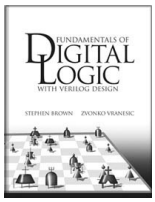
Welcome to CSE370

- Instructor: Bruce Hemingway
  - TAs: Bryan Nelson and John Hwang
  - Tool Specialist: Cory Crawford
- Class web
  - http://www.cs.washington.edu/education/courses/370/CurrentQtr/
  - Add yourself to the mailing list→ see the web page
- Today's lecture
  - Course overview
  - The Digital Age

Special Thanks!

Lecture Materials:

- Chris Diorio
- Gaetano Borrielo
- Carl Ebeling



- **Stephen Brown and Zvonko Vranesic, *Fundamentals of Digital Logic with Verilog Design*, 1st Edition, McGraw-Hill, 2003**

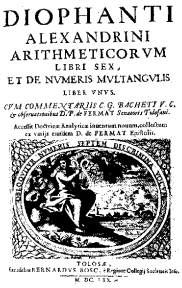
Why you are here

- 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


DIOPHANTI
ALEXANDRINI
ARITHMETICORVM
LIBRI SEX,
ET DE NVMERIS MVLTANGVLIS
LIBER VNVS.
CVM COMMENTARIIS C. G. BACHETI V. C.
& observationibus D. P. de FERMAT Senatoris Tolosani.
Accessit Doctrinæ Analyticæ inventorum nostrum collectio
ex variis epistolis D. de FERMAT Excerpta.
TOLOSÆ
Excudebat BERNARDVS BOSC, Regiorum Collegij Lectionum Typ.
M. DC. LIX.

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.**

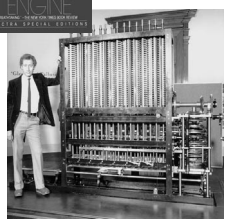
The Later Alexandrian Age was a time when mathematicians were discovering many ideas that lead to our concept of mathematics today.

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

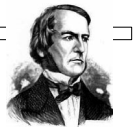
- Charles Babbage
  - Father of computing
- 1822 Difference Engine
  - A calculator
- 1834 Analytical Engine
  - A computer
  - Programmable



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

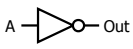
A	B	Out
0	0	0
0	1	0
1	0	0
1	1	1

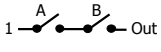
OR

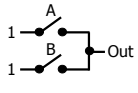
A	B	Out
0	0	0
0	1	1
1	0	1
1	1	1

1938

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







NOT

A	Out
0	1
1	0

AND

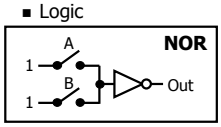
A	B	Out
0	0	0
0	1	0
1	0	0
1	1	1

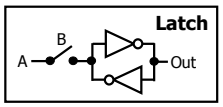
OR

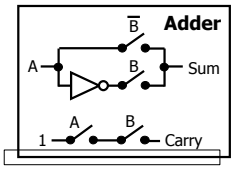
A	B	Out
0	0	0
0	1	1
1	0	1
1	1	1

Computer Hardware

- Components
  - Logic
  - Memory







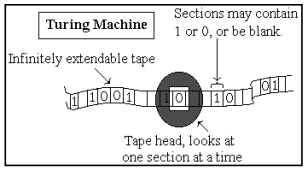
Adder

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

1937

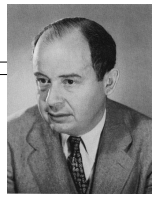
- Alan Turing
  - Turing Machines
- Simple computer model
  - Can something be computed?

