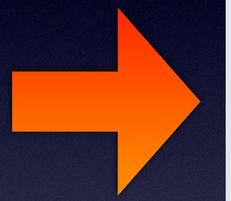
CmpSci 335

Inside the Box: How Computers Work

Chip Weems
weems@cs
https://people.cs.umass.edu/~weems/homepage/courses/index.html

Office hours: CS342, M 10:30 - 11:45
More will be added. Appointments welcome!

On the Site:



CmpSci 335
Inside the Box

Course Notes

Keep in mind that these are just a brief summary of what was covered each day, together with the lecture slides and any handouts or special announcements. They are not meant to be a complete record of everything mentioned in class.

Syllabus

Course description, grading, project info, general homework policies, exams, office hours, reading materials

Schedule

Reading assignments, outline of subject matter, project due dates, exam dates, holidays

In addition to Moodle

Course Sequence

- CmpSci 335: Digital logic, computer organization, microcontroller programming (C++ & assembly), introduction to architecture
- CmpSci 535: Architecture, including caches, pipelines, branch predictors, I/O, parallel
- CmpSci 635: Modern Architecture, advanced performance features, current hot topics

Inclusiveness Statement

No matter who you are or how you define yourself you are welcome in this class. Each person here is a human being deserving of dignity and respect. My goal is to help you learn the subject matter in a way that you will find useful, and to help you have an enjoyable and empowering experience in doing so. It is important to keep in mind that we are all coming to this class with different backgrounds. For many, this is a first hardware course, while others have some prior experience. We are all here to learn together! There are no dumb questions! From time to time, I may enlist some students to help others in class. If I ask you to help, remember that we all have different modes of learning, and there is no stigma to be associated with needing assistance. Please reach out to me if you have any concerns.

Course Goals

- Dive down beneath the abstractions
- Understand hardware and architecture from the bottom up
- Start from transistors and device physics
- See how gates and memory are constructed
- Logic level design of main components
- Machine organization and assembly language programming for microcontrollers
- C programming and C-assembly interface
- I/O device interfacing

Course Meta-Goals

- Dispel the magic
- Understand principles behind abstractions
- Appreciate limitations of hardware
- Basic mechanisms of compiler and OS operation
- Have fun with microcontrollers (biggest segment of computer industry applications)

Survey

- In Lecture 1 on Moodle — Not a test (other than of my ability to make a survey using Moodle)
- Get a sense of what you already know
- Guidance for adjusting course coverage
- Take 8 minutes to do now

8 minutes

Grading

- Digital logic labs 16%, assembly language programming labs: 12%
- Term project 16% plus project proposal 2% and presentation 4%
- Participation in class exercises 50%
- No exams - class exercises take their place in grading this term

Final Exam Period

- Used for project presentations to class
- 3-minute demo video with voice-over, explaining operation and what went into/was learned in development
- Everyone will watch and rate for impressiveness
  - Scale of “pretty cool” to “totally amazing”
  - Small part of project grade

Homework

- Let me know in advance if you need an alternate due date because of a conflict - better to submit for partial credit than late
- Logic labs are meant to help you see how machines are built
- Assembly labs help you appreciate the lowest software layer of abstraction
  - Will be due a week after they are given
- Putting it into practice: semester project proposal and project

