

The diagram on the left side of the slide is set against a blue background. It consists of three circular icons arranged vertically, connected by arrows pointing upwards. The bottom icon shows a transistor circuit diagram. An arrow points from it to the middle icon, which is a purple square labeled 'CPU'. Another arrow points from the CPU icon to the top icon, which is a high-level block diagram with an orange rectangle at the top, a blue diamond in the middle, and a red rectangle at the bottom, with a green horizontal bar to the right.

Final Exam Review

Original slides from Gregory Byrd, North
Carolina State University
Modified slides by C. Wilcox, S. Rajopadhye
Colorado State University

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.

Review Topics

- Number Representation & Arithmetic
- Transistors, Gates, Combinational Logic & Logical Completeness
- Sequential Circuits & Finite State Machines
- LC-3 Architecture & Instruction Set
- Programming & LC-3 Assembly
- Subroutines, I/O, Traps & Interrupts
- Stack, Parameter Passing & Activation records
- Assembly process, Symbol table & compilation
- C (control, pointers, data structures, functions, parameter passing, memory model)

2

Number Representation & Arithmetic

- Conversion, representation and operations
 - decimal to binary
 - binary to decimal
 - octal, hex and other radices
 - Sign-magnitude, 2's complement
 - floating point
 - Operations on numbers
 - Add, subtract, bit-wise
 - Add/subtract two floating point numbers without first converting to decimal and then back.

Transistors, gates, combinational logic

- Transistor as a switch & switch networks
- complementary MOS
- Gates & logical completeness
- Design simple combinational circuits
- Compose/hook them together to build larger circuits
 - “fat” gates, decoders, muxes, demuxes
- Sequential circuits
 - Registers & memory
 - FSM

LC-3

- von-Neuman architecture (stored program m/c)
- LC-3 data path: don't memorize, but **understand** what each part does
 - Instruction register, PC, register file, MAR/MDR, status bits
 - Muxes, ALU, incrementers, Zext, Sext
 - Memory & busses
- Instruction set (don't memorize but understand)
- Addressing modes

Programming & Assembly

- Programs & data as binary/hex instructions
 - What does a small snippet of code do?
 - e.g., problems from HW3
- Manually disassemble a few instructions
- Assemble a snippet of code
 - Address calculation and symbol table
- Programming
 - Control flow and logic, condition codes and branch
 - Jump and subroutines

Subroutines, I/O, Traps & Interrupts

- Call-return mechanism and return address
- Parameter passing, saving/restoring registers
- Traps and I/O service routines
- Memory mapped I/O: device registers & polling
- Interrupt driven I/O
 - Generation of Interrupt signals
 - Handling the interrupt (in the machine controller)
 - Priority, Processor Status Register
 - Interrupt Service & RTI
 - Nested interrupts

Stacks, Parameters, Activation Record

- Stack Pointer
- Push and Pop operations
- Where does the stack live in memory
- How does it grow
- Functions in C (parameter passing)
- Activation record
 - Return address
 - Return value
 - Caller's frame pointer

The assembly process

- Two pass assembly
- Symbol table
- Generating the machine code

C programming

- Variables, operators, expressions & statements
- lexical scope rules
- Static and dynamic variables
- C memory model – where variables are stored
 - global, heap & stack organization of memory
- Control structures (conditionals and loops)
- Functions
- Pointers & Arrays
- Recursion & parameter passing (back to activation records)