ECE 6721:
Emerging Computing Technologies

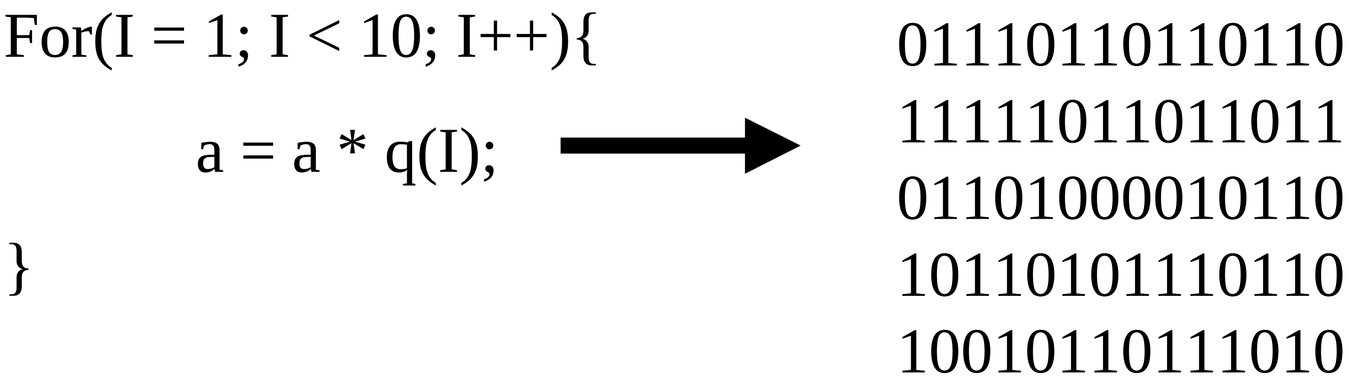
Lecture 2:
Motivation and History

What is Unconventional Computer Architecture?
What are Emerging Technologies?

- **Start by defining “Conventional” computer architecture**
  - **Uses stored-program model of computation**
  - **Implemented using silicon VLSI**
- **An unconventional computer architecture is one that doesn’t have both of these attributes**
  - **ASICs**
  - **Reconfigurable devices**
  - **Dataflow**
  - **Cellular Automata and Systolic Arrays**
  - **Quantum**
  - **Biological (DNA, proteine)**
  - **Molecular Electronics**
  - **reversible**
  - **optical**
  - **membrane computing**
  - **nano-technologies**
  - **….**

Stored-Program Model

- One of the key developments in early computing
  - Also known as “Von Neumann” model



```
For(I = 1; I < 10; I++){

          a = a * q(I);

}
```

```
01110110110110
11111011011011
01101000010110
10110101110110
10010110111010
```

Why is the Stored-Program Model Good?

- Treat *programs as data*
  - Load and store them from disk/punch cards
    - Much better than flipping switches
  - Programs can modify themselves
  - Programs can modify/create other programs
    - Assemblers
    - Compilers
    - Debuggers
- Stored-program Computers as ***Universal*** Devices
  - Church's thesis

Why Isn't the Stored-Program Model Perfect?

- Big complexity/computational power cost to provide *flexibility*
  - ASICs
  - Efficiency of Reconfigurable Logic
  - Large number of research projects based around "making programmable computers closer to custom VLSI"

Alternatives to Stored-Program Model

- Dataflow
  - Instructions explicitly encode dependencies
  - Goal is to expose fine-grained parallelism
  - Really an alternate form of stored-program
- Embedding Computation In Hardware
  - ASICs
  - Reconfigurable Logic

Why Look at Alternatives

- Growth of embedded systems
  - Flexibility less key than performance and hardware efficiency
  - Current state-of-the-art is hand-designed ASIC
- Changing application domain
  - Multi-phased streaming applications

Silicon VLSI

- Take a lump of silicon, throw in some *dopants*, make transistors
- Geometric scaling in density/speed
  - 50%/ year increase in density
  - 35%/ year increase in system performance has actually improved to 50%/ year in the last decade
    - Use of increased transistors/chip for architectural improvements
- Low cost, high reliability, acceptable yield
  - Very impressive given the variability of individual devices
- Integration creates new possibilities
  - Tremendous jump in system performance with each level of integration
    - Kinda **at the end** of the road for this
  - ***Mixed*-mode system-on-a-chip (SOC)** products as the next big thing

Limitations?

