CENG 3420
Computer Organization & Design

Lecture 01: Introduction

Bei Yu
CSE Department, CUHK
byu@cse.cuhk.edu.hk

(Textbook: Chapters 1.3 & 1.4)

2024 Spring

Course Information

Course Administration

Instructor:

- Bei Yu (byu@cse.cuhk.edu.hk)
- Office: SHB 907
- Office Hrs: H14:30–16:30

Tutors:

- Su Zheng (szheng22@cse.cuhk.edu.hk)
- Shixin Chen (sxchen22@cse.cuhk.edu.hk)
- Shuo Yin (syin22@cse.cuhk.edu.hk)
- Mingjun Li (mjli23@cse.cuhk.edu.hk)
- Lancheng Zou (lczou23@cse.cuhk.edu.hk)

Grading Information

Grade Determinates

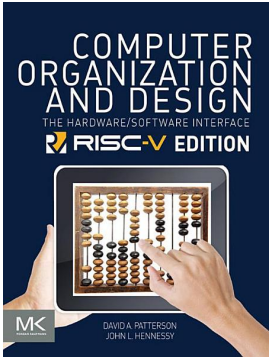
5%	Attendance
15%	Homework
15%	Midterm (Mar. 21)
30%	Three Labs (Individual project)
35%	Final Exam

- Late submission **per day** is subject to 10% of penalty.
- A student must gain at least 50% of the full marks in order to pass the course.
- A student must attend at least 80% of lectures in order to gain all class attendance credits.

General References

Textbook:

- *Computer Organization and Design*, RISC-V Edition
- Soft copy, `amazon.cn`, or `amazon.com`



Manuals:

- LC-3 Instruction Set Architecture (ISA)
- Lab tutorials (slides)

Slides:

- On the course web page before lecture
- Summary may be uploaded afterwards

Course Content

- Introduction to the major components of a computer system, how they function together in executing a program.
- Introduction to CPU datapath and control unit design
- Introduction to techniques to improve performance and energy-efficiency of computer systems
- Introduction to multiprocessor architecture

Course Content

- Introduction to the major components of a computer system, how they function together in executing a program.
- Introduction to CPU datapath and control unit design
- Introduction to techniques to improve performance and energy-efficiency of computer systems
- Introduction to multiprocessor architecture

Philosophy

To learn what determines the capabilities and performance of computer systems and to understand the interactions between the computer's architecture and its software so that future software designers (compiler writers, operating system designers, database programmers, application programmers, ...) can achieve the best cost-performance trade-offs and so that future architects understand the effects of their design choices on software.

Why Learn This Stuff?

- You want to call yourself a "computer scientist/engineer"
- You want to build HW/SW people use (so need performance/power)
- You need to make a purchasing decision or offer "expert" advice

Both hardware and software affect performance/power

- Algorithm determines number of source-level statements
- Language/compiler/architecture determine the number of machine-level instructions
- Processor/memory determine how fast and how power-hungry machine-level instructions are executed

What You Should Already Know

- Basic logic design & machine organization
  - logical minimization, FSMs, component design
  - processor, memory, I/O
- Create, run, debug programs in an assembly language
  - Will be introduced in tutorial
- Create, compile, and run C/C++ programs
- Create, organize, and edit files and run programs on Unix/Linux

What You Should Already Know

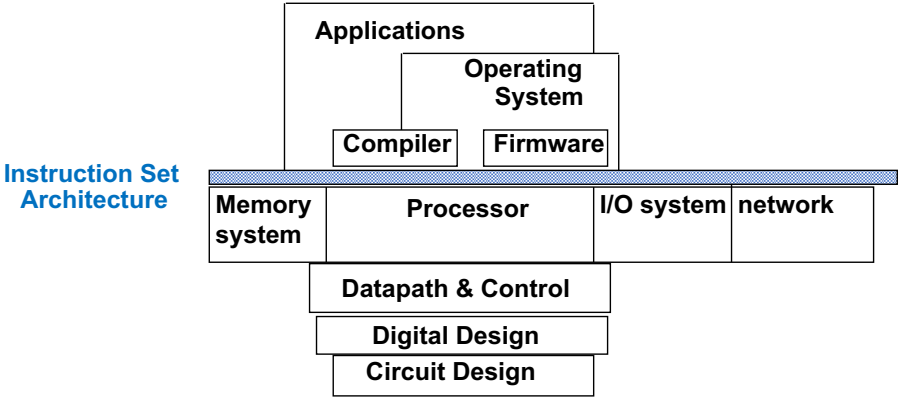
- Basic logic design & machine organization
  - logical minimization, FSMs, component design
  - processor, memory, I/O
- Create, run, debug programs in an assembly language
  - Will be introduced in tutorial
- Create, compile, and run C/C++ programs
- Create, organize, and edit files and run programs on Unix/Linux

