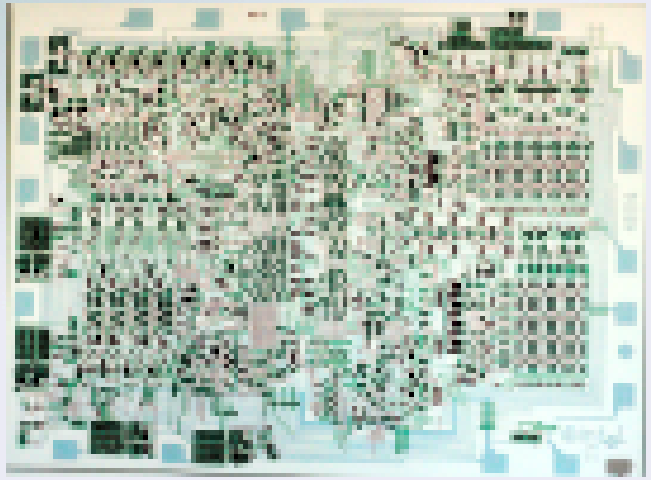

# Lecture 5b: Programming: What, Where, How

CSIS11: **Computer Architectures and Organization**

- **What** - *What do you want to solve, learn, develop?*
- **Where** - *Where do you want to do this (Windows, all Platforms, macOS, Uno, bare metal)?*
- **How** - *How are you going to get this done (tool-chain)?*
- **Why** - *is typically self-determined (job, school, **fun**)*

# What

(or *What am I going to do?*)

- *What* and *Why* go hand-in-hand
- *Why* might be "work", "learn", "school" etc
- Which in turns, drives *What*:
  - **work -** enterprise application, research notebook
  - **fun -** a *wearable*, or *IOT*, a game
  - **school -** an assignment, class project

## For this class

- *Why* - because your grade depends on it
- *What* - Simple Math Calculator (*SMC*) on *LC-3*

## Where

(or *Where am I going to do it?*)

- **Work** - Windows or macOS, Cloud application
- **fun** - embedded microcontroller, Windows/Steam, Raspberry Pi
- **MPC** - Windows or your computer

## For this class

- *Where* - **LC-3 Simulator**

# How

(or *How am I going to do it? Tool-chain*)

- **Work** - *Git/GitHub*, VS Code, *Markdown*, *React* or *PHP/SQL* or *Java* or *C++* or *Python*
- **fun** - *Git/GitHub*, *VS Code*, *Markdown*, *Arduino*, *CircuitPython*
- **MPC** - *Java*, *Python*, *C/C++*

## For this class

- *How* - *VS Code*, *Markdown*, *LC-3 Tools*, *Git/GitHub*

# Simple Math Calculator (*SMC*)

- **Add**
- **Subtract**
- **Multiply**
- **Divide**

## Iterative Development

1. Develop math routines using registers
2. Show output via console
3. Request input via console
4. Error checking

# LC-3 Simulator

- 15 *opcodes* or commands, with only **ADD**, **AND** and **NOT** as operators
- *ADD* exists - use for **Add**
- **Subtract**?
- **Multiply**?
- **Divide**?
- **Console In** - Trap instructions, however *ASCII-based*
- **Console Out** - Trap instructions, however *ASCII-based*

## Tool-chain

- LC-3 Tools - download and install
- *VS Code* - download and install
- *VS Code* [LC-3 Extension] (qili.vscode-lc3) install via VS Code Extensions

*Will be installed on MPC PC's by Monday next week.*

## How to Begin?

Programming a new project is an iterative task:

1. Start with small tasks, **completed successfully**.
2. **Focus on building confidence** and muscle memory as to how to develop code
3. If something doesn't work, figure out why and repeat until its always right. *Many bugs are caused by sloppy initial work, which manifest more bugs, later.*

## Go Slow, to, Go Fast

## Start with VS Code

Main reasons for us using it:

1. Has an extension for LC-3
2. Free
3. Every student needs to be familiar with it

# Demo of VS Studio Code

## Use Markdown for Documentation

- *Markdown* is the lingua franca of modern, technical, development documentation.
- It is used in all proper README files and is considered the standard for most informal documentation
- **It will be the expected format in this class for documentation outside of the program file**

# Markdown Overview

- Simplified HTML formatting
- Easy to read
- Easy to use and understand
- Simple formatting characters which don't obstruct meaning

# Demonstration of Using Markdown

## Git/GitHub

1. Using *Git* allows you to keep versions of your code
2. It also acts as a distribution format for code
3. *GitHub* is invaluable for distributing software and for backing up your own code

# Demo of Git and GitHub

## LC-3 Simulator

- The *LC-3* Simulator will be our computer for assembly language
- It's value is that it's simplicity enables us to learn quickly
- It is logically complete with AND and NOT
- With ADD, we are able to solve most math equations

**However, it requires a significant amount of code to perform very seemingly simple tasks.**

*I see this as a positive.*

# Demo of LC-3 Simulator *hello_world*