

CSC2/452 Computer Organization Systems of Tomorrow

Sreepathi Pai

URCS

Nov 25, 2019

Outline

Administrivia

Introduction

CPUs of Tomorrow

Memory and Storage of Tomorrow

Systems of Tomorrow

Operating Systems of Tomorrow

x of tomorrow

Outline

Administrivia

Introduction

CPUs of Tomorrow

Memory and Storage of Tomorrow

Systems of Tomorrow

Operating Systems of Tomorrow

x of tomorrow

Administrivia

- ▶ Assignment #4 is due tomorrow 7PM
- ▶ Assignment #5 (last!) will be out Dec 2
 - ▶ Due on Dec 11

Outline

Administrivia

Introduction

CPUs of Tomorrow

Memory and Storage of Tomorrow

Systems of Tomorrow

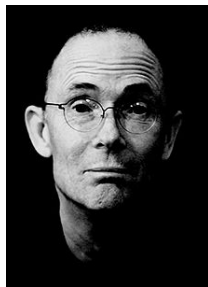
Operating Systems of Tomorrow

x of tomorrow

Tomorrow

The future is already here – it's just not very evenly distributed.

- ▶ William Gibson



1945: First Draft of the EDVAC

- ▶ Not very long ago
- ▶ Birth of the “CPU + RAM” design
- ▶ “Organization”

Dennard Scaling: 1974

TABLE I
SCALING RESULTS FOR CIRCUIT PERFORMANCE

Device or Circuit Parameter	Scaling Factor
Device dimension t_{ox}, L, W	$1/\kappa$
Doping concentration N_a	κ
Voltage V	$1/\kappa$
Current I	$1/\kappa$
Capacitance $\epsilon A/t$	$1/\kappa$
Delay time/circuit VC/I	$1/\kappa$
Power dissipation/circuit VI	$1/\kappa^2$
Power density VI/A	1

Dennard, et al., 1974, Design of Ion-implanted MOSFET's with Very Small Physical Dimensions, IEEE J. of Solid-State Circuits

The Bounty from Scaling

- ▶ More transistors, Same Area, Same Power
- ▶ Same number of transistors, Less Area, Less Power
 - ▶ Half the size, could run at twice the frequency
- ▶ Hence popular notion: speed of computers doubles every 18 months
- ▶ Enabled everything from Pixar to mobile phones in your pocket

When it all broken down: The Noughties

- ▶ Around 2004 to 2006, Dennard Scaling broke down
 - ▶ Transistors still growing smaller, but power usage not decreasing
 - ▶ More transistors, more power
 - ▶ More power in smaller area, heat dissipation problems
- ▶ Moore Scaling still applies
 - ▶ But pace has slowed down significantly
 - ▶ Lower limit is atomic size
- ▶ Modern chips have billions of transistors
 - ▶ Possibly will get to tens of billions

Dark Silicon?

- ▶ Even if we can build chips with hundreds of billions of transistors, can we switch them all on?
- ▶ Dark Silicon is the idea that we will be unable to power all of the chip simultaneously
 - ▶ Some parts of the chip will be without power, i.e. “dark”
 - ▶ These parts will only be switched on demand

Outline

Administrivia

Introduction

CPUs of Tomorrow

Memory and Storage of Tomorrow

Systems of Tomorrow

Operating Systems of Tomorrow

x of tomorrow

Multicore CPUs

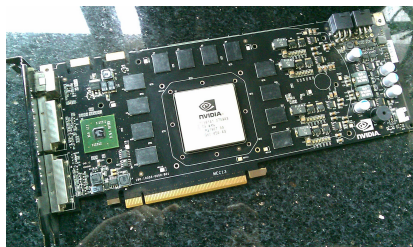
- ▶ Performance cannot be improved by reducing t (average time per unit of work)
- ▶ Increase P , parallelism!
- ▶ Thus, in 2004, Intel announces multicore CPUs
- ▶ But these are not transparent to programmers
 - ▶ You must rewrite software to take advantage of them!