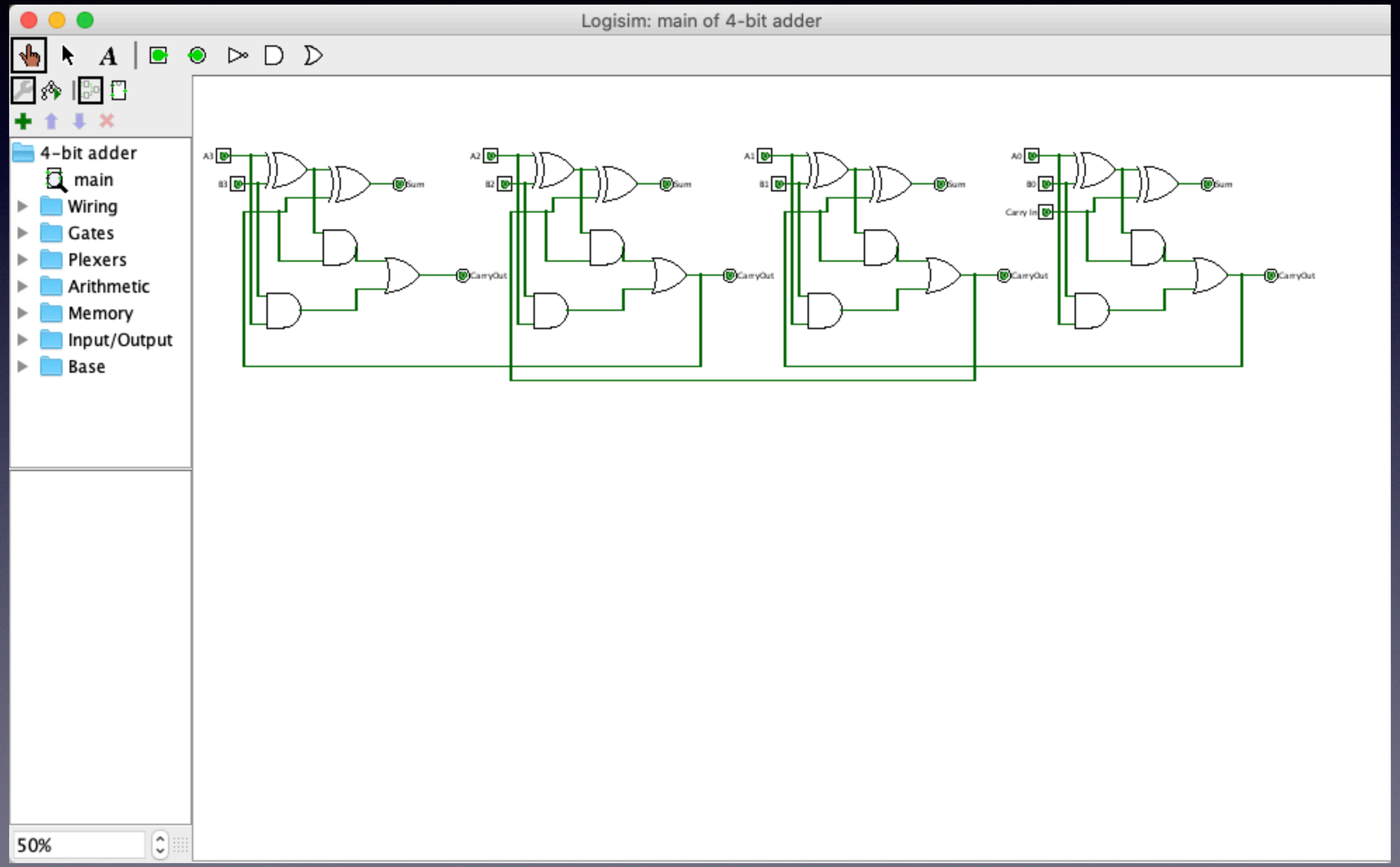
Lab HW Philosophy

- Goal is for you to get experience in working with various tools to see how machines work at the lowest levels, on which all software is built
- Discovery oriented, with skill development
- Designed to be an individual experience
  - If asked for help, take role of a teacher, rather than giving answers
- Taking initiative to explore further on your own earns extra credit

Digital Logic Labs

- Four Digital Logic Labs
  - Using Logisim to build circuits up to a processor datapath
  - Step-by-step instructions, assignments build on each other
  - Fill in answers on worksheet, attach screenshots of work
  - Some will be started in class or converted to hardware in class

