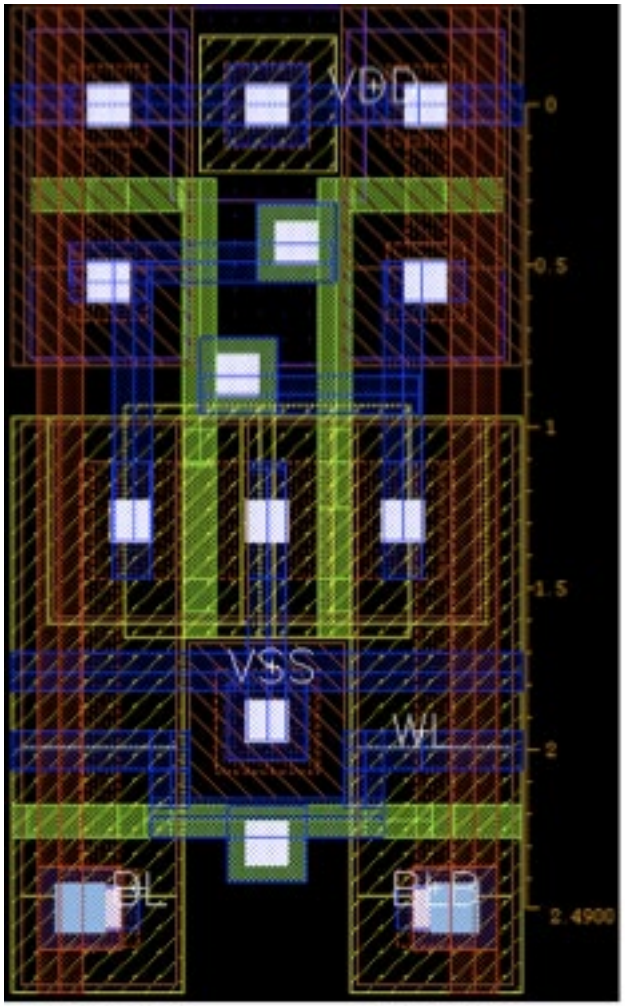
1 Bit SRAM in 6 Transistors



10^{-7} meters

Physical Layout of SRAM Bit

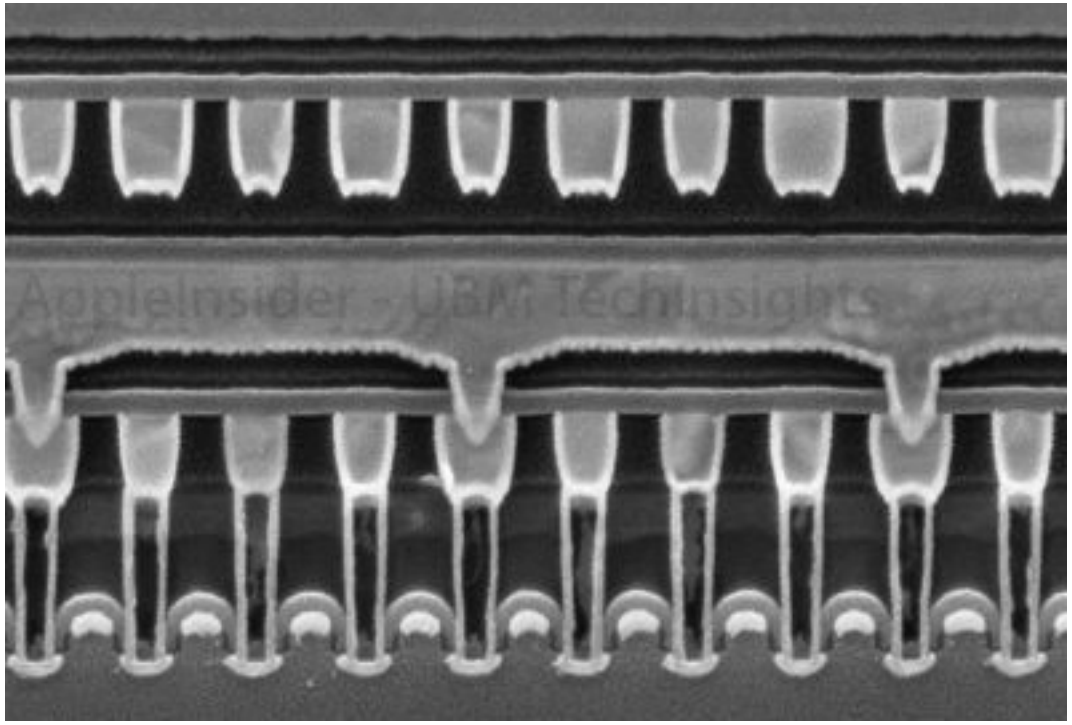
