Welcome to CSE370

- Instructor: Bruce Hemingway
  - TAs: Jon Froehlich and Iain Liechti
  - Lab Specialist: Michelle Goodstein
- Class web
  - http://www.cs.washington.edu//370
  - Add yourself to the mailing list→ see the web page
- Today's lecture
  - Course overview
  - The Digital Age

Text

Contemporary Logic Design (2nd Edition)

- **Randy H. Katz, U. California, Berkeley**
  and
  Gaetano Borriello, U. Washington, Seattle
- You will be using a pre-publication version of the textbook this quarter.
- Softbound copies of the book as a coursepak are available at the Communication Copy Center in the Communications bldg, Rm B-042, cost: $44.60.
- **DO NOT USE A PREVIOUS VERSION OF THIS COURSEPAK!**
- **This has been extensively revised this summer; you will do the wrong homework problems if you use an old version!**

Workload

- The course consists of the following elements:
- **Lectures:** There will be 28 lectures. Attendance and participation at all of them is strongly encouraged and expected.
- **Laboratory Assignments:** There will be a total of eight (8) laboratory assignments (there will not be a laboratory meeting during the first week and the last assignment will span two weeks). Although you'll be able to use the lab all week, attendance at one of the scheduled times is very important as that is when the TAs will be available. We will work hard to ensure that the laboratory assignments take no more than the three hour sessions to complete. Laboratory assignments will be closely tied to the written homework assignments and are intended to give you a taste of working with real digital hardware. We will use them to reinforce key concepts. You should attend the session for which you are registered. With permission of the TA, you can attend the other section in case of unusual circumstances.
- **Reading:** We will cover most of the Contemporary Logic Design (2nd edition) text. Readings will be part of each weekly assignment.
- **Assignments:** Weekly problem sets involving digital logic analysis and design, to be solved with and without the use of computer-aided design tools. The last assignment will include a larger design project and will span two weeks.
- **In-class Quizzes:** Four in-class quizzes, scheduled biweekly throughout the quarter. Together these replace a mid-term exam. Each quiz will be approximately 15 minutes.
- **Final exam:** A two-hour exam during finals week.

Grading

- We will compute your course grade as follows:
- 30%: weekly assignments
- 20%: laboratory assignments
- 20%: biweekly quizzes
- 30%: final exam
- Your grade will be determined by how well you understand the material as evidenced by the assignments, labs and tests. We would like nothing better than to give the entire class a 4.0

Homework and Quizzes

- **Assignments**
- Your weekly assignments are due at the beginning of class on the assigned due date. Assignments handed in during or immediately after class will incur a 10% penalty. We will penalize your assignment 10% per day for each additional day late. Assignments due Friday will be charged 20% if turned in over the weekend, 30% if turned in on Monday, etc.
- Assignment problems will sometimes be graded on a random basis. To get full credit for an assignment, you must, of course, turn-in solutions for each assigned problem. Only a subset of the problems will actually be graded in detail. You will not know in advance which problems this will be - so make sure to do all of them.
- Please review the assignment solutions carefully before questioning a grade with either the instructor or the teaching assistants.
- **Quizzes**
- There will be no makeup for missed quizzes. If you miss a quiz, you will receive a score of zero so please plan your schedule carefully. We do not have the resources to be able to give make-up quizzes. Please review the quiz solutions carefully before questioning a grade with either the instructor or the teaching assistants.

Collaboration and Cheating

- **Collaboration**
- **Homework:** Unless specifically stated otherwise, we encourage collaboration on homework, provided (1) You spend at least 15 minutes on each and every problem alone, before discussing it with others, and (2) You write up each and every problem in your own writing, using your own words, and understand the solution fully. Copying someone else's homework is cheating (see below), as is copying the homework from another source (prior year's notes, etc.). The quiz problems will be very similar to the homework problems; if you truly understand the homework, then the quizzes will be easy. If you have copied the homework... **Quizzes:** A quiz is a short exam—*no collaboration or discussion is permitted*. If you have a question during a quiz, ask the instructor.
- **Cheating**
- Cheating is a very serious offense. If you are caught cheating, you can expect a failing grade and initiation of a cheating case in the University system. Basically, cheating is an insult to the instructor, to the department and major program, and most importantly, to you. If you feel that you are having a problem with the material, or don't have time to finish an assignment, or have any number of other reasons to cheat, then talk with the instructor. Just don't cheat. To avoid creating situations where copying can arise, never e-mail or post your solution files. You can post general questions about interpretation and tool use but limit your comments to these categories. If in doubt about what might constitute cheating, send the instructor email describing the situation.