- Wire delay already becoming a limiting factor on system speed
  - Transistors get faster, wires slower
  - Approaching 85% of overall delay
- Transistors (FETs) are bulk devices
  - Rely on having many atoms in each region
  - Becomes impossible to lightly dope regions as we get to .01-micron fabrication
  - Predictions :
    - HP recently quoted **2012** as end of the road
    - Current density curves get to <1 atom/bit by **2020**

Alternatives

- **Molecular/Quantum Electronics**
  - Most similar to current technology
  - Essentially replace FETs with other devices that have similar behavior
- **Carbon Nanotubes**
  - Mostly what the name says, tubes made out of carbon atoms
  - Can get switch-like behavior, make wires
- **Quantum Computing**
  - Expose quantum effects to the programming model
  - Offers potential for performance that's impossible in conventional systems

We will start with reversible and cellular concepts that are base of many technologies

What lessons can we learn from history of computing?

Who Invented the Computer?

- Questions we can ask:
  - **Who invented the computer?**
  - **Why was the computer invented?**
- It is more accurate to say the computer *evolved*, rather than that it was invented.
  - (In fact, no one owns a patent for the invention of computers.)
- Many prototypes were invented, each based on earlier work or ideas.

- Let’s look at the evolution of computers...

This will continue..

Let’s start at the beginning...

- Ancient times:
- People wanted to count things (sheep), to keep track of how many they had (last night I had 53 sheep). To help keep track of what they were counting, they used counting aids:
  - fingers
  - pebbles
  - notched sticks
  - knotted rope
  - etc.

Let’s start at the beginning...

- Ancient times continued:
- Some transactions (trade) required calculations (I traded 3 sheep for 5 bottles of wine... this morning I have 47 sheep and a headache... someone’s stealing my sheep!).

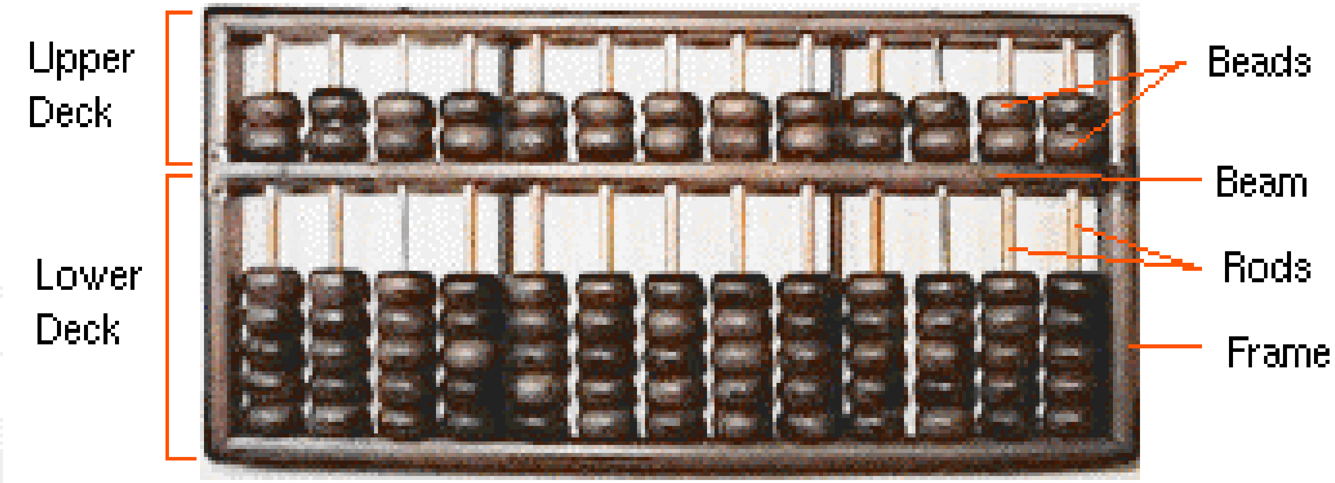
- Calculations are based on algorithms.

KEY CONCEPT

An *algorithm* is a step-by-step process that manipulates data.

