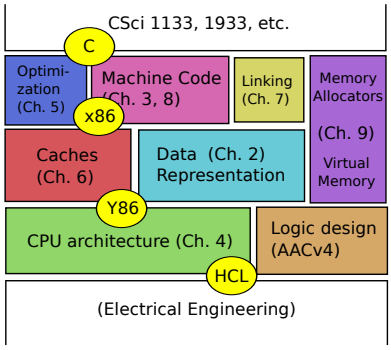
CSci 2021: Final Review Lecture

Stephen McCamant
University of Minnesota, Computer Science & Engineering

Abstraction layers (in one slide)



Implementing high-level code (1)

- Machine-level code representation
  - Instructions, operands, flags
  - Branches, jump tables, loops
  - Procedures and calling conventions
  - Arrays, structs, unions
  - 32-bit versus 64-bit
  - Buffer overflow attacks
- Code optimization
  - Machine-independent techniques
  - Instruction-level parallelism

Implementing high-level code (2)

- Linking
  - Symbols and relocation
  - Libraries, static and dynamic
- Dynamic memory allocation
  - Heap layout and algorithms
  - Garbage collection
  - C memory-usage mistakes

What hardware does

- Number representation
  - Bits and bitwise operators
  - Unsigned and signed integers
  - Floating point numbers
- Memory hierarchy and caches
  - Disk and memory technologies
  - Locality and how to use it
  - Cache parameters and operation
  - Optimizing cache usage
- Virtual memory
  - Page tables and TLBs
  - Memory permissions and sharing

Building hardware

- Logic design
  - Boolean functions and combinational circuits
  - Sequential circuits and state machines
- CPU architecture
  - Y86 instructions
  - Control logic and HCL
  - Sequential Y86
  - Pipelined Y86

Outline

Finish off state machines

Layered course overview

Post quiz 2 topics

Course evaluations

Virtual memory structures

- Pages are units of data transfer (e.g., 4KB)
  - Can be in RAM or on disk
- Page table maps virtual addresses to physical pages
  - For efficiency, use multiple levels
- A TLB is a cache for page-table entries

Virtual memory uses

- Avoid capacity limits on RAM
- Cache data from disk for speed
  - Demand paging of code
- Implement isolation between processes
  - Separate page tables
  - User/kernel protections
- Share reused data
  - Executable code, shared libraries

Memory allocation

- Data structures to represent the heap
  - Boundary tags and the implicit list
  - Explicit free list(s)
- Algorithms for heap management
  - First fit vs. best fit
  - Size segregation
- Memory errors in C code
- Alternative: garbage collection

Linking mechanics

- Symbols include functions and variables
  - Some are file-local, stack variables not even considered
- Symbols are resolved to the correct definition
  - At most one strong definition, or one of many weak ones
- Code is relocated so it runs correctly at is final address

Libraries

- Collections of reusable code
- Static libraries
  - Several `.o` files grouped together
  - Only needed files are selected
  - Copied into executable just like other object files
- Dynamic shared libraries
  - Not loaded until program startup or later
  - Single copy can be used by different programs
  - Uses position-independent code

Boolean functions

- Inputs and outputs are finite, just bits
- Can always express using minimal abstraction of gates
- Formulas transformed according to Boolean algebra rules
- Truth table is a complete representation
  - Can use for specification or equivalence checking

Combinational design

- Truth table direct to SOP: inefficient
- Karnaugh maps
  - Good for one output, up to 6 inputs
  - Power-of-two rectangles correspond to product terms
  - Look for minimal cover of large rectangles
- Bigger: use building blocks, or CAD

Logic building blocks

- Combinational:
  - En/decoders, (de)multiplexers
  - Half and full adders
  - ALUs and more complex math
- Sequential:
  - S-R latches: transparent
  - D, T, and J-K flip-flops: edge triggered
  - Registers and shift registers

State machines

- Convenient representation for systems storing a small amount of data
  - Inputs and outputs are just wires
  - States are encoded a bit patterns, e.g. binary or one-hot
  - State bits stored in flip-flops
  - State update and output are combinational functions
- Moore machines:
  - Output depends only on state
  - Output changes only sequentially
- Mealy machines:
  - Output depends on state and inputs
  - Usually need fewer states

Self-promotion

- Did you enjoy the bomb and buffer labs?
- Want to learn more about security attacks and defenses?
- Later in your studies (after 4061), consider:
- CSci 5271, Introduction to Computer Security
- Taught in the fall, recently by me

Outline

Finish off state machines

Layered course overview

Post quiz 2 topics

Course evaluations

Why are these important?

- Help us do a better job next time
- What worked well, what not so well?
- If you were running the course, what activities would you spend more or less time on?
- I will read your written comments, after grades submitted