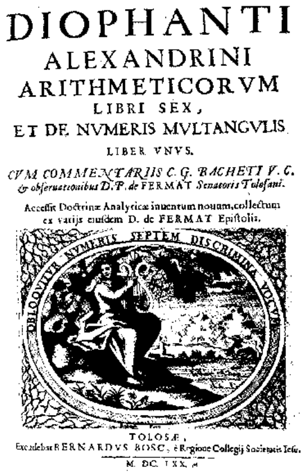
Why you are here

- Required class
- To learn about digital design
  - Design process and techniques
  - The basis for digital computing
- Exposure to new ideas
  - Emergent behavior
    - Complex functions from simple elements
    - With only NORs and wire you can build a computer
  - Parallel computation
    - Digital hardware is inherently parallel

The Digital Age

- Computing is in its infancy
  - Processing power
    - Doubles every 18 months
    - Factor of 100 / decade
  - Disk capacity
    - Doubles every 12 months
    - Factor of 1000 / decade
  - Optical fiber transmission capacity
    - Doubles every 9 months
    - Factor of 10,000 / decade
- The bases are mathematics and switches
  - How did we get here?

Diophantus of Alexandria b. ~200 BCE



Known as the "father of algebra"

Arithmetica **is a collection of 130 problems that gives numerical solutions of determinate equations, which have a unique solution, and indeterminate equations.**

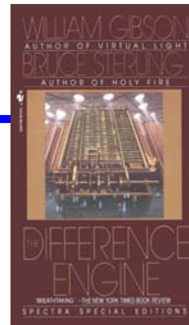
The Later Alexandrian Age was a time when mathematicians were discovering many ideas that lead to our concept of mathematics today.

850 AD

- Abu Ja'far Muhammad ibn Musa **al-Khwarizmi**
- Lived in Baghdad, 780 to 850 AD. One of the first to write on algebra (using words, not letters) and also Hindu-Arabic numbers (1, 2, 3, ...).
- From his name and writings came the words "algebra" and "algorithm".
- Book: *Hisab **al-jabr** w'al-muqabala*

1822

- Charles Babbage
  - Father of computing
- 1822 Difference Engine
  - A calculator
- 1834 Analytical Engine
  - A computer
  - Programmable



Analytical
Engine

1854

- George Boole
  - Boolean algebra
- Number system with 2 values
  - 0/1 ⇔ false/true
  - Do math on logic statements
  - 3 operations (NOT, AND, OR)

All computers use
Boolean algebra

NOT

A	Out
0	1
1	0

AND

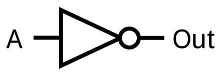
A	B	Out
0	0	0
0	1	0
1	0	0
1	1	1

OR

A	B	Out
0	0	0
0	1	1
1	0	1
1	1	1

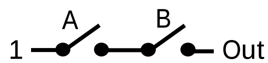
1938

- Claude Shannon
  - Implemented Boolean algebra using switches
  - Described information using binary digits (bits)