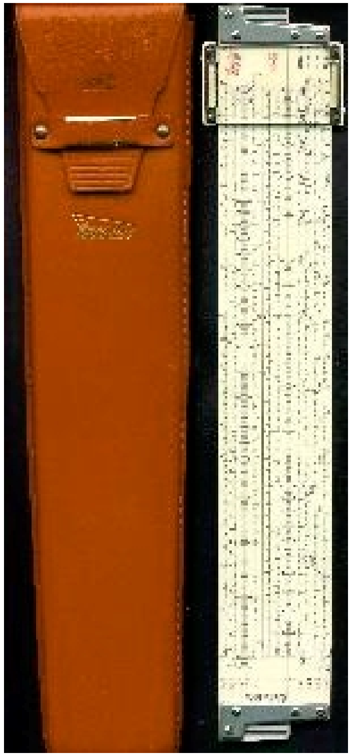
Let’s start at the beginning...

- Ancient times continued:
- The abacus was invented ~ 5,000(?) years ago by the Babylonians, later upgraded in Asia.



- The abacus is the original *mechanical* counting device. Possible operations include:
  - addition, subtraction, multiplication, and division
  - even fractions, root square and statistics

Continuing on...



- 1600s:

–In 1621, William Oughtred invented the slide rule (based on John Napier's logarithms).

–In 1642, Blaise Pascal invented the "Pascaline", the first *mechanical digital* calculator. Operations:

  - addition and subtraction
  - multiplication and division functionality added later

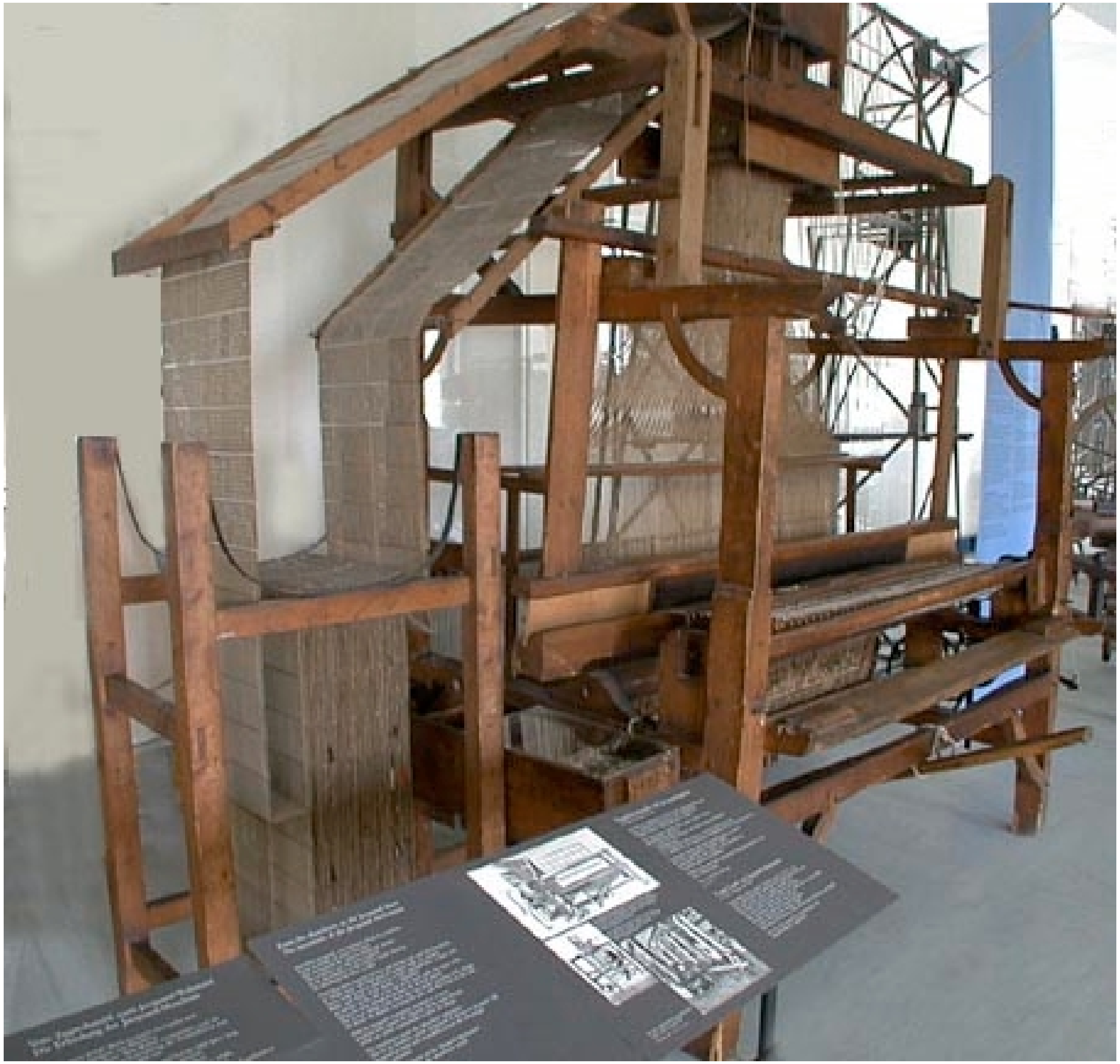
Side note

- 1800s:

–Joseph Jacquard (a silk weaver) automated the pattern-weaving process in 1804. He encoded patterns on punched cards, which were read by the machine.

