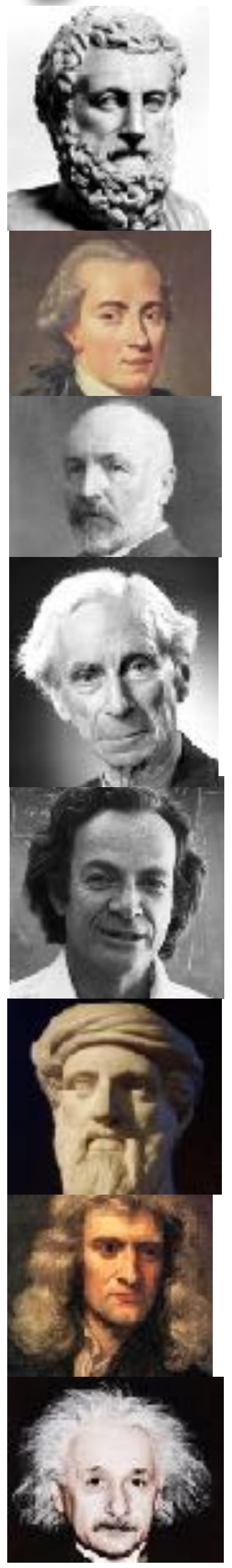
Lecture 1

Pete Manolios
Northeastern

Introductions

- Stand up
  - Background
    - Where are you from?
    - Interests
  - Why are you taking this class?
  - What do you want to get out of this class?
  - One interesting thing about you

Logic: A Short History



- Aristotle (384-322 B.C.) invented syllogistic logic
  - His “Organon” books were the foundations of logic for 2000 years.
  - Kant: *Logic ... since Aristotle … has been unable to advance a step, and thus to all appearance has reached its completion.*
  - Karl von Prantl: logicians who say anything new about logic are *“confused, stupid or perverse.”*
- Cantor in 1800’s introduced modern set theory & nailed infinity
- Russell and others found paradoxes
  - $X = \{y : y \notin y\}$. Now $X \in X$ iff $X \notin X$
  - Paradoxes are not tolerated in mathematics; proofs provide certainty
  - Contrast with physics where experiment is king. Feynman:
    - *First you guess. Don’t laugh, this is the most important step. Then you compute the consequences. Compare the consequences to experience. If it disagrees with experience, the guess is wrong. In that simple statement is the key to science. It doesn’t matter how beautiful your guess is or how smart you are or what your name is. If it disagrees with experience, it’s wrong. That’s all there is to it.*
    - Mathematics: Proof is king and provides certainty
    - Compare Pythagorean Theorem (forever) with Newtonian Mechanics (displaced by Einstein …)
- There was a crisis in foundations

Birth of Computer Science

- Hilbert's program: put mathematics on a firm foundation
  - Remove appeals to intuition in proofs
  - What does that mean?
  - There is an effective procedure to check it
  - A (Turing) machine can check it
- Many equivalent formalizations
  - Church: the lambda calculus
  - Kleene, Godel: recursive functions
- The birth of CS: what problems can computers solve? (decidability)
- What problems are unsolvable? (undecidability)

Formal Methods

- Computational systems
  - Hardware, software, systems, programs, models, requirements, …
- Describe/model computational systems using formal notations
- Reason about computational systems
  - What rules govern the behavior of systems?
  - How do we specify requirements?
  - How do we verify requirements are met?
  - How we can we predict system behavior?
- Similar to fundamental question in physics
  - How do we model the physical universe?
  - What rules govern the behavior of the physical universe?
- FM: the most fundamental way of understanding, analyzing computation

Fundamental Questions

- Basic questions in computer science
  - How do we describe computational systems?
  - How do we specify requirements?
  - What rules govern the behavior of computational systems?
  - How do we verify requirements are met?
  - What are the limits of computation?
- All of these questions are answered by logic
- Connections
  - Philosophy: are humans more powerful than computers?
  - Science: theoretical, experimental, computational
  - Society: automation (factories, self-driving cars, …)
  - Biology: DNA is a essentially a program (that we can modify)
  - Applications: aviation, crypto-currency, IoT, finance, robots, energy, …

Safety & Reliability

- Why is reasoning about computation important?
  - Safety-critical systems, embedded systems, …: loss of life
  - Power plants, critical infrastructure, …: major disruptions
  - Medical records, …: loss of privacy
  - Financial institutions, e-commerce, …: loss of money
  - The AI apocalypse: extinction?