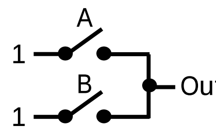
NOT

A	Out
0	1
1	0



AND

A	B	Out
0	0	0
0	1	0
1	0	0
1	1	1

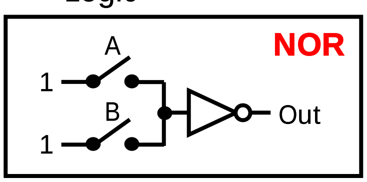


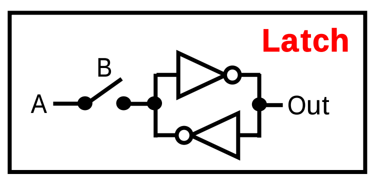
OR

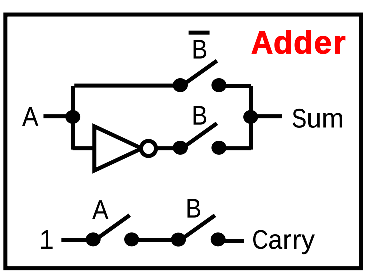
A	B	Out
0	0	0
0	1	1
1	0	1
1	1	1

Computer Hardware

- Components
  - Logic
  - Memory





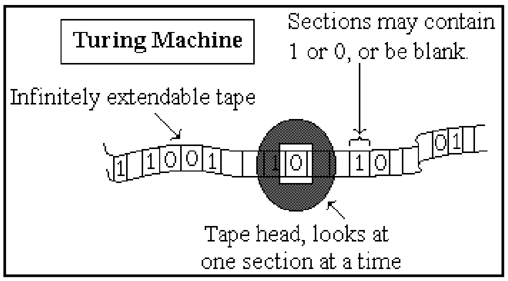


Adder

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

1937

- Alan Turing
  - Turing Machines
- Simple computer model
  - Can something be computed?



Also pioneered artificial intelligence

1945

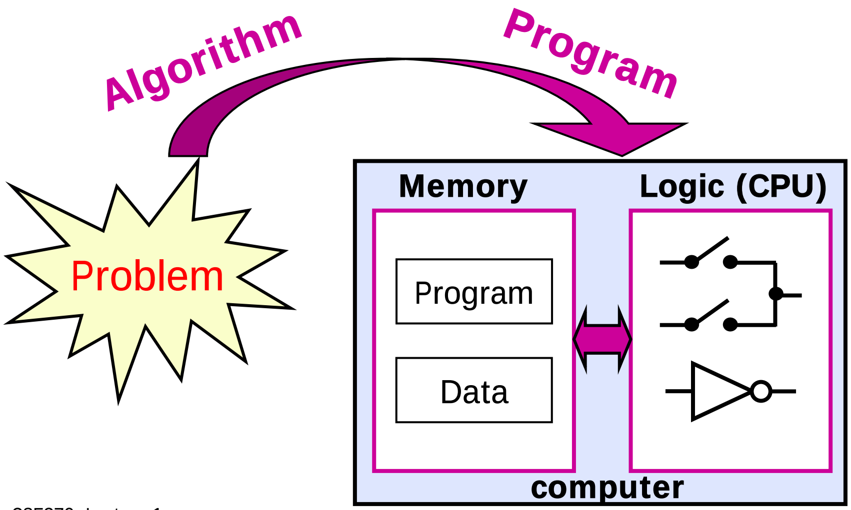
- John von Neumann
  - First stored computer program
- A sequence of operations
  - Read from memory
  - Operate using logic gates
  - Store result into memory

Other contributions:
Quantum Mechanics
Cellular Automata
Game Theory

Stored Programs = Software

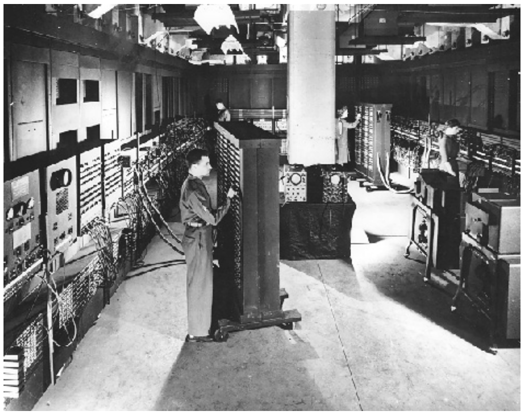
Bill Gates and Paul Allen, Lakeside, 1968

Hardware + Software



1946

- ENIAC...the first computer
  - Vacuum tubes for switches



1000x faster than anything before...

