



*Unconventional
Computing*

§0 Introduction

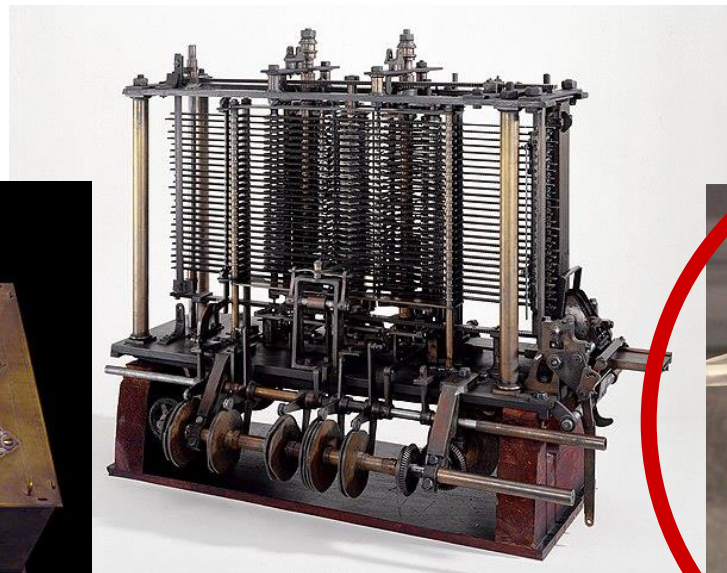
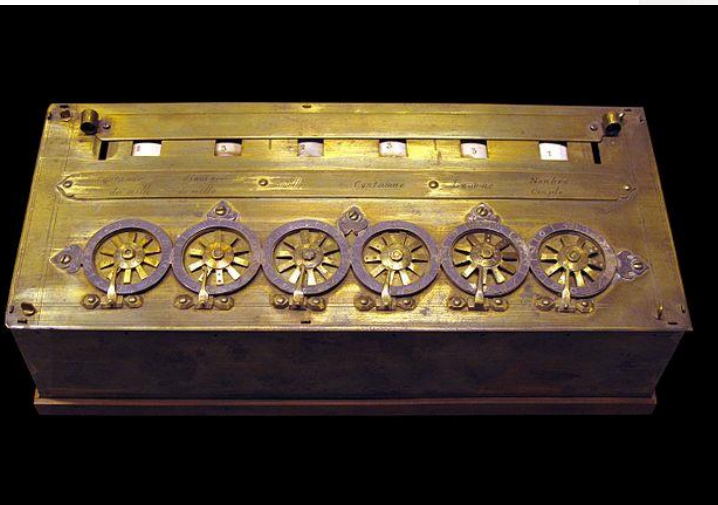
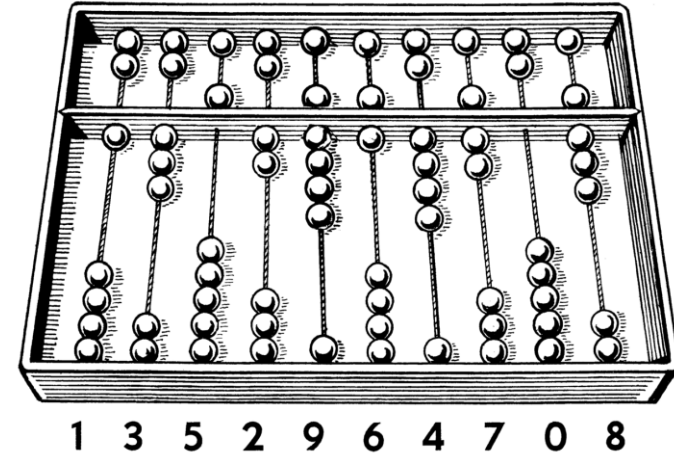
- Administrative Matters
- Student Topics
- History of Computing
- Moore's Law
- Power of Abstraction
- Principles+Models of Computing

§0 Student Topics

- Artificial Chemistry + Reaction-Diffusion + Membrane + P Computing
- DNA + Molecular + Bacterial + Cellular Computing
- Slime Mold Computing
- Reservoir Computing
- Amorphous Computing
- Social Algorithms
- Inductive Turing Machines (req. CS422!)
- Unconventional Problems