Safety & Reliability

- Examples
  - Ariane 5, $7B rocket devel. over 10 years, explodes due to software bug: cost $2B
  - Space shuttle: testing costs $1000 per LOC
  - Functional verification accounts for >40% of total chip costs
  - Toyota found guilty by Oklahoma jury after death due to sudden acceleration (software bugs)
  - NIST: cost of software bugs is $59.5B a year
  - Gulf War: Patriot misses incoming Iraqi Scud: 28 soldiers killed; 100s injured; GAO: software bug
  - USS Vincennes, using Aegis system, shot down an Iranian aircraft with pilgrims to Mecca: 290 dead
  - People given lethal doses of radiation from Therac-25 machines (1980s) & Multidata software (2000s)
  - Knight capital lost $500M in 1/2 an hour due to bug in trading software
  - 1982 Soviet gas pipeline: CIA sabotaged trans-Siberian gas pipeline software: largest non-nuclear explosion
  - Love virus: Virus from email attachment with subject “ILOVEYOU” infected millions of computers: $9B cost

Mechanized Reasoning

Instead of debugging a program, one should prove that it meets its specifications, and this proof should be checked by a computer program.

"A Basis for a Mathematical Theory of Computation" 1961

John McCarthy

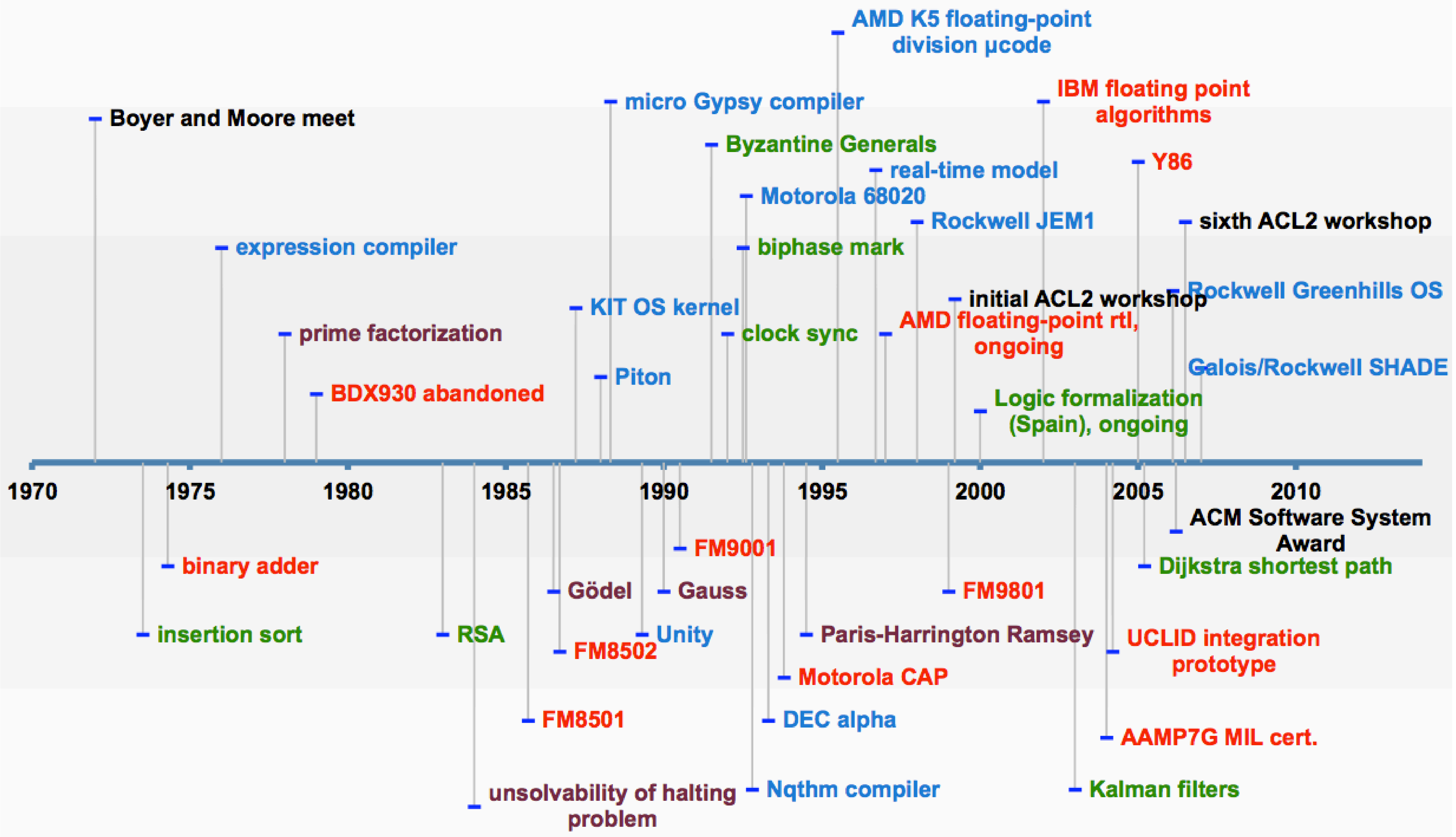
Course Webpages

Boyer-Moore Theorem Provers

- 1970’s
  - Edingurgh Pure Lisp Theorem Prover (1973)
  - A Computational Logic (1978)
- 1980’s
  - NQTHM (1981)
  - ACL2 (1989) A Computational Logic for Applicative Common Lisp
- 1990’s-Present
  - Kaufmman joins as developer
  - Workshops (16 already); huge regression suite
- 2000’s:
  - ACL2 books
  - Development environments (ACL2 Sedan)
  - 2005 ACM Software System Award (Boyer, Kaufmann, Moore)