–*So what?*

–First *programmable* machine.



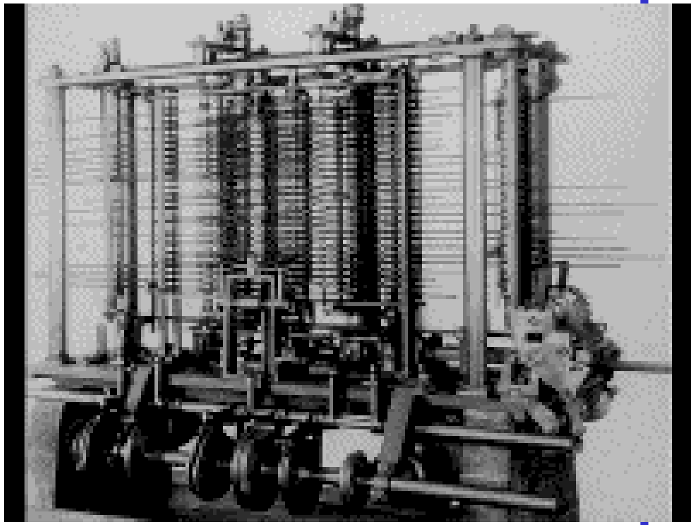
Continuing on...

- 1800s, continued...
- English mathematician Charles Babbage wanted to calculate using steam. Why?
  - human computers make too many mistakes.
  - steam was latest, greatest technology.
  - steam does physical tasks, why not mental?

- Proposed the difference engine. (1822)
  - wheels & shafts calculate using method of difference (easy process to mechanize)
  - printed results
  - never completed

Now we're getting somewhere...

- 1800s, continued...
- Charles Babbage moved on to the analytical engine. (1834)
  - *general purpose* calculating device
  - embodies many modern computing concepts:
    - memory
    - programmable processor
    - output device
    - user-definable input of programs & data
  - proposed using punched cards
  - never built

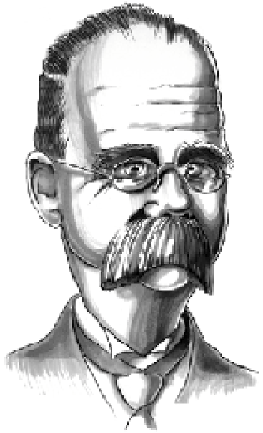


- Lady Ada Lovelace was the first programmer
  - suggested using binary
  - loops

Birth of Big Blue

–1800s, continued...

–1880s: problems completing US census on time, competition for 1890 census.



–Herman Hollerith won with his tabulating machine

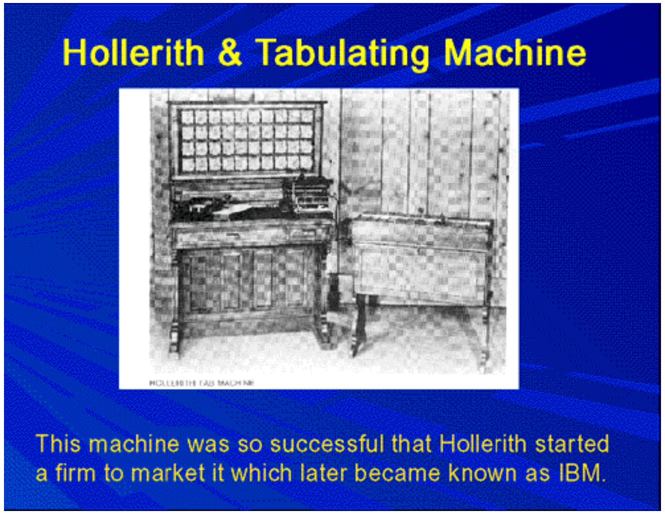
- – used punched cards
- – census took 6 months (plus 2 yrs)

•Hollerith started **The Tabulating**

•**Machine Company**, later became

•**International Business Machines**

•(IBM).