§0 History of Computing

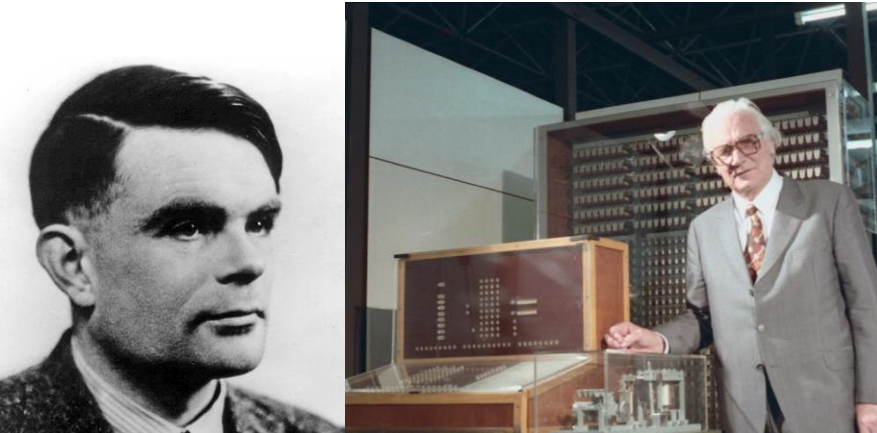
- *Abacus* (ca.2500 BC)
- *Antikythera* mechanism (ca.100 BC, solar system model)
- Blaise's *Pascaline* (1840ies)
- Babbage *Analytical Engine* ("1837")



§0 History of Computing

- 1936: Alan Turing publishes "On Computable Numbers, with application to *Entscheidungsproblem*"
- 1941: Konrad Zuse's Z3 becomes operational

- a) Define mathematical model of "digital computer"
- b) Demonstrate fundamental capabilities (*universal TM*)
- c) Prove ultimate limitations (*Halting Problem*)
- d) Guide to/wards Engineers/Implementation



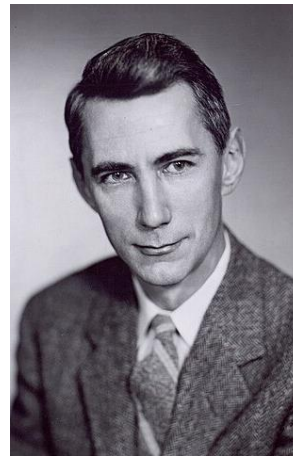
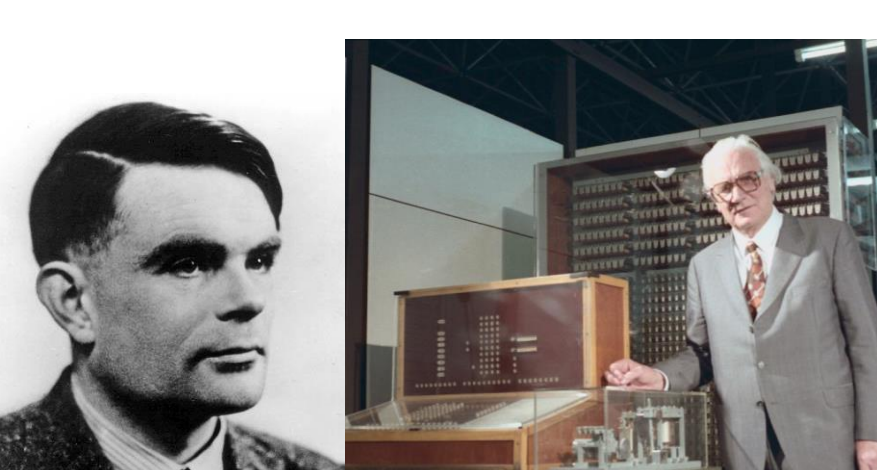
§0 History of Computing

- d) Guided by Engineers/Implementation
- a) Define mathematical model of "*analog* computer"
- b) Show fundamental capabilities (*universal GPAC*)
- c) Prove ultimate limitations (Γ Function, Riemann ζ)