Boyer-Moore Theorems Proved

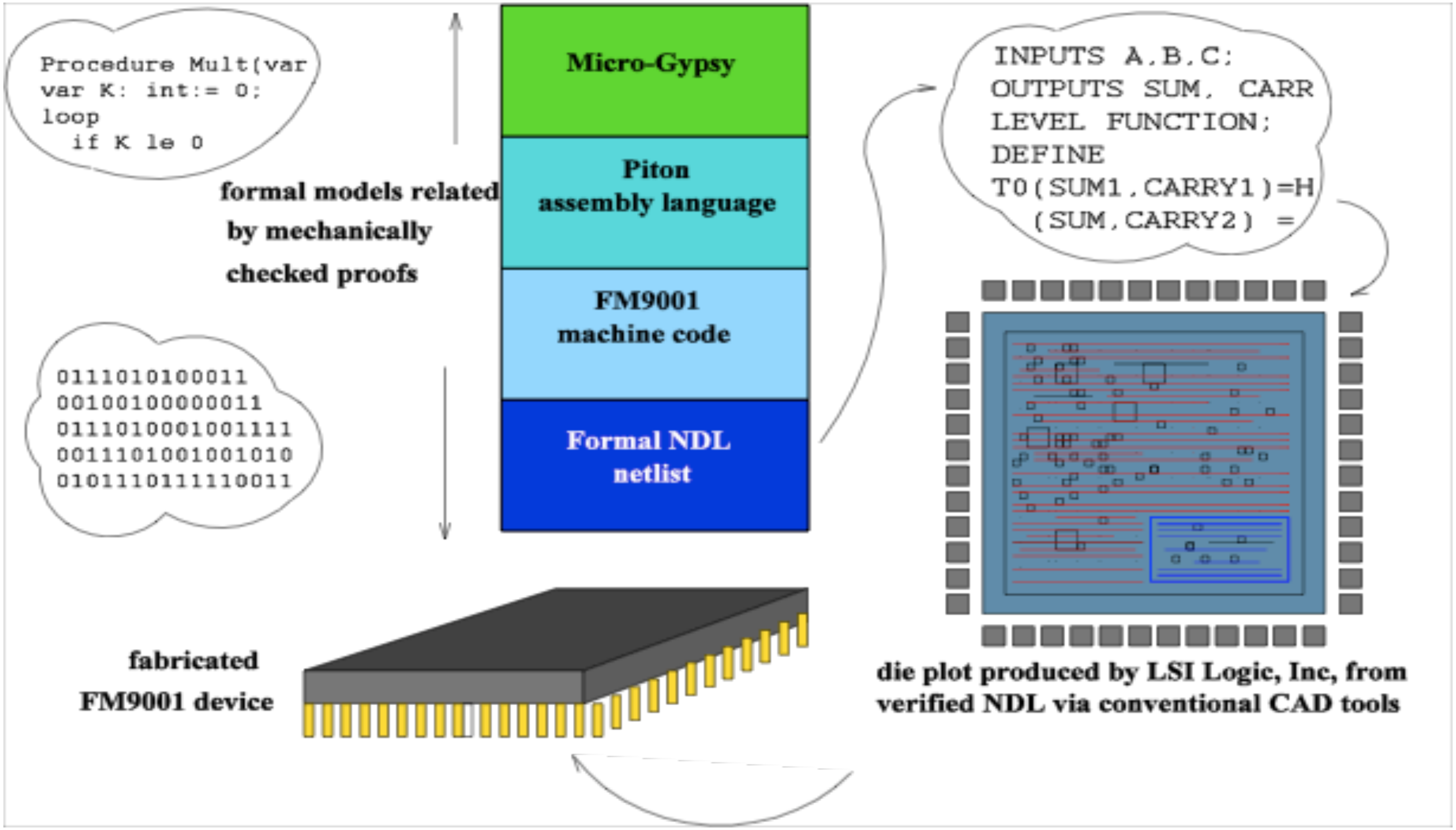
- 1970's: Simple List Processing
  - Associativity of append
  - Prime factorizations are unique
- 1980's: Academic Math & CS
  - Invertibility of RSA
  - Undecidability of halting problem
  - Gödel's First Incompleteness Theorem
  - Gauss' Law of Quadratic Reciprocity
  - CLI Stack:
    - Microprocessor
    - Assembler-linker-loader, Compiler, OS
    - High-level language