Hollerith & Tabulating Machine

This machine was so successful that Hollerith started a firm to market it which later became known as IBM.

Laziness as a virtue

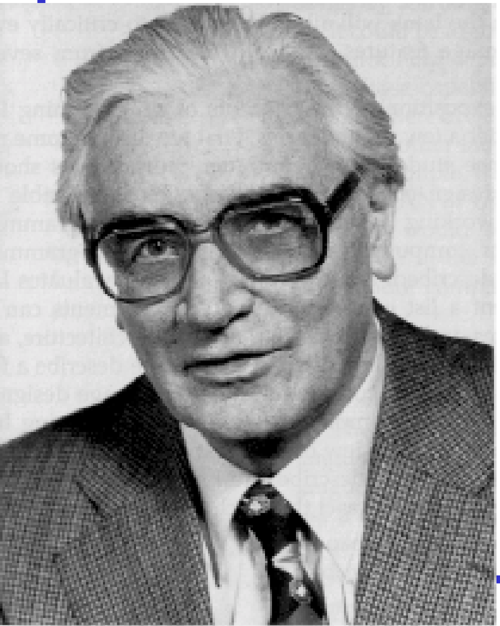
- 1930s–40s:

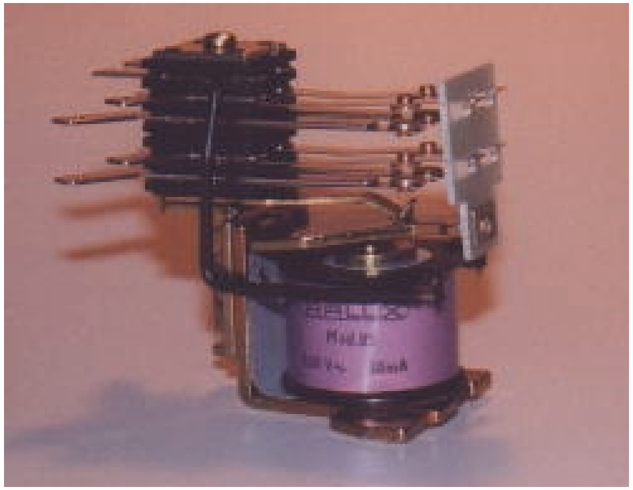
–Konrad Zuse was lazy: he didn’t want to perform calculations by hand, so he invented a computer.

- used electric relays, 2 states (on/off)
- used binary instead of decimal (easier to represent)

–War broke out, funding appeared.

- Konrad Zuse’s Z3:
- 1st programmable, general-purpose, *electromechanical* computer.





Delunay story. Laziness will continue..

Wartime Codebreaking

- So far, computers crunch numbers.

- British mathematician Alan Turing believed machines could do any theoretical process a human could do.

- Letters are just symbols: use machine to break codes.
COLOSSUS, a top secret machine to break the ENIGMA’s codes.

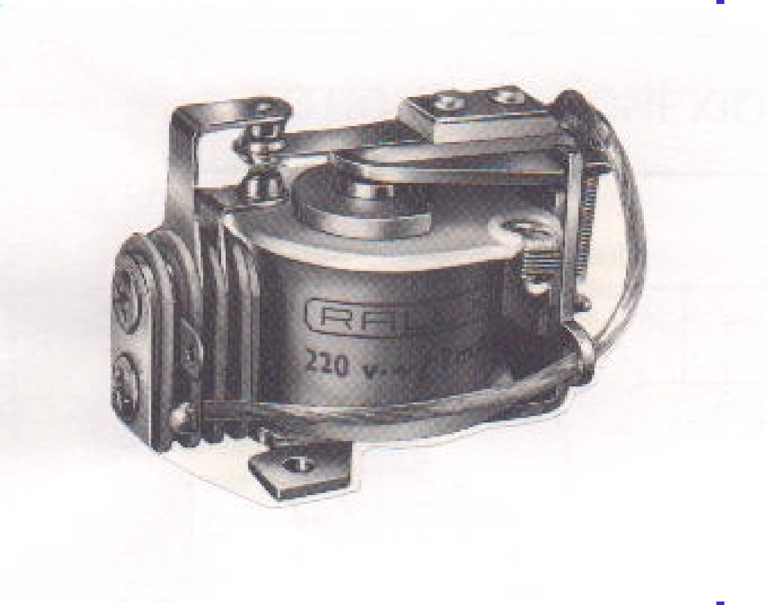
- Turing test:
  - Given 5 minutes, a keyboard & a monitor, if we are not more than 70% sure it is a machine, we have to admit it has shown some intelligence.