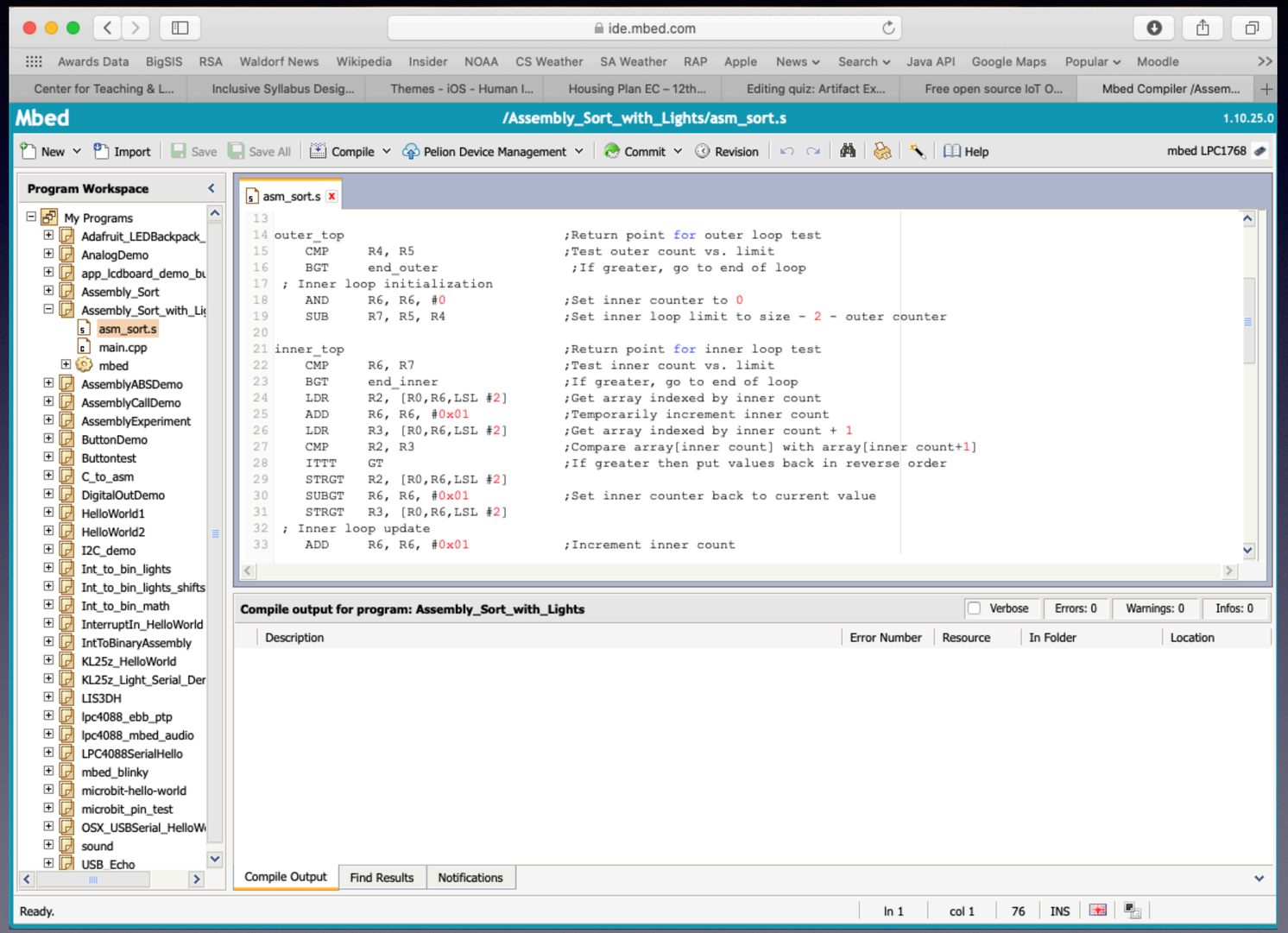
Logisim


Logisim: main of 4-bit adder
4-bit adder
main
Wiring
Gates
Plexers
Arithmetic
Memory
Input/Output
Base
50%
A3
B3
Sum
CarryOut
A2
B2
Sum
CarryOut
A1
B1
Sum
CarryOut
A0
B0
Carry In
Sum
CarryOut

Assembly Programming Labs

- Three C++/Assembly Labs
  - Solidify the concepts of the hardware/software interface
  - Many pieces will be covered in class
    - Just need to generalize/adapt, connect together
  - Can also be preparation for the semester project
  - Submit video demo of project working, with code

MBED Environment



In-Class Exercises

- Some will be small group discussions (submit shared Google doc with notes)
- Some will be homework preparation, experimentation with board
  - Screenshot, photo, or video submissions
- Some will be worksheets (sometimes done in small groups)

Semester Project

- Goal is for you to go beyond what we can cover in class
- Discovery and initiative are key aspects
- Teams of two collaborate on this
  - Best if there is a clear division of labor
- Can be individual work if you have prior experience