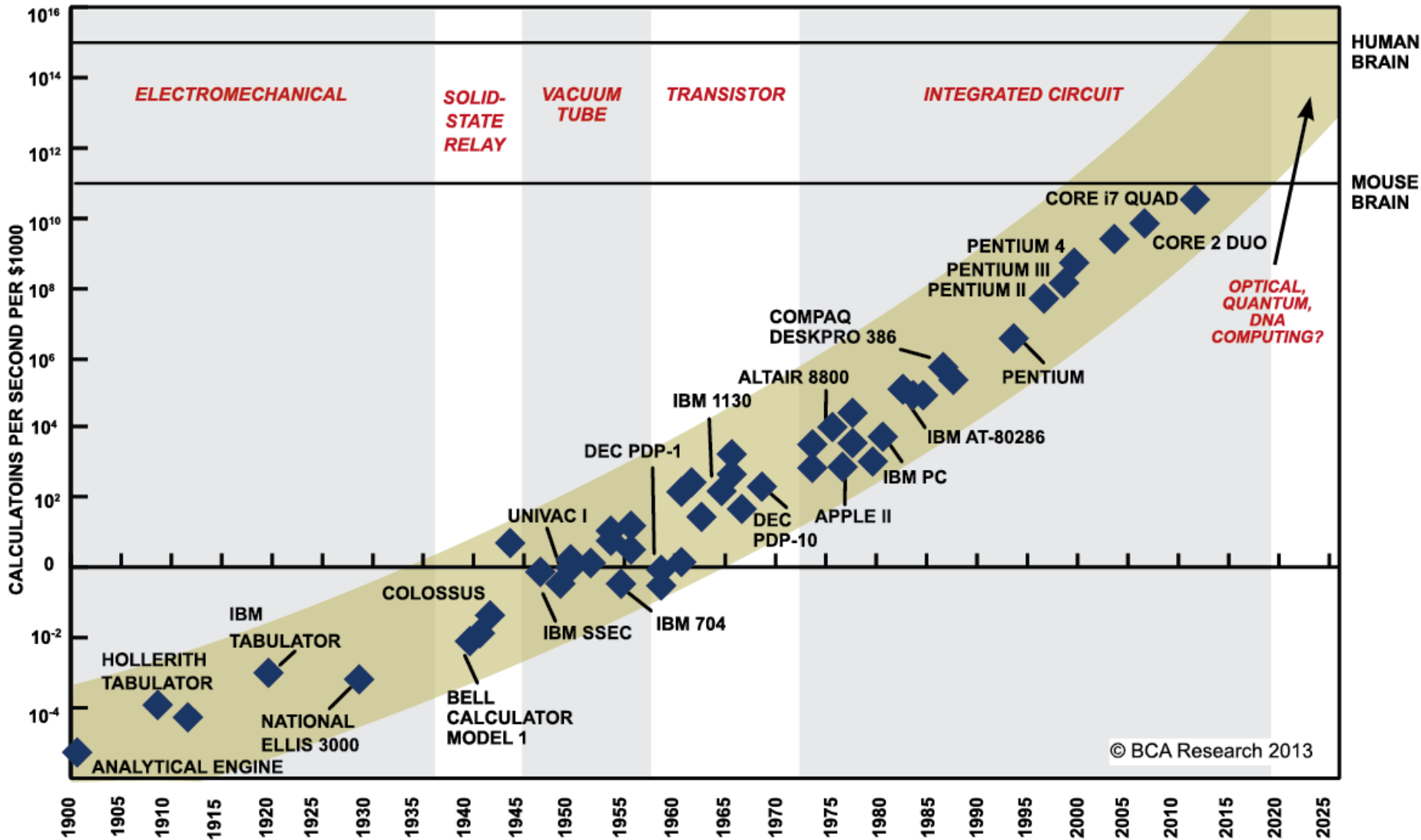
1931: Vannevar Bush's general-purpose analog computer operational at MIT

1941: Claude Shannon publishes

"Mathematical Theory of the Differential Analyzer"



§0 Moore's Law

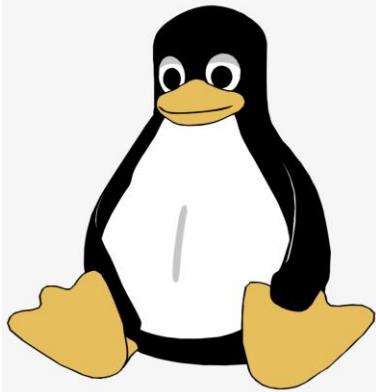


SOURCE: RAY KURZWEIL, "THE SINGULARITY IS NEAR: WHEN HUMANS TRANSCEND BIOLOGY", P.67, THE VIKING PRESS, 2006. DATAPPOINTS BETWEEN 2000 AND 2012 REPRESENT BCA ESTIMATES.