This will continue..

War

- America Enters the War, needed firing tables calculated ⇒ funding appears in USA

- Howard Aiken, and IBM:
  - Mark I: *Electromechanical* digital computer

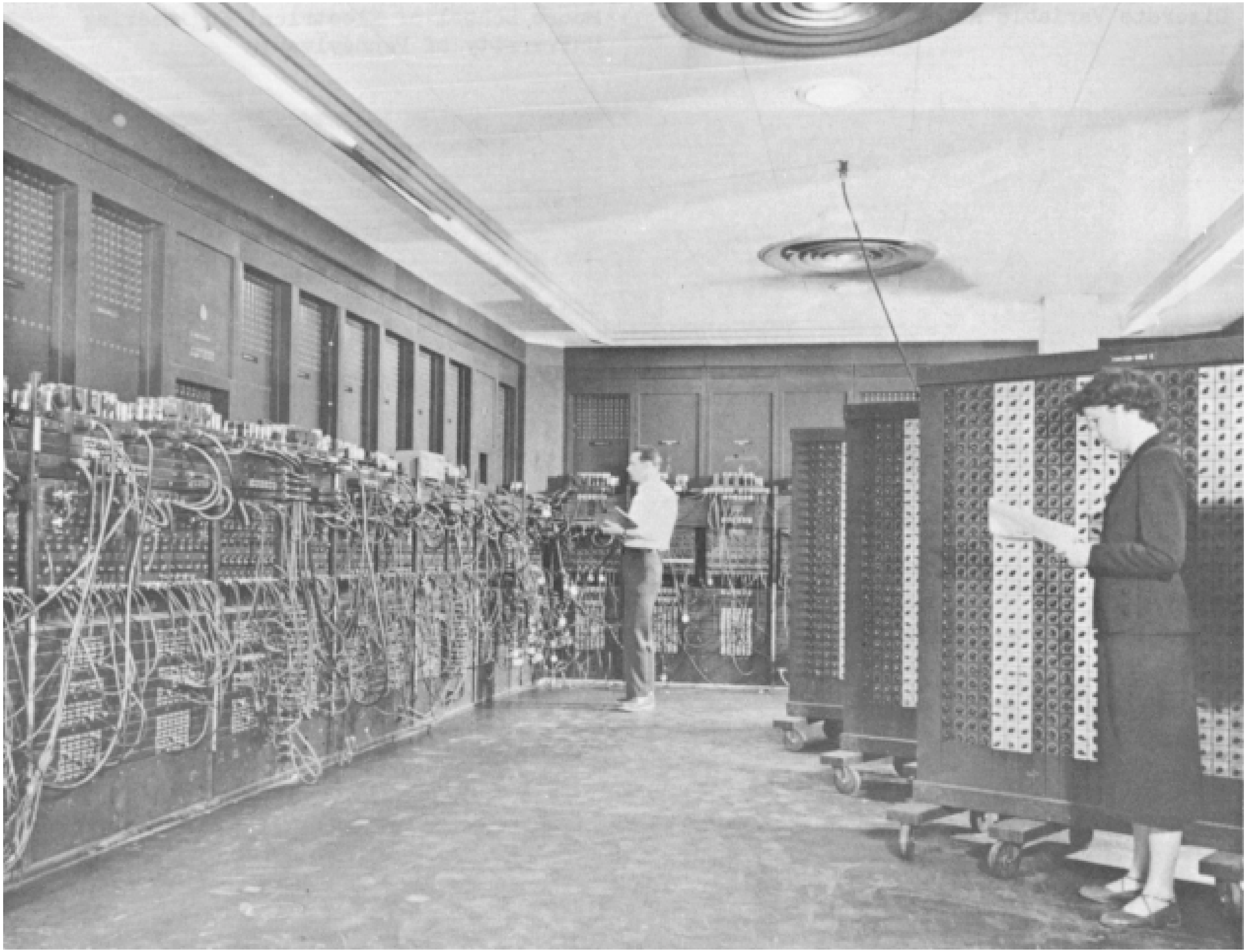


- Need something *faster*.