19,000 tubes

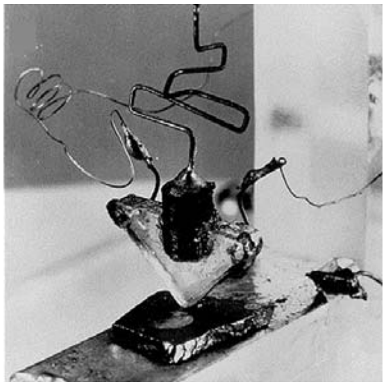
200 kilowatts

357 multiplies per second

1947

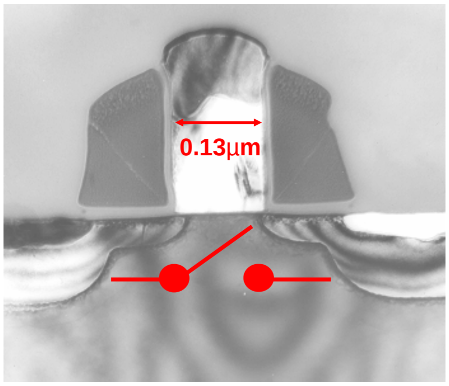
- Bardeen, Brattain, Shockley invent the transistor

1947



Nobel Prize, 1956

2000



Courtesy Mark Bohr, Intel

1958

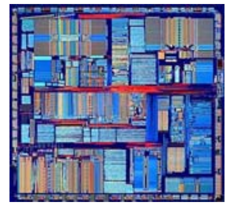

- Kilby and Noyce invent the integrated circuit

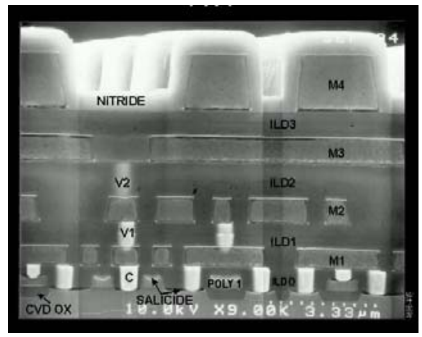
1958

Nobel Prize, 2000

2000

Pentium

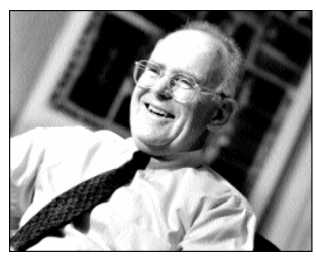


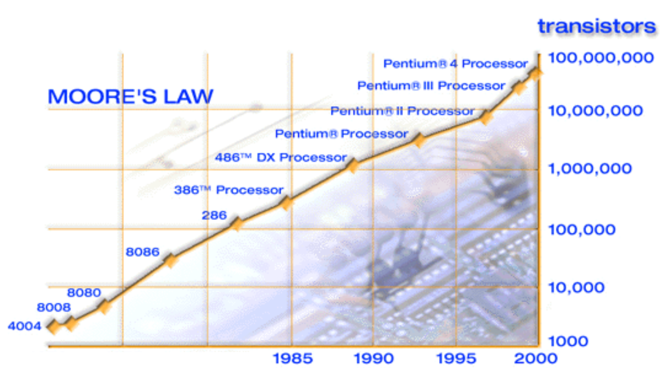


Courtesy Yan Borodovsky, Intel

1965

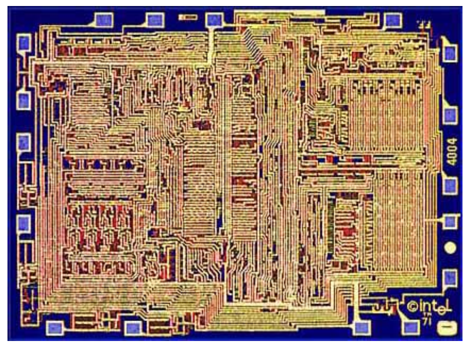
- Gordon Moore
  - Moore's Law: The transistor density of silicon chips doubles every 18 months