§0 Power of Abstraction

high-level
program.

obj.library

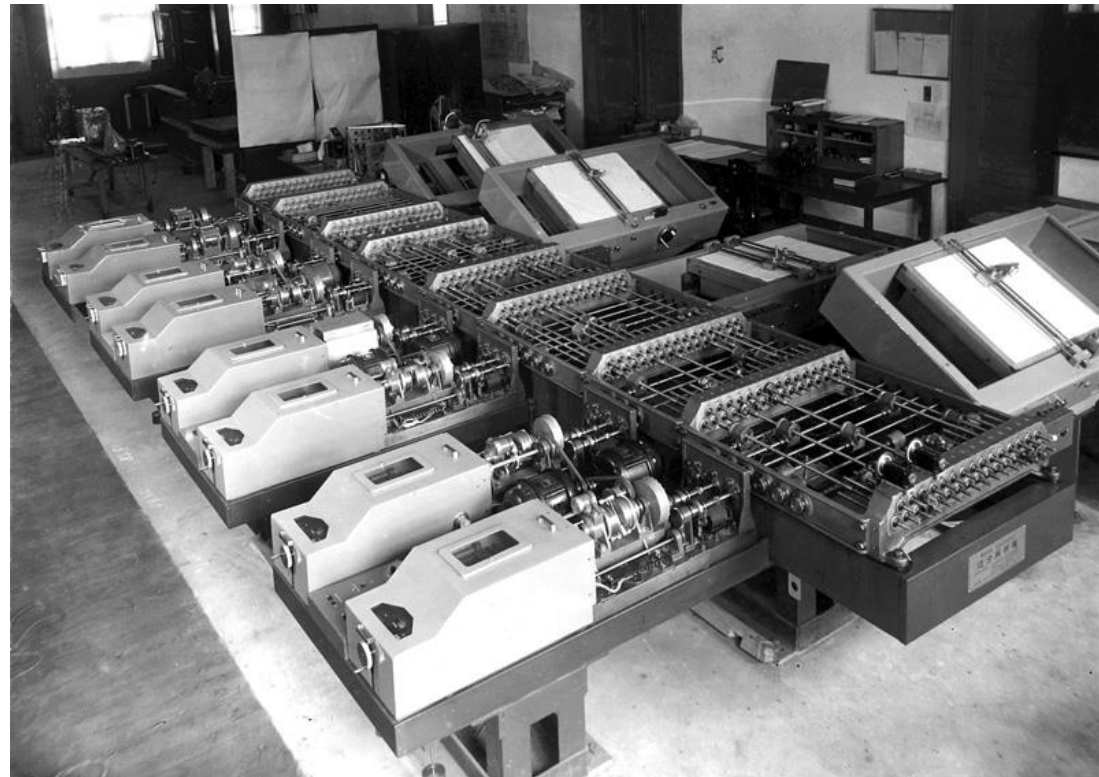
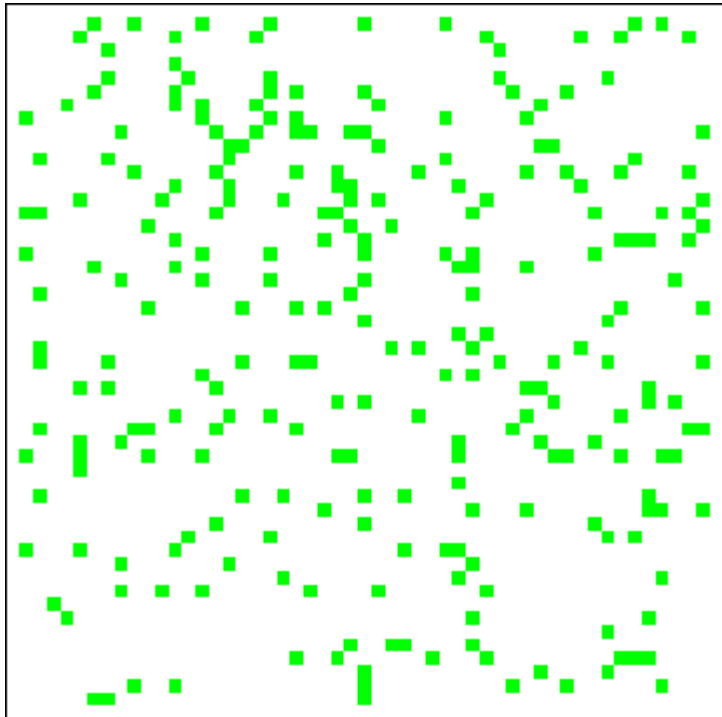


Computer
Hardware



§0 Principles of Computing

- Few and simple "*basic*" primitives
(operations, gates, cells etc.)
- Combine to realize "*complex*" behavior/functionality



§0 Model of Computing

- Few and simple "*basic*" primitives
(operations, gates, cells etc.)
- Combine to realize "*complex*" behavior/functionality

Rigorously defines: said "basic" primitives
and rules, how they can/not be combined
with resulting behavior/functionality
(*"syntax and semantics"*)

§0 Conclusion

- Administrative Matters
- Student Topics
- History of Computing
- Moore's Law
- Power of Abstraction
- Principles+Models of Computing