War

–J. Presper Eckert & John Mauchly

- ENIAC: Electronic Numerical Integrator & Computer
- vacuum tubes (speed increased thousandfold)
- Patented as 1st *electronic*, general-purpose computer in 1946. Patent later voided. (ABC first, see text pg. 389)
- ready *after* the war…. (oops)
- limitations:
  - no internal storage
  - rewire plugboards & set switches
  - took days to re-program
- knew problems, but no time to fix



Post-War

- Post-war: 1940s and early 50s
- John von Neumann
  - *memory* easier to change than rewiring hardware.
  - separate hardware & software
  - store program & data
  - theoretical blueprint for all future computers

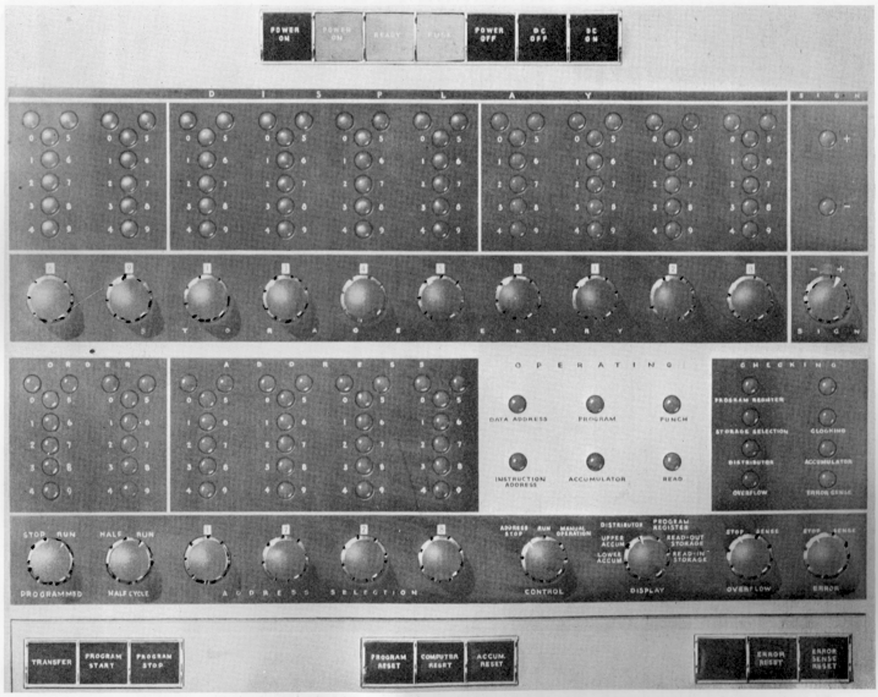
- Freddy Williams designed the **EDSAC**: the first stored-program computer.

- Eckert & Mauchly Computer Company 1946
  - **UNIVAC**: first ***commercial*** general-purpose computer, delivered to US Census Bureau by Remington-Rand, 1951.

Post-War

- Post-war: 1940s and early 50s, continued...
- Would you like a pastry with that computer?
  - J. Lyons Co., purveyors of tea & pastry, want a computer to streamline operations. Problem: none to buy in London.
  - Make their own (with Cambridge): **LEO**: Lyons Electronic Office
  - Others interested, add to product line

- IBM notices threat to empire, enters
- computer market with the IBM 650.
  - disadvantage: slow
  - advantage: IBM sales force
  - 1000 sold within a year
  - used punched cards



Problems!

- Late 1950s, into 1960s
- Growing problem: SOFTWARE!

- Programming in machine language:
  - 0's & 1's
  - hardware specific
  - difficult & tedious to write & debug programs!
  - everyone has custom software
- Not enough programmers!
- Software costs 2-4 times the amount of the machine!