One example here!

Computer Organization and Design

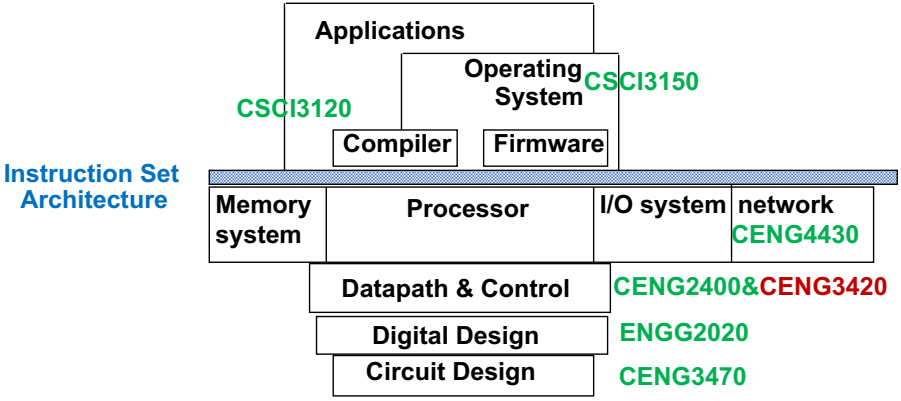
- This course is all about how computers work
- But what do we mean by a computer?
  - Different types: embedded, laptop, desktop, server
  - Different uses: automobiles, graphics, finance, genomics ...
  - Different manufacturers: Intel, Apple, IBM, Sony, Oracle ...
  - Different underlying technologies and different costs
- Analogy: Consider a course on "automotive vehicles"
  - Many similarities from vehicle to vehicle (e.g., wheels)
  - Huge differences from vehicle to vehicle (e.g., gas vs. electric)
- **Best way to learn**:
  - Focus on a specific instance and learn how it works
  - While learning general principles and historical perspectives

How Do the Pieces Fit Together?



- Coordination of many levels of abstraction
- Under a rapidly changing set of forces
- Design, measurement, and evaluation

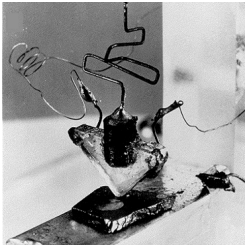
How Do the Pieces Fit Together?



- Coordination of many levels of abstraction
- Under a rapidly changing set of forces
- Design, measurement, and evaluation

A Bit of History

When was the first transistor invented?

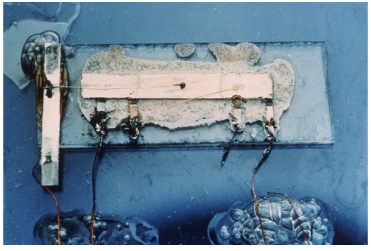


(a)

(b)

(a) 1947, bi-polar transistor, by John Bardeen et al. at Bell Laboratories; (b) UNIVAC I (Universal Automatic Computer): the first commercial computer in USA.

When was the first IC (integrated circuit) invented?