100 nanometers



10^{-7} meters

SRAM Cross Section

100 nanometers



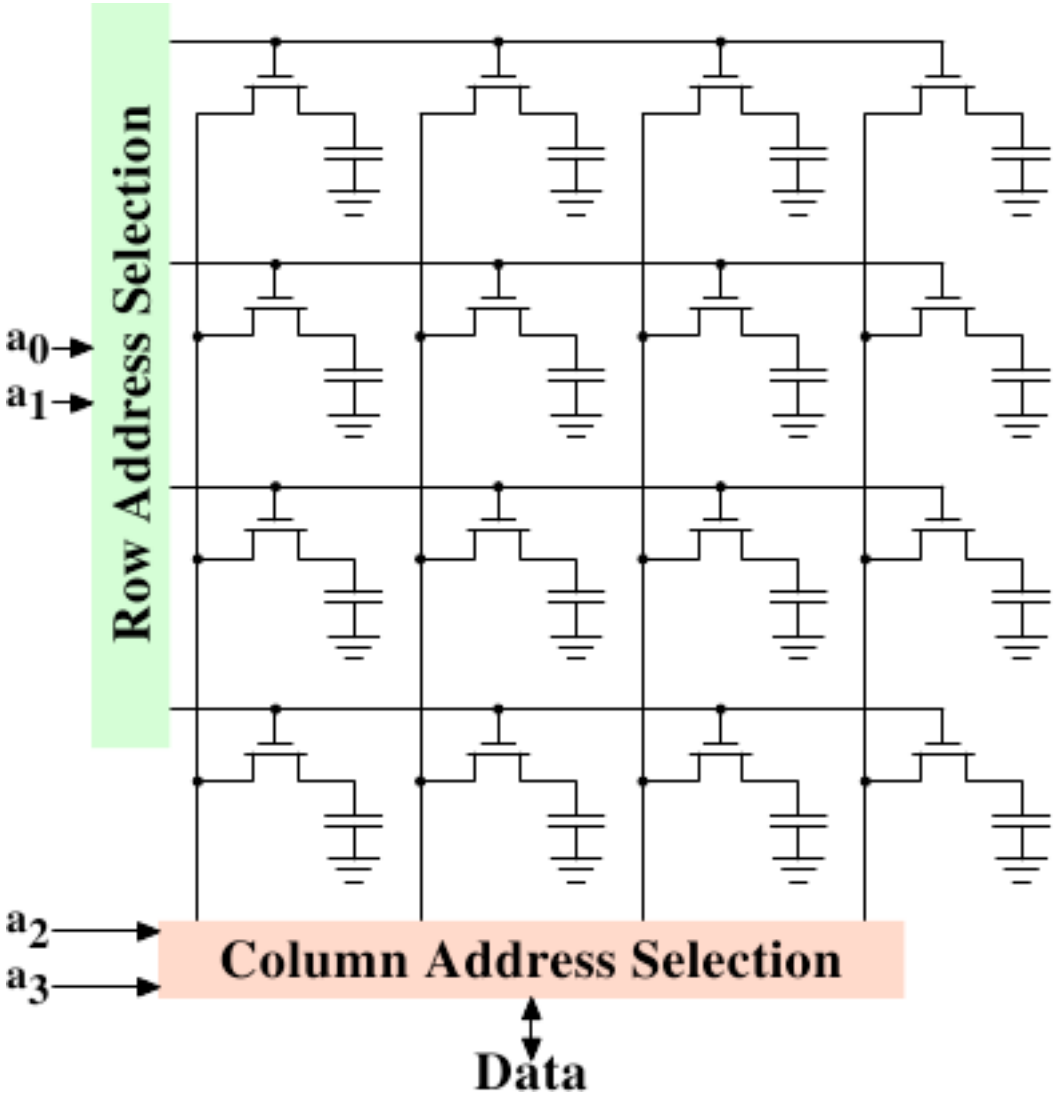
DIMM Module

- 