- **Compilers:** Fortran & COBOL

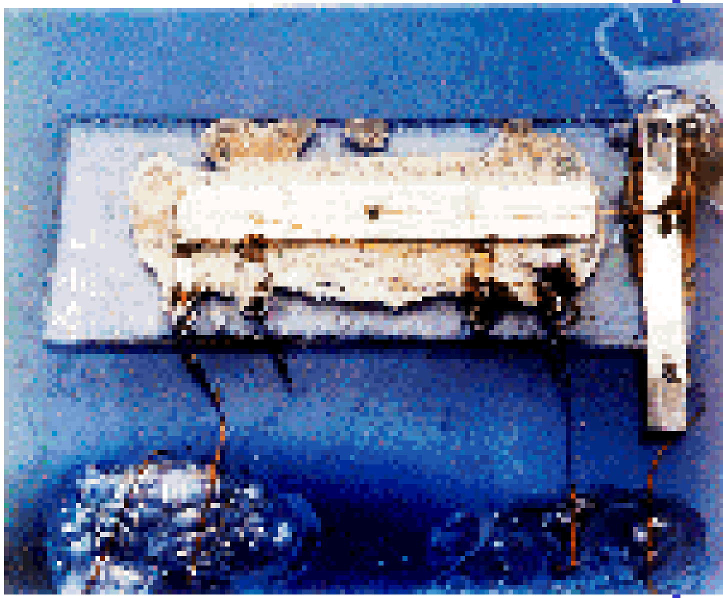
This will continue..

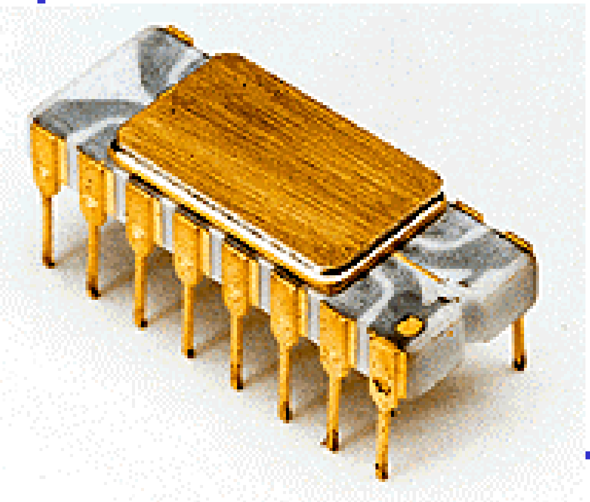
Replacing the Vacuum Tube

- Late 1950s, into 1960s
- Transistors invented in 1956
  - 50th the size of vacuum tubes
  - no heat
  - 100th weight
  - less power needed
- **New problem:** wiring the transistors together
  - “tyranny of numbers”
  - tangled mess of wires, hard to trace
- **Solution:** integrated circuit
  - silicon, altered to create transistors & other components, with layer of metal on top (which is evaporated except for connections)
  - wiring now part of manufacturing process

Smaller than ever

- 1959: first integrated circuit (IC) announced
  - Not used right away: too expensive
  - First IC cost $1000
- 1960s:
  - Drive to put man on the moon.
  - Need to fit computer in spaceship.
- 1970s and Silicon Valley
  - ICs: smaller, denser, faster, cheaper
- 1971: first “microprocessor”: Intel 4004





Altair

- Hobbyists dreamed of owning computers.

- 1975: Altair 8800, first *commercial* microcomputer.

- $395 for kit, $650 built.

- Entered program via switches on the front, LED readout in binary format.

“Modern” computing

- 1975: Microsoft licenses BASIC
- 1976: Apple Computer Company is launched…
- 1979: Apple II and Visicalc $1298
- 1979: IBM wants in…needs an OS!
  - Why not use Apple OS?
  - What transpired…
  - Bill gets rich
- 1981: IBM PC $1265
  - Soon after: PC clones.
  - Bill gets richer. **Why?**
- By 1982: IBM owns more than half of PC market. **Why?**

IBM PC

“Modern” computing

- 1984: The Macintosh. How was it different?
  - commercial: http://www.apple-history.com/1984.html
- 1990: Windows 3.0 (=heartburn for Apple Co.)
- 1995+: Win 95/98/M.E…
- 90s and TODAY:
  - Faster, cheaper, smaller…whoah!
  - Obsolescence
  - The Internet and Web
  - Networking your home …inter-connectivity

This will continue..

Advances in Technology

- Speed: doubles every 1-2 years
- Memory: doubles every 3-4 years
- Weight, Size: relatively constant except,
  for notebooks & PDAs

Moore's Law:

Gordon Moore predicted that the number of transistors per integrated circuit would double every 18 months.

Conclusions from history

- Progress is exponential in time. *How long*?
- Realization technology for computers changes permanently. *Forever*?
- Basic ideas are not technology related by mathematics and algorithm related. But new mathematics is invented and new physics (*quantum*)
- More use of biology and psychology
- Operations, algorithms, programmability, memory, flexibility, reconfigurability.
- ***Everything interesting is still ahead of us!***

Please do not forget about reading assignments from last lecture. I will not remind any more.