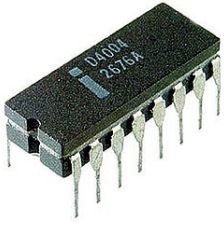


(a)

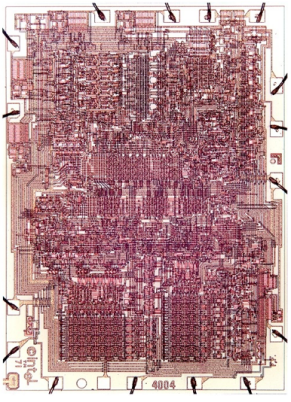
(b)

(a) 1958, by Jack Kilby@Texas Instruments, by hand. Several transistors, resistors and capacitors on a single substrate. (b) IBM System/360, 2MHz, 128KB – 256KB.

When was the first Microprocessor?



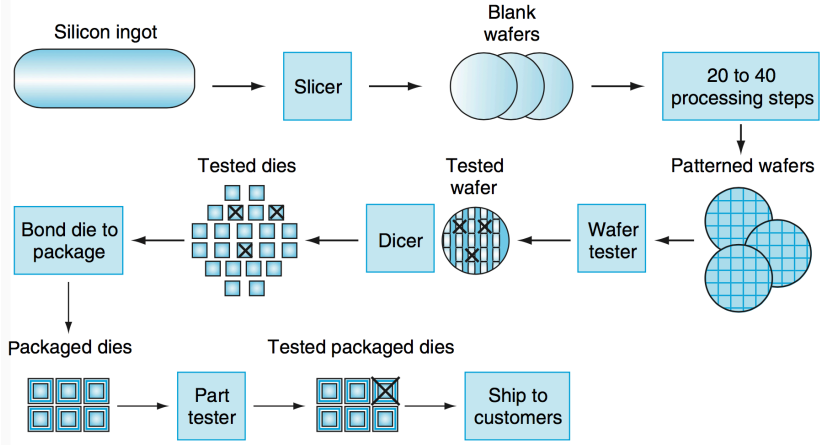
(a)



(b)

1971, Intel 4004.

The IC Manufacturing Process

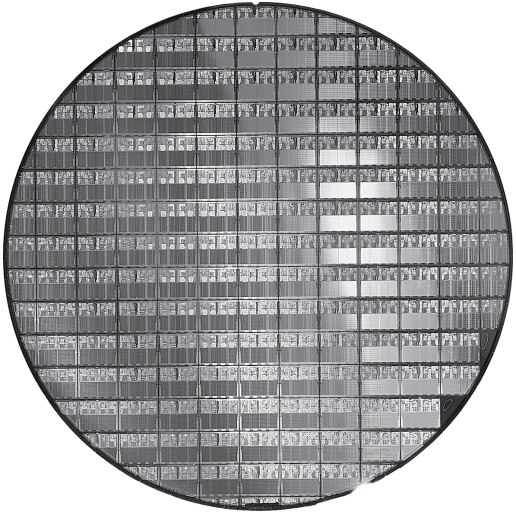


Yield

Proportion of working dies per wafer

Check this: https://youtu.be/d9SWNLZvA8g?list=FLELqiXCJQW-jcijW8ZAbA8w

AMD Opteron X2 Wafer



300mm wafer, 117 chips, 90nm technology.

Integrated Circuit Cost

$$\text{Cost per die} = \frac{\text{Cost per wafer}}{\text{Dies per wafer} \cdot \text{Yield}}$$

$$\text{Dies per wafer} = \text{Wafer area} \;/\; \text{Die area}$$

$$\text{Yield} = \frac{1}{[1 + (\text{Defects per area} \cdot \text{Die area} \;/\; 2)]^2}$$

Nonlinear relation to area and defect rate

- Wafer cost and area are fixed
- Defect rate determined by manufacturing process
- Die area determined by architecture and circuit design