Accelerator Computing

- ▶ If you have to rewrite software, are there alternatives to CPUs?
- ▶ Lots of accelerators proposed around 2004–2006
 - ▶ Cell BE (Sony PS3)
 - ▶ Clearspeed
 - ▶ Graphics Processing Units (GPUs)
- ▶ An accelerator is a processor that can speed up some tasks
 - ▶ CPU offloads these tasks to the accelerator

Graphics Processing Units: GPUs

- ▶ Originally designed to speed up graphics
 - ▶ Also games
- ▶ Could only do limited computation
- ▶ This changed around 2006
 - ▶ Could actually run C-like programs



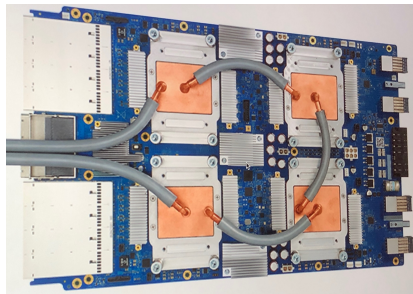
GPUs enable HPC+Deep Learning+Bitcoin

- ▶ 2006 to 2011, GPUs enabled high performance computing
 - ▶ Mostly matrix multiply
 - ▶ Nearly all supercomputers had GPUs by then
- ▶ In 2011, GPUs were also found to be really fast at training “deep” neural networks
 - ▶ 100x than CPUs
 - ▶ (Re-)started the AI hype
- ▶ A few years later, GPUs also enabled Bitcoin/Ethereum, etc.

Domain-specific accelerators: When GPUs are too slow

- ▶ GPUs and CPUs are general-purpose
 - ▶ Can perform most computations
- ▶ Higher performance can be obtained by specializing
- ▶ Google's Tensor Processing Unit is an example
 - ▶ Specialized for tensor operations

By Zinskauf - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=77299254>



Accelerators Galore

- ▶ Apple's A8 Mobile System-on-Chip
- ▶ CPU, GPU, Caches and Memory Controllers highlighted.
- ▶ What about the rest of the chip?

Hill and Reddi, Accelerator-level Parallelism: Mobile SoCs as Harbinger of the Future, ISPASS 2019



Programming Processing Units

- ▶ Fixed-function accelerators
 - ▶ Custom interfaces to each
 - ▶ Use libraries
- ▶ General-purpose accelerators
 - ▶ New programming languages
 - ▶ GPUs: CUDA, OpenCL, ...
- ▶ Multicore CPUs
 - ▶ Parallel Programming Models (PThreads, OpenMP, etc.)
- ▶ Heterogeneity
 - ▶ Which pieces of your program should run where?

Outline

Administrivia

Introduction

CPUs of Tomorrow

Memory and Storage of Tomorrow

Systems of Tomorrow

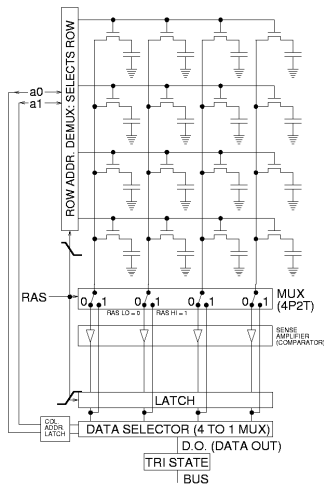
Operating Systems of Tomorrow

x of tomorrow

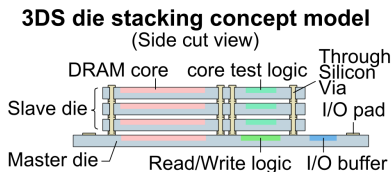
DRAM

- ▶ Current form of DRAM was invented in 1968
- ▶ Modern computers use DDR SDRAM
 - ▶ Introduced in 1992
- ▶ Slow, much focus on improving memory bandwidth
 - ▶ GPUs usually have around 300GByte/s
 - ▶ Usually 256 bits/cycle

Figure by Glogger at English Wikipedia - Transferred from en.wikipedia to Commons., CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=5549293>



Stacked DRAM: HBM and HMC



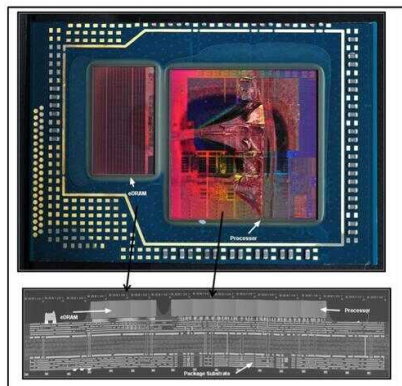
- ▶ Stack layers of DRAM in 3 dimensions
- ▶ High-bandwidth memory (HBM)
 - ▶ supports up to 4096 bits/cycle
- ▶ Hybrid Memory Cube
 - ▶ abandoned now
- ▶ Supported in some GPUs
 - ▶ upto 1TByte/s
 - ▶ A DVD is around 4–8GB