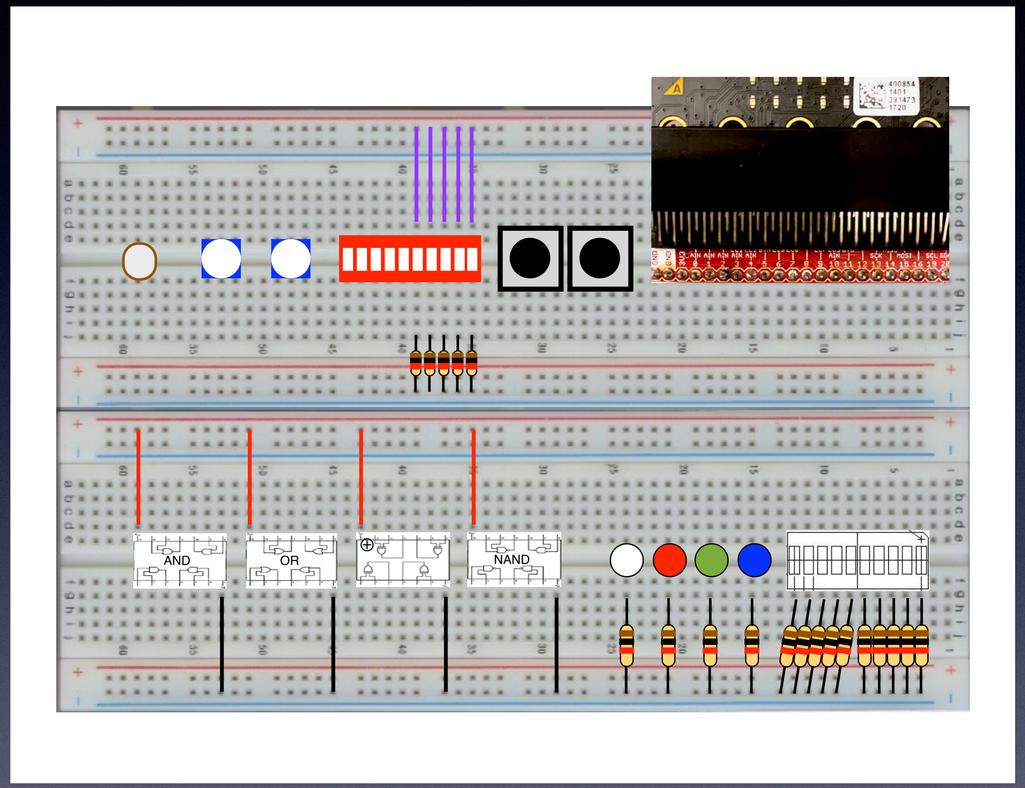
Semester Project

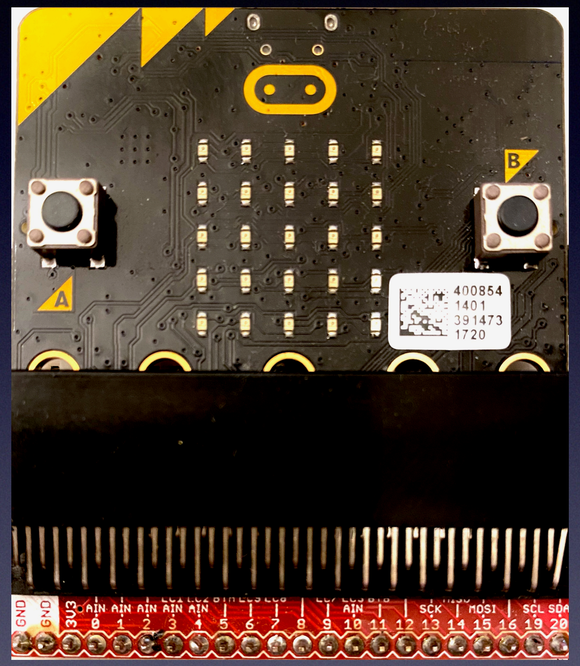
- Self-defined — entirely your choice
- Use the microcontroller kit to build something you think will impress the rest of the class
- You can buy additional small components, e.g., Adafruit, Sparkfun, etc.
- Evaluated on depth of learning demonstrated (how it takes advantage of microcontroller features, complexity, etc.), how well it fulfills its goals, effort involved, impressiveness

Semester Project

- Getting a basic project working well is B-level work
- Taking initiative to explore further is the key to A-level work
- OK to use code from other sources
  - If you clearly identify it and cite the source
  - In that case, it's also important to identify the code that is yours
    - Document changes you make to imported code too
- The final demo will be a 3-minute video presentation to the class
  - Class will rate impressiveness (small portion of project grade)

Microcontroller Kits





BBC Micro:Bit microcontroller on prototyping board with switches, lights, light sensor, variable resistors, logic chips. Micro:Bit has 5x5 LED matrix, buttons, gyro, accelerometer, temperature sensor, bluetooth, analog and digital inputs

Microcontroller Kits

Usually loaned out, with more powerful controller and more components

Because of remote learning, the kit has been redesigned to reduce cost,
and you need to purchase and assemble it

Order from Circuit Specialists for $78
https://www.circuitspecialists.com/umass_compsci_335_lab_kit.html

Order soon – we'll start using it in the 6th class,
and you'll need some time to build it before we use it in class.
Kit building guide videos will be up on Moodle later this week

Other Hardware Requirements

- Need a machine with a USB port (to connect with Micro:Bit), running Linux, MacOS, or Windows
- Need a standard browser (to use ARM MBED cloud IDE)
- Need to be able to run Java (for Logisim freeware)
- Need to be able to take photos/videos of kit (for demo submissions)
  - e.g., cell phone camera

Optional Book

- Computer Organization and Architecture, Themes and Variations, Alan Clements, ©2014, Cengage Learning
- Reading schedule
  - Lectures give you the highlights, book fills in background and details, with more extensive explanation

Computing Technology

Breakout Group Discussion using resources on Moodle

- Examine historical artifacts and descriptions in Artifacts file
- Discuss with your group and order them in time
- Use Artifact Exercise quiz to record your ordering
- Second question in quiz asks for names of your group members
- Note your answers for discussion

[redacted] minutes

Discussion/Questions