Impacts of Advancing Technology

Processor

- Logic capacity: increases about 30% per year
- Performance: 2× every 1.5 years

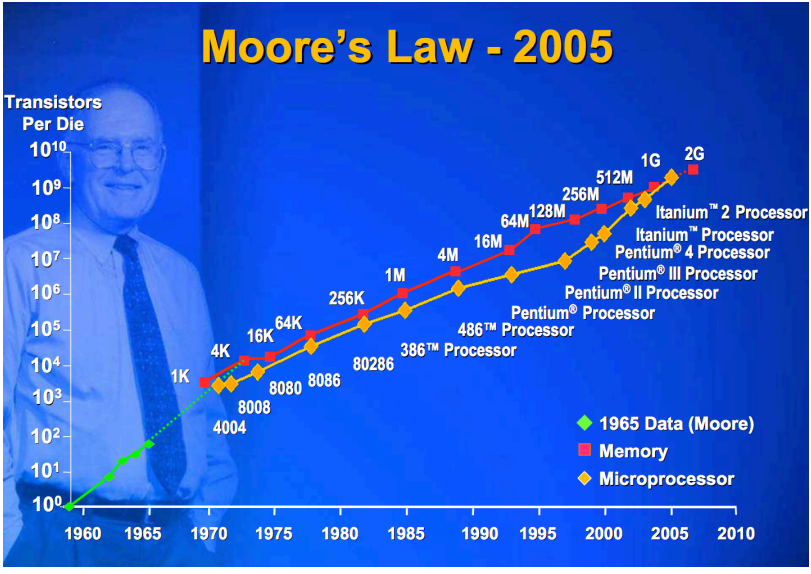
Memory

- DRAM capacity: 4× every 3 years, about 60% per year
- Memory speed: 1.5× every 10 years
- Cost per bit: decreases about 25% per year

Disk

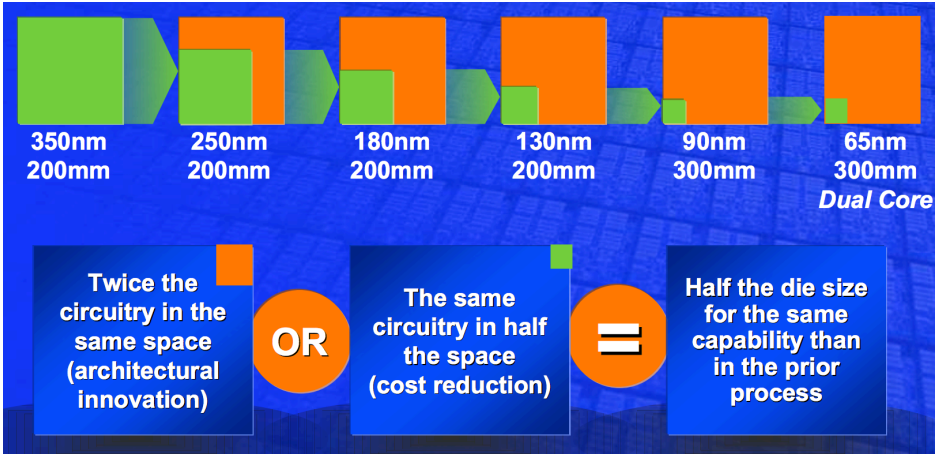
- Capacity: increases about 60% per year

Moore's Law for CPUs and DRAMs



From: "*Facing the Hot Chips Challenge Again*", Bill Holt, Intel, presented at Hot Chips 17, 2005.

Main driver: device scaling ...



From: "*Facing the Hot Chips Challenge Again*", Bill Holt, Intel, presented at Hot Chips 17, 2005.

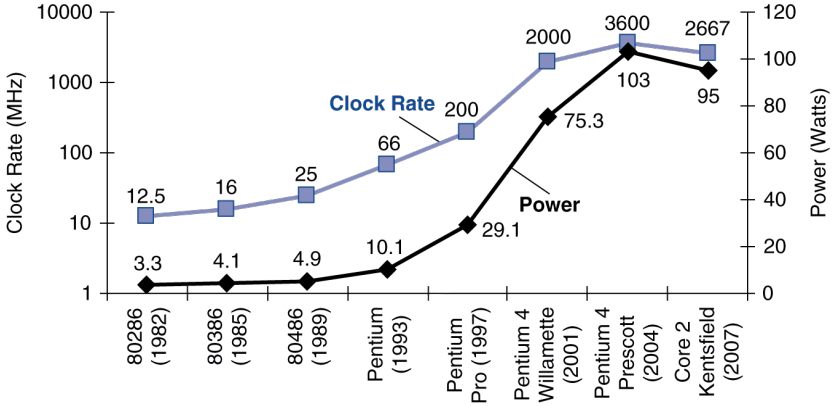
Technology Scaling Road Map (ITRS)