Figure by Shigeru23, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=17303708>

embedded DRAM

- ▶ Move DRAM onto the main processor package
- ▶ Can use it for building caches instead of SRAM
 - ▶ Enables much larger caches
- ▶ Featured here: Intel Haswell GT3e

Arabinda Das,
<https://www.eetimes.com/author.asp?doc;d=1323410>



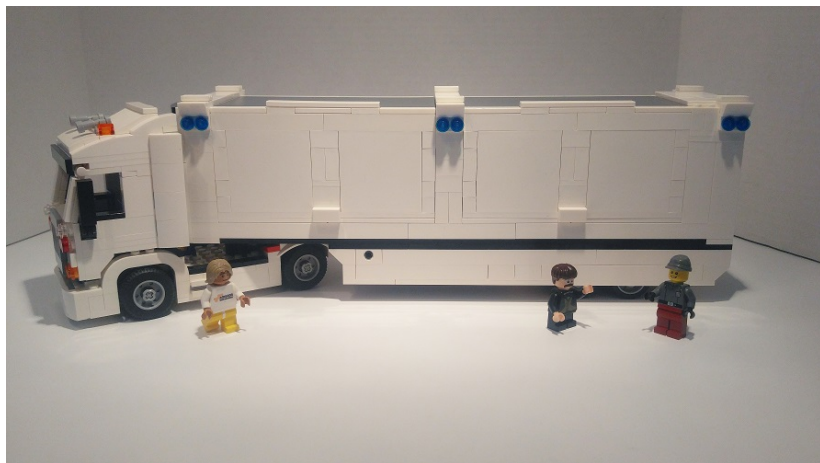
Storage: Non-volatile Memory

- ▶ Rotating Media
 - ▶ Magnetic Hard Disks (Shingled – 10TB+)
 - ▶ Tape Systems (10TB+)
- ▶ Still very important, even when most data is stored remotely
 - ▶ Amazon Snowball Edge – 100TB (featured here)
 - ▶ Used to transfer data to Amazon's servers

Source: <https://aws.amazon.com/blogs/aws/aws-snowball-edge-more-storage-local-endpoints-lambda-functi>



Amazon Snowmobile



- ▶ Can transfer 100PB/trip
- ▶ Who generates this much data?
 - ▶ Satellite imagery is apparently around 10PB/year

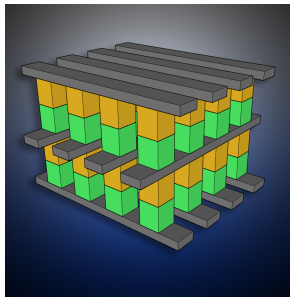
Source: <https://aws.amazon.com/blogs/aws/aws-snowmobile-move-exabytes-of-data-to-the-cloud-in-weeks/>

Flash Memory

- ▶ Solid-state storage
 - ▶ No moving parts
 - ▶ Limited endurance (number of writes)
- ▶ Most common
 - ▶ USB Flash
 - ▶ SSD

3D XPoint

- ▶ Developed by Intel/Micron
 - ▶ Stores data using “bulk properties” of materials
- ▶ Two primary uses:
 - ▶ SSD (Flash memory replacement)
 - ▶ NVDIMM (DRAM replacement)
- ▶ Legend
 - ▶ Ochre: Selector
 - ▶ Green: Memory cell
 - ▶ Gray: Power



Picture by Trolomite - Own work, CC BY-SA 4.0,
<https://commons.wikimedia.org/w/index.php?curid=45277623>

Non-volatile Main Memory?

- ▶ Intel Optane DC
 - ▶ Claimed to be 8X denser than DRAM (6TB per system)
 - ▶ Others on their way
- ▶ Data remains in memory when you:
 - ▶ Switch off
 - ▶ Program ends
- ▶ How will writing programs change?

Source: <https://www.intel.com/content/www/us/en/architecture-and-technology/optane-dc-persistent-memory.html>



Programming NVM

- ▶ The Good:
 - ▶ No more translating between in-memory and on-disk representations
- ▶ The Bad:
 - ▶ What if a program crashes?
 - ▶ How to detect inconsistent state?
- ▶ The Strange:
 - ▶ Writes much slower than Reads
 - ▶ Both slower than DRAM
- ▶ Heterogeneity in RAM
 - ▶ Which parts of data structures in DRAM? Which parts in NVRAM?