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

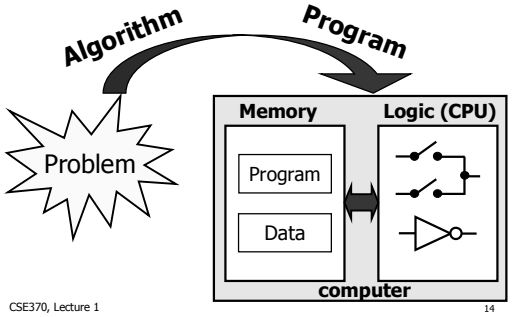


Other contributions:
Quantum Mechanics
Cellular Automata
Game Theory

Stored Programs = Software

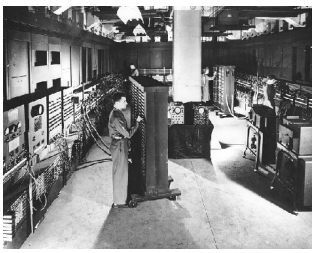
Bill Gates and Paul Allen, Lakeside, 1968

Hardware + Software



1946

- ENIAC…the first computer
  - Vacuum tubes for switches



1000x faster than anything before...

19,000 tubes

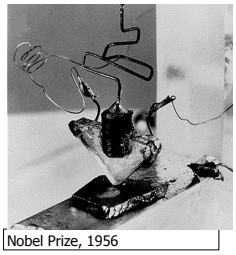
200 kilowatts

357 multiplies per second

1947

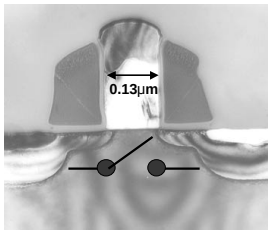
- Bardeen, Brattain, Shockley invent the transistor

1947



Nobel Prize, 1956

2000



Courtesy Mark Bohr, Intel

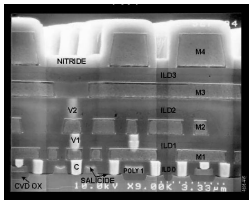
1958

- Kilby and Noyce invent the integrated circuit

1958

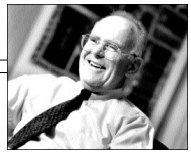
Nobel Prize, 2000

2000

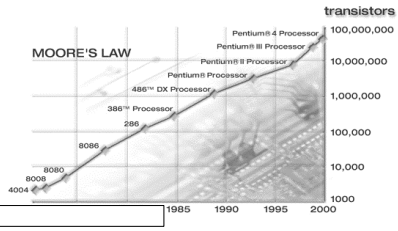


Courtesy Yan Borodovsky, Intel

1965



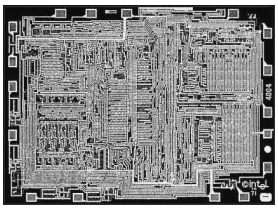
- Gordon Moore
  - Moore's Law: The transistor density of silicon chips doubles every 18 months



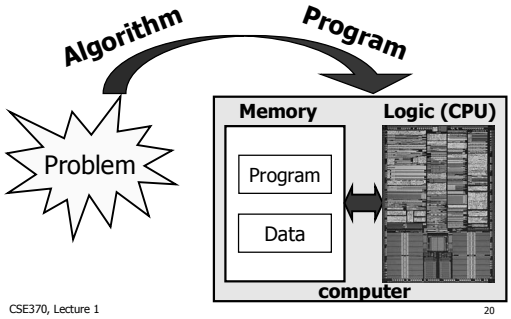
1971

- Ted Hoff invents the microprocessor

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

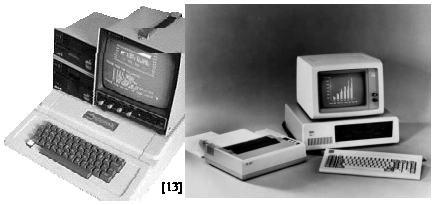


Hardware + Software + Technology



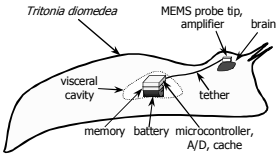
1977 and 1981

- Apple II and IBM PC
  - The first microcomputers



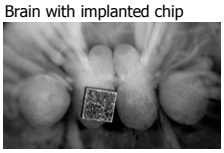
A modern example

- 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