Year	2004	2006	2008	2010	2012
Feature size (nm)	90	65	45	32	22
Intg. Capacity (BT)	2	4	6	16	32

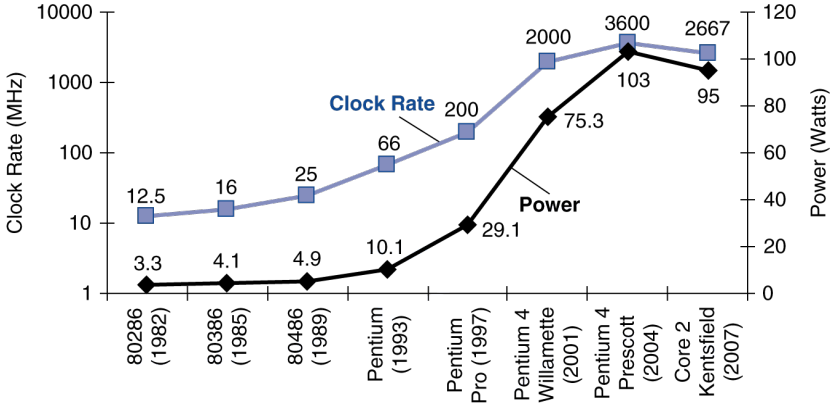
Fun facts about 45nm transistors

- 30 million can fit on the head of a pin
- You could fit more than 2,000 across the width of a human hair
- If car prices had fallen at the same rate as the price of a single transistor since 1968, a new car today would cost about 1 cent

Highest Clock Rate of Intel Processors



Highest Clock Rate of Intel Processors



What if the exponential increase had kept up? Why not?

- Due to process improvements
- Deeper pipeline
- Circuit design techniques

Power Issue

$$\text{Power} = \text{Capacitive load} \cdot \text{Voltage}^2 \cdot \text{Frequency}^1$$

Example

For a simple processor, if capacitive load is reduced by 15%, voltage is reduced by 15%, maintain the same frequency, how much power consumption can be reduced?

- A: 27.8%
- B: 38.6%
- C: 85.0%

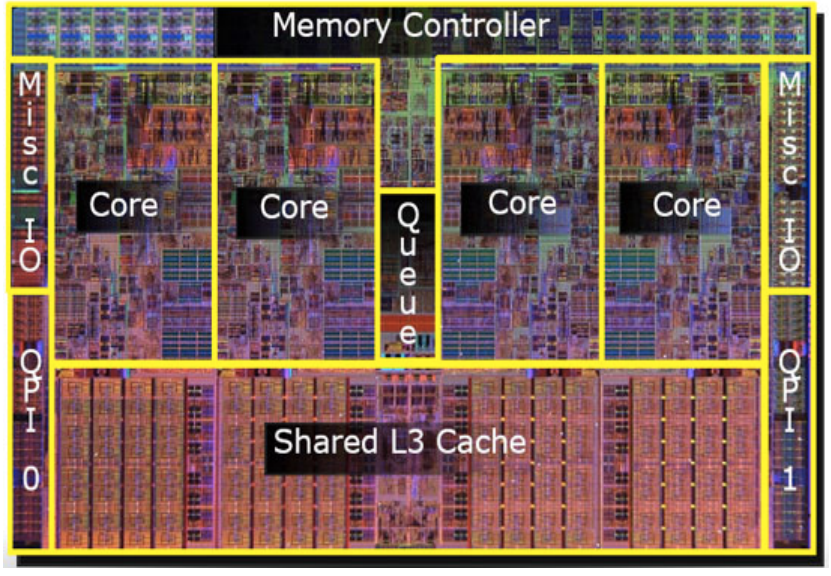
[1] here we only consider dynamic power, but not static power