Timeline from J Moore

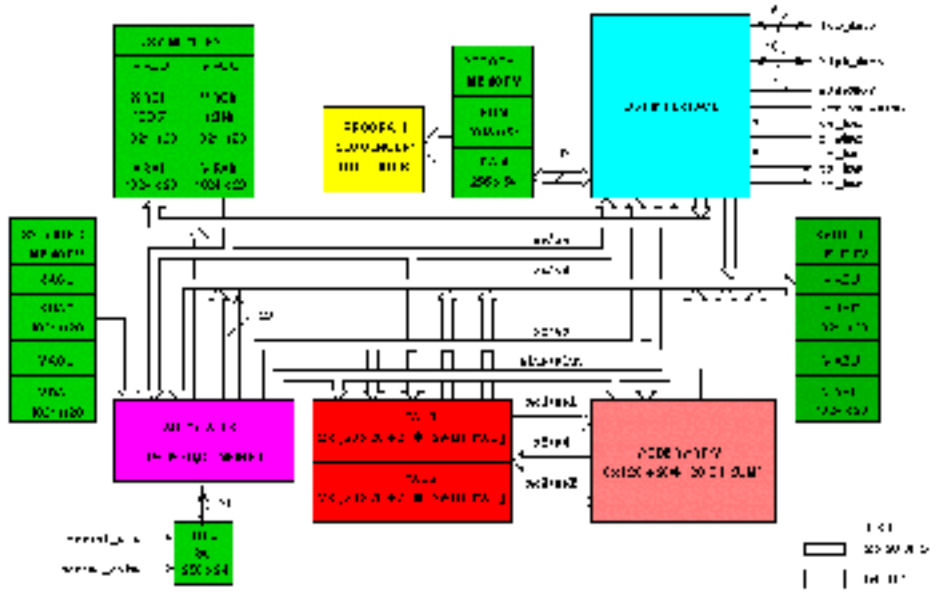
Slides by Pete Manolios for CS4820

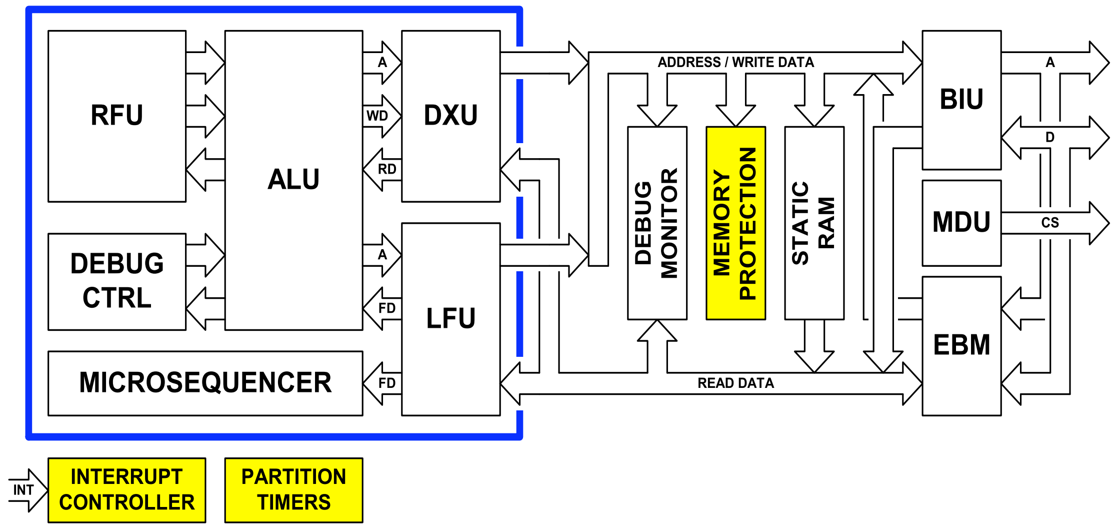
CLI Stack


Procedure Mult(var
var K: int:= 0;
loop
if K le 0
formal models related by mechanically checked proofs
Micro-Gypsy
Piton assembly language
FM9001 machine code
Formal NDL netlist
0111010100011
0010010000011
0111010001001111
0011101001001010
0101110111110011
INPUTS A,B,C;
OUTPUTS SUM, CARR
LEVEL FUNCTION;
DEFINE
T0(SUM1,CARRY1)=H
(SUM,CARRY2) =
fabricated FM9001 device
die plot produced by LSI Logic, Inc, from verified NDL via conventional CAD tools

Industrial Applications

- FDIV AMD Floating Point, IBM ...
- Motorola CAP DSP
  - Bit/cycle-accurate model
  - Run fasters than SPW model
  - Proved correctness of pipeline hazard detection in microcode
  - Verified microcode programs
- Rockwell Collins JEM1
- Rockwell Collins AAMP7
  - MILS EAL-7 certification from NSA for their crypto processor
  - Verified separation kernel
- Centaur: Media Unit





Slides by Pete Manolios for CS4820

Next Time

- Get the book!
- Send me your preferred email address, sign up in piazza
- HWK1 Due Tue night (show).
  - Install ACL2s, ACL2, emacs, SBCL
- Readings
  - RAP (Chapters 1-5)
  - Defdata paper
  - Harrison 1-2.6