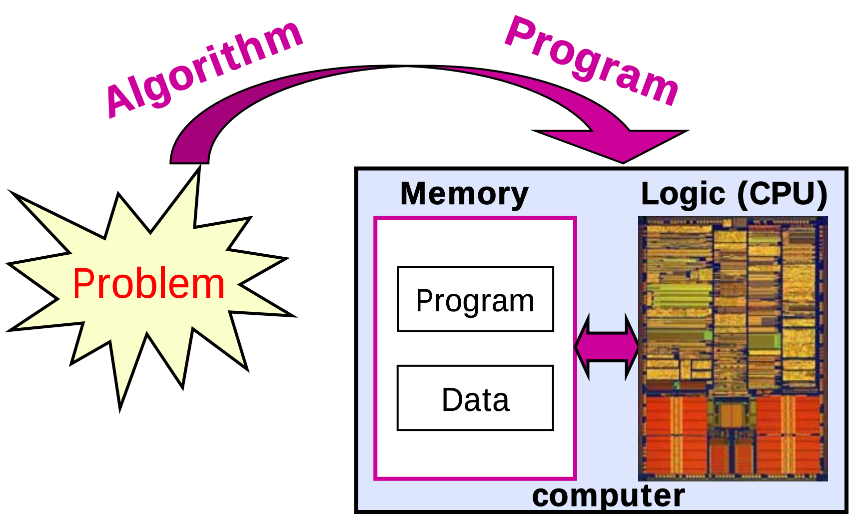


1971

- Ted Hoff invents the microprocessor
- Intel 4004
  - 2,300 transistors
  - 3 mm by 4 mm
  - As powerful as the ENIAC



Hardware + Software + Technology

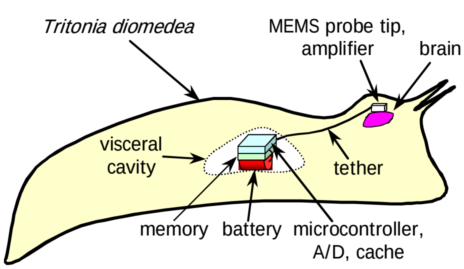


1977 and 1981

- Apple II and IBM PC
  - The first microcomputers

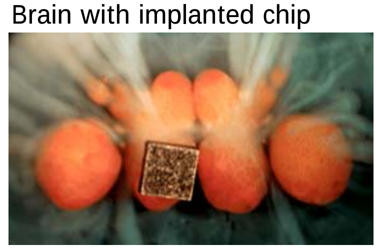
A modern example

- Goal: Interface a computer to an animal brain
  - Measure brain signals in intact animals



Tritonia and seapen

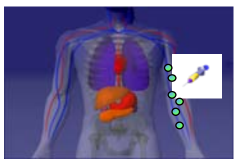
Brain with implanted chip



Courtesy Jim Beck and Russell Wyeth

More modern examples

- Computing everywhere
  - Wireless/wired networking
  - Wearable devices
  - Smart sensors



What is logic design?

- Using digital logic...
  - The underlying basis is Boolean algebra
  - The physical basis is transistor switches
- ...to solve a problem...
  - Within size, cost, and other bounds
  - Within the constraints imposed by our bases
    - Encode as logical statements
    - Compile into physical hardware
- ...with logical values encoded as physical quantities
  - If (0V < *voltage* < 0.8V) then *symbol* is a “0”
  - If (2.0V < *voltage* < 5V) then *symbol* is a “1”

Terminology

- Digital: Discrete-valued
  - Usually binary
  - Transistor switches have 2 states (on/off)
- Combinational: Without memory
  - Output depends on present input
- Sequential: With memory (state)
  - Output depends on present and/or past inputs
- Synchronous: Values change at discrete timesteps
  - Synchronous ⌧ clocked

What you will learn in CSE370

- Physical devices (transistors, resistors, wires)
- Switches
- Truth tables
- Boolean algebra
- Combinational logic
- Sequential logic
- State in digital systems
- Finite-state machines
- Hardware description languages
- Register-transfer description
- Concurrent abstract specifications