A Sea Change Is at Hand

- The power challenge has forced a change in the design of microprocessors
- Since 2002 the rate of improvement in the response time of programs on desktop computers has slowed from a factor of 1.5 per year to less than a factor of 1.2 per year
- As of 2006 all desktop and server companies are shipping microprocessors with multiple processors – cores – per chip
- Plan of record is to add two cores per chip per generation (about every two years)

Product	AMD Barcelona	Intel Nehalem	IBM Power 6	Sun Niagara 2
Cores per chip	4	4	2	8
Clock rate	~2.5 GHz	~2.5 GHz	4.7 GHz	1.4 GHz
Power	120 W	~100 W	~100 W	94 W

Intel Core i7 Processor



45nm technology, 18.9mm x 13.6mm, 0.73billion transistors, 2008

A Computer

Desktop computers

Designed to deliver good performance to a single user at low cost usually executing 3rd party software, usually incorporating a graphics display, a keyboard, and a mouse

Other Classes of Computers

Servers

Used to run larger programs for multiple, simultaneous users typically accessed only via a network and that places a greater emphasis on dependability and (often) security

Supercomputers

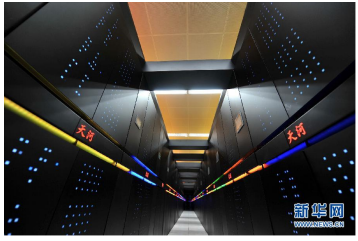
A high performance, high cost class of servers with hundreds to thousands of processors, terabytes of memory and petabytes of storage that are used for high-end scientific and engineering applications.

Embedded computers (processors)

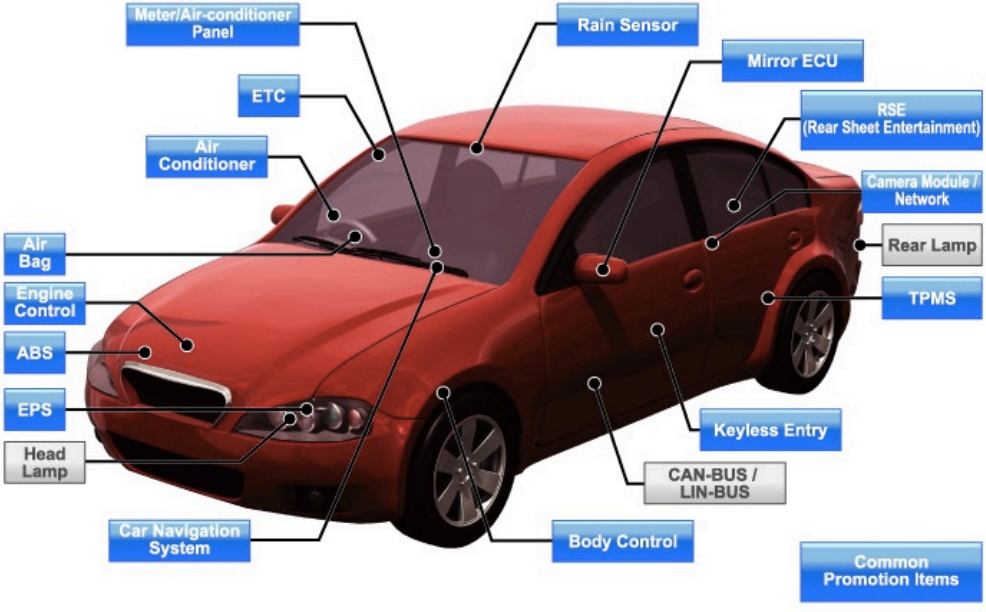
A computer inside another device used for running one predetermined application

Tianhe-2 (MilkyWay-2)

- Over 3 million cores
- Power: 17.6 MW (24 MW with cooling)
- Speed: 33.86 PFLOPS (peta $= 10^{15}$)



Embedded Computers in You Car



Personal Mobile Device (PMD)