Outline

Administrivia

Introduction

CPUs of Tomorrow

Memory and Storage of Tomorrow

Systems of Tomorrow

Operating Systems of Tomorrow

x of tomorrow

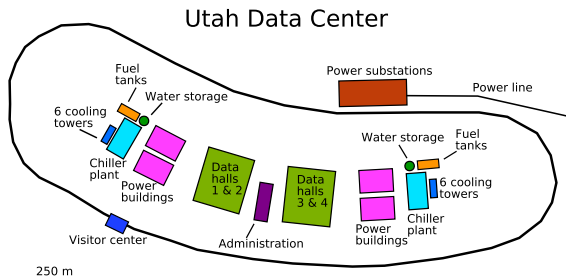
Mobile Computing

- ▶ Mobile phones obviously
- ▶ Virtual Reality
 - ▶ e.g. Head-mounted computers
- ▶ Augmented Reality
 - ▶ Glasses (mostly)
- ▶ Key challenges
 - ▶ Size, energy, power, performance



Hololens image by Ramadhanakbr - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=52546267>

Data Centre Computing



- ▶ Consolidate compute, data storage in one central, physical location
- ▶ Access data and applications remotely
- ▶ Key challenges:
 - ▶ Size, energy, power, performance

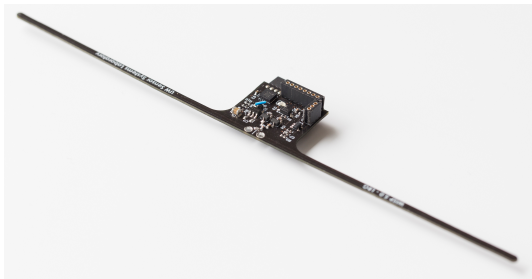
By A5b - Own work, CC0, <https://commons.wikimedia.org/w/index.php?curid=30380190>

Cloud Computing

- ▶ On-demand access to computing resources
- ▶ Arguably, made famous by Amazon
 - ▶ Amazon AWS
- ▶ Amazon builds and runs data centres
 - ▶ You rent portions of it
 - ▶ Pay per time/data transferred, etc. (like a utility)



Intermittent Computing



- ▶ Featured here, a device that runs off the energy of radio waves
 - ▶ Think Wifi-powered devices
- ▶ Such energy-harvesting devices compute “intermittently”
 - ▶ Power could be lost any moment

Lucia et al., "Intermittent Computing: Challenges and Opportunities", SNAPL 2017

Others Systems of the Future

Where will computing systems appear next?

Programming Tiny Computers in your Pocket and Computers in the Cloud

- ▶ Programming Languages and Tools?
- ▶ Program support?
 - ▶ Processes?
 - ▶ Virtual memory?
 - ▶ Parallelism?
 - ▶ Networking?

Outline

Administrivia

Introduction

CPUs of Tomorrow

Memory and Storage of Tomorrow

Systems of Tomorrow

Operating Systems of Tomorrow

x of tomorrow

Virtualization

- ▶ Cloud computing requires a machine to run multiple operating systems
 - ▶ Virtualize entire machine
 - ▶ Can stop a virtual machine and move it to another machine in the cloud
- ▶ OS for the cloud
 - ▶ Virtual machines (e.g. Qemu, VirtualBox, VMware)
 - ▶ Hypervisors (e.g. Xen)
 - ▶ “Ultravisors”



Containers

- ▶ Same OS, different “applications”
- ▶ Isolating applications from each other
 - ▶ Different “tenants” in the cloud
 - ▶ Different installed apps
- ▶ As lightweight as possible
 - ▶ “Serverless” computing



Challenges

- ▶ How to secure tenants?
 - ▶ Similar to protecting processes from each other
- ▶ How to account and bill for resources?
 - ▶ CPU time, bandwidth, storage, etc.?

Outline

Administrivia

Introduction

CPUs of Tomorrow

Memory and Storage of Tomorrow

Systems of Tomorrow

Operating Systems of Tomorrow

x of tomorrow

What about the rest of the system?

- ▶ Programming languages of tomorrow
 - ▶ Not C, possibly not Python, possibly not Java
 - ▶ Rust? Go?
 - ▶ WebAssembly? (not a direct programming language)
- ▶ Applications of tomorrow?
 - ▶ AI?

Reflections

- ▶ Machines will be increasingly heterogeneous
 - ▶ Both in processors and RAM
- ▶ Programming models will be very different
 - ▶ General-purpose
 - ▶ Domain-specific
 - ▶ Fixed-function
- ▶ Performance is now the responsibility of the programmer!
- ▶ Security is a major concern