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

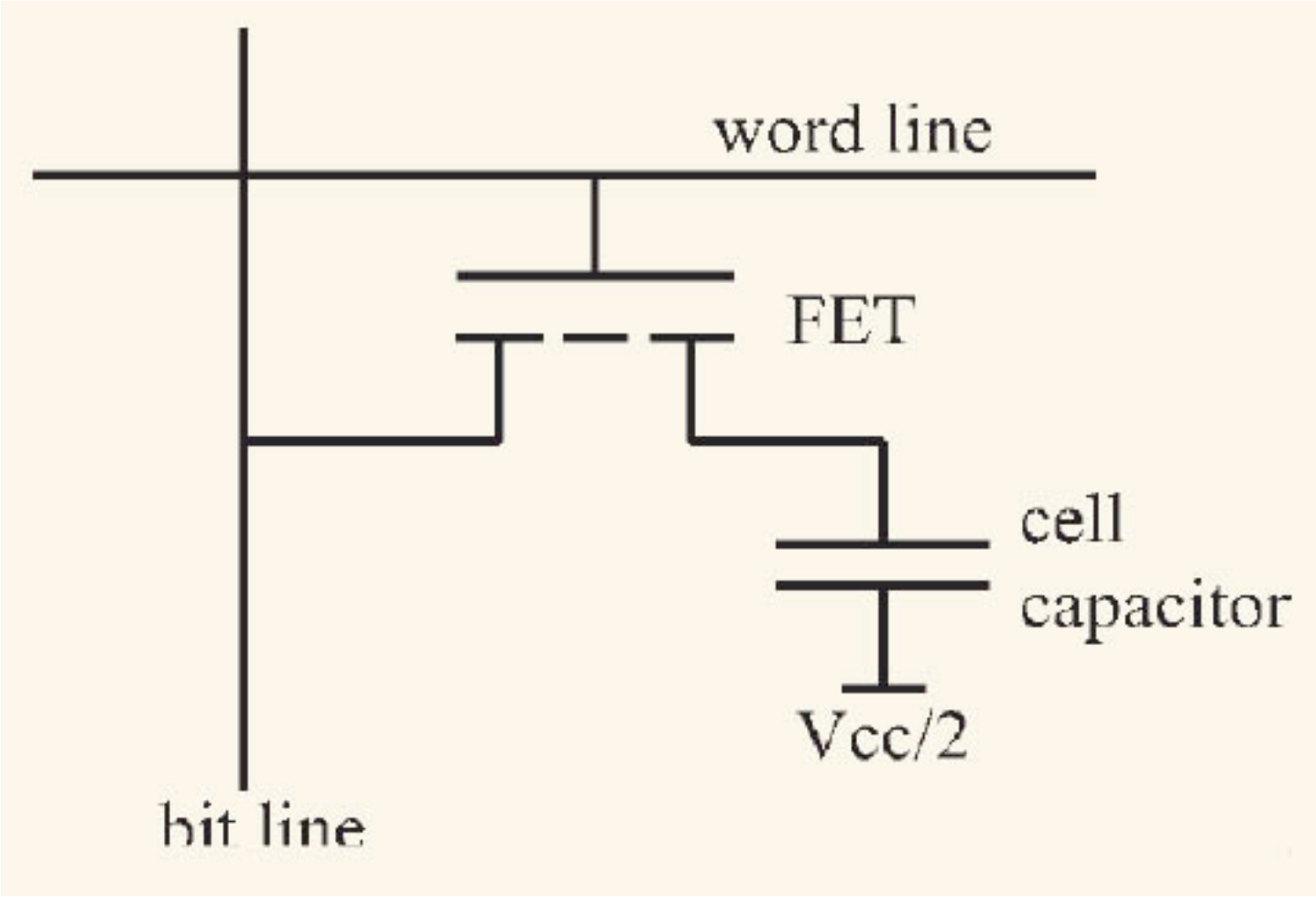
10^{-6} meters

DRAM Bits

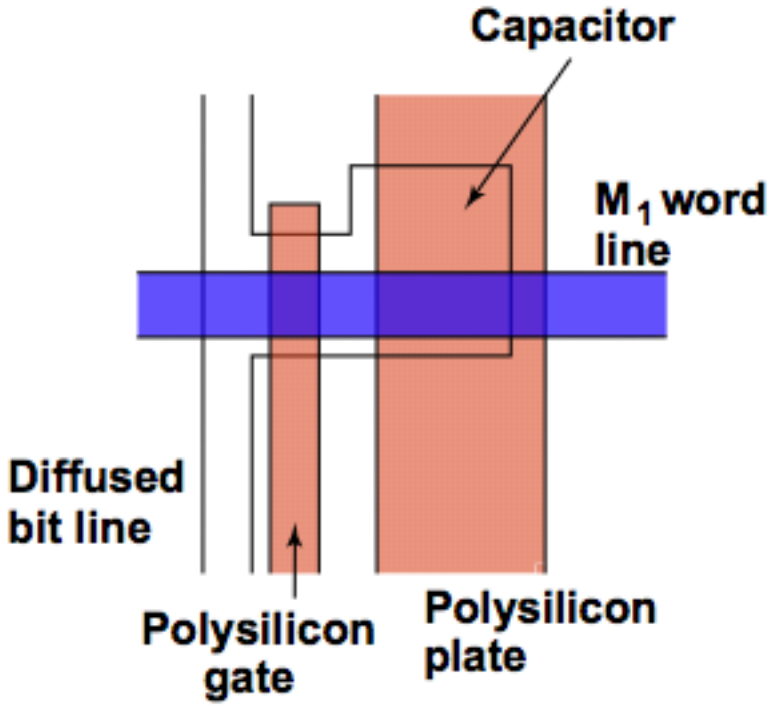
1 micron