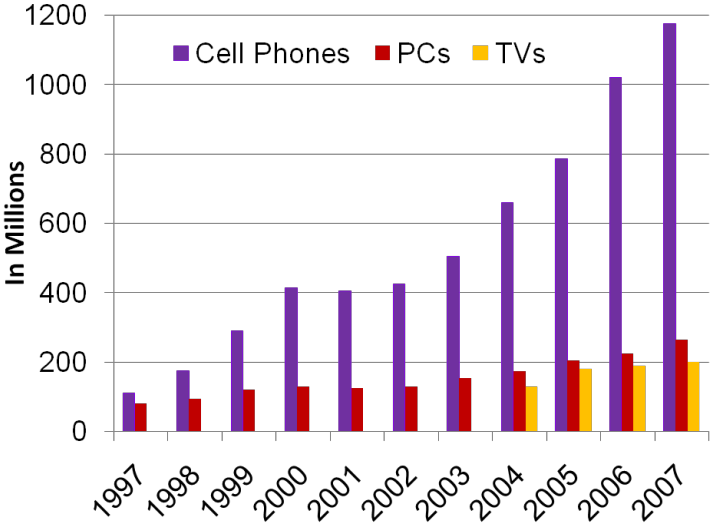
Battery-operated device with wireless connectivity

Warehouse Scale Computer (WSC)

Datacenter containing hundreds of thousands of servers providing software as a service (**SaaS**)

Growth in Cell Phone Sales (Embedded)

- embedded growth >> desktop growth
- Where else are embedded processors found?



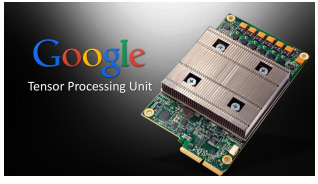
When Machine Learning Meets Hardware

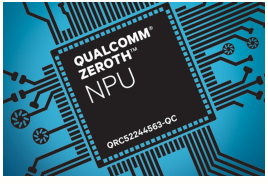
Convolution layer is one of the most expensive layers

- Computation pattern
- Emerging challenges

More and more end-point devices with limited memory

- Cameras
- Smartphone
- Autonomous driving





Convolutional Neural Network (CNN)

Autonomous drive

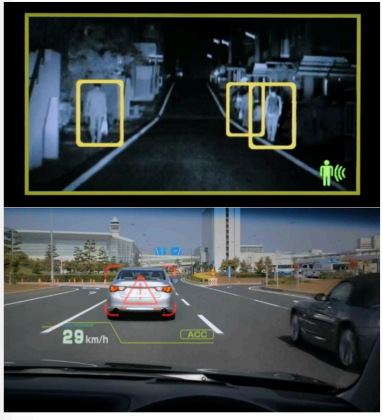
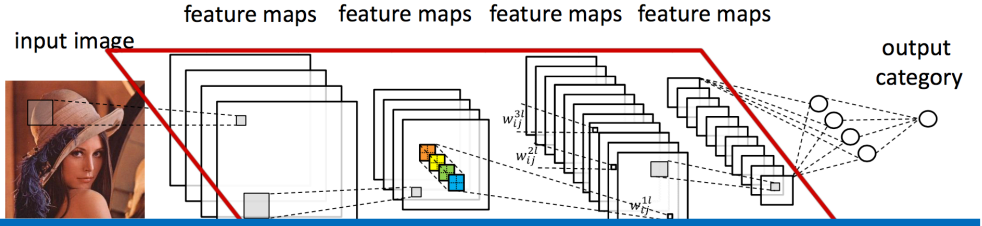


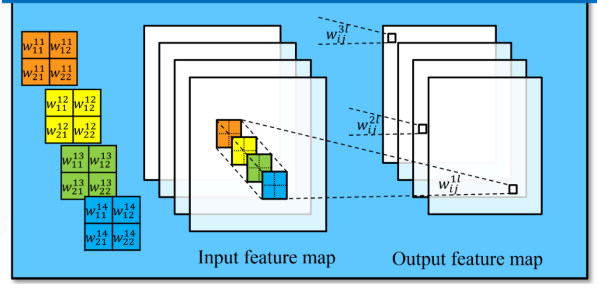
Image recognition

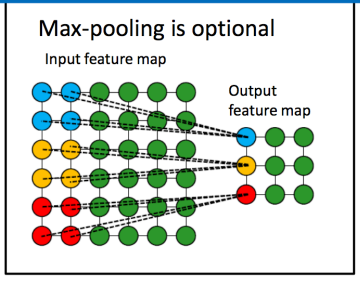
Bottleneck of CNN



Convolutional layers account for over 90% computation

[1] A. Krizhevsky, etc. Imagenet classification with deep convolutional neural networks. NIPS 2012.
[2] J. Cong and B. Xiao. Minimizing computation in convolutional neural networks. ICANN 2014





In-Datacenter Performance Analysis of a Tensor Processing Unit™

Norman P. Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, Rick Boyle, Pierre-luc Cantin, Clifford Chao, Chris Clark, Jeremy Coriell, Mike Daley, Matt Dau, Jeffrey Dean, Ben Gelb, Tara Vazir Ghaemmaghami, Rajendra Gottipati, William Gulland, Robert Hagmann, C. Richard Ho, Doug Hogberg, John Hu, Robert Hundt, Dan Hurt, Julian Ibarz, Aaron Jaffey, Alek Jaworski, Alexander Kaplan, Harshit Khaitan, Daniel Killebrew, Andy Koch, Naveen Kumar, Steve Lacy, James Laudon, James Law, Diemthu Le, Chris Leary, Zhuyuan Liu, Kyle Lucke, Alan Lundin, Gordon MacKean, Adriana Maggiore, Maire Mahony, Kieran Miller, Rahul Nagarajan, Ravi Narayanaswami, Ray Ni, Kathy Nix, Thomas Norrie, Mark Omernick, Narayana Penukonda, Andy Phelps, Jonathan Ross, Matt Ross, Amir Salek, Emad Samadiani, Chris Severn, Gregory Sizikov, Matthew Snelham, Jed Souter, Dan Steinberg, Andy Swing, Mercedes Tan, Gregory Thorson, Bo Tian, Horia Toma, Erick Tuttle, Vijay Vasudevan, Richard Walter, Walter Wang, Eric Wilcox, and Doe Hyun Yoon

Google, Inc., Mountain View, CA USA

Email: {jouppi, cliffy, nishantpatil, davidpatterson}@google.com

To appear at the 44th International Symposium on Computer Architecture (ISCA), Toronto, Canada, June 26, 2017.

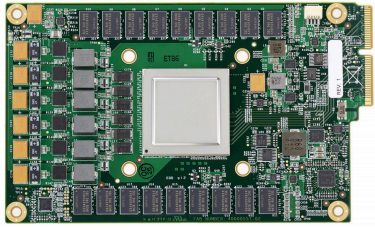


Figure 3. TPU Printed Circuit Board. It can be inserted in the slot for an SATA disk in a server, but the card uses PCIe Gen3 x16.

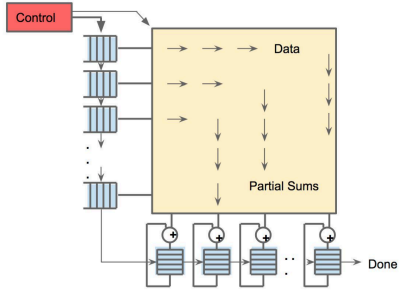
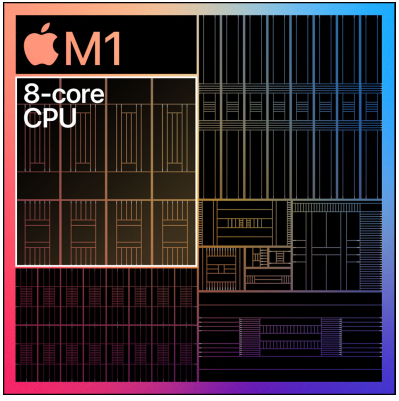


Figure 4. Systolic data flow of the Matrix Multiply Unit. Software has the illusion that each 256B input is read at once, and they instantly update one location of each of 256 accumulator RAMs.

Apple M1 Processor



- 8-core CPU
- 8-core GPU
- 16-core Neural Engine