DRAM Cell in Transistors



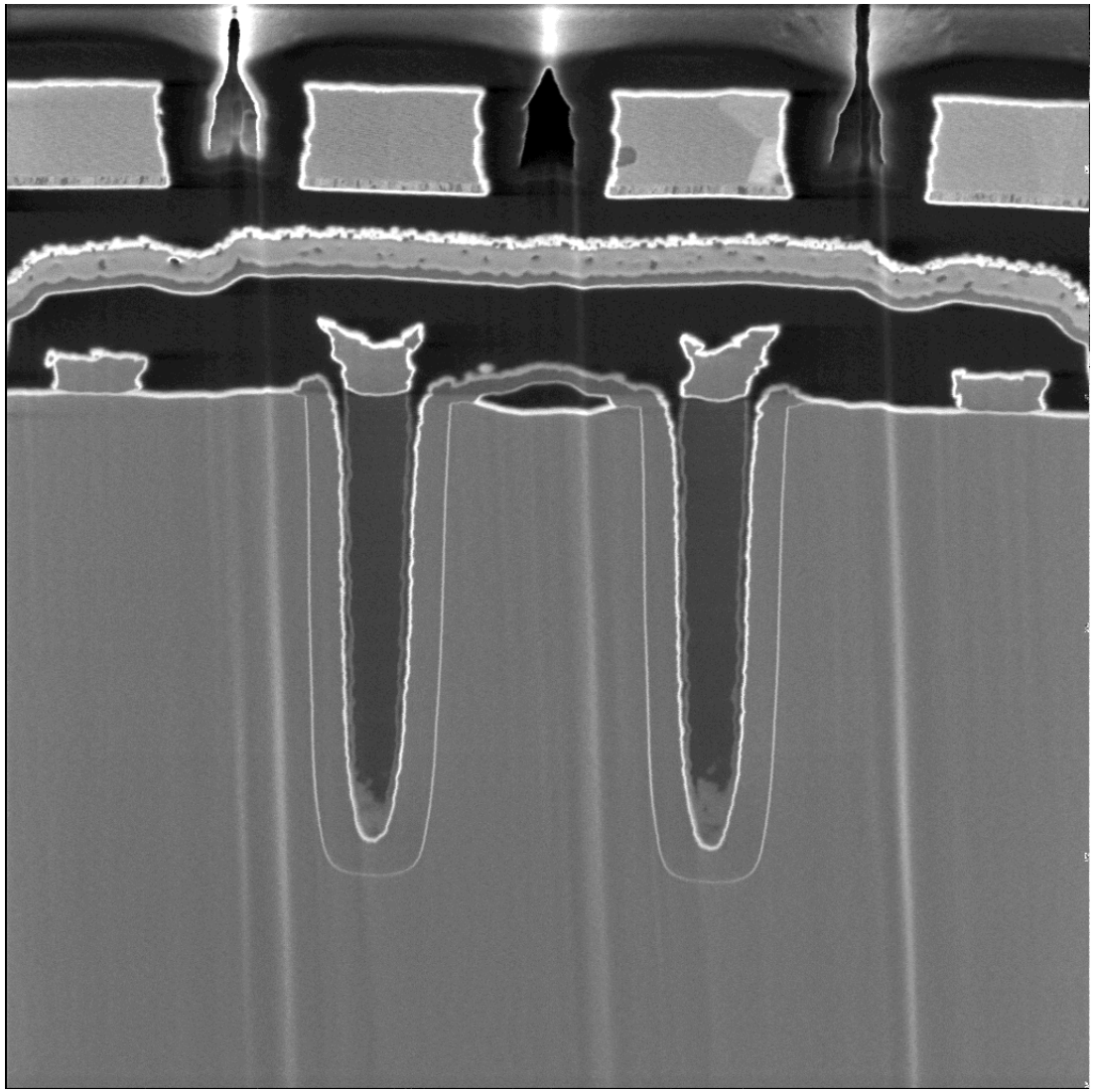
Physical Layout of DRAM Bit



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