

CSE 141L: Introduction to Computer Architecture Lab

Implementing

Pat Pannuto, UC San Diego

ppannuto@ucsd.edu

Milestone 1 is due in 7 days

- What to submit?
 - SOMETHING
- M1 is graded for completion, not accuracy
 - The purpose of milestones is to *help you* manage large, long-term project
 - TAs will use gradescope “grades” to help give feedback
 - Recall: Only Milestone 4 (final submission) is actual grade*
 - *With exceptions for things such as skipping milestones altogether

Today's Objectives:

What does implementing look like?

- Some tips for setting up CloudLabs
- Exploring the `basic_proc` example
- Exploring ModelSim/Quarta and Quartus

n.b. the rest of these slides not presented, but here as a reference

The code editors in ModelSim/Quarta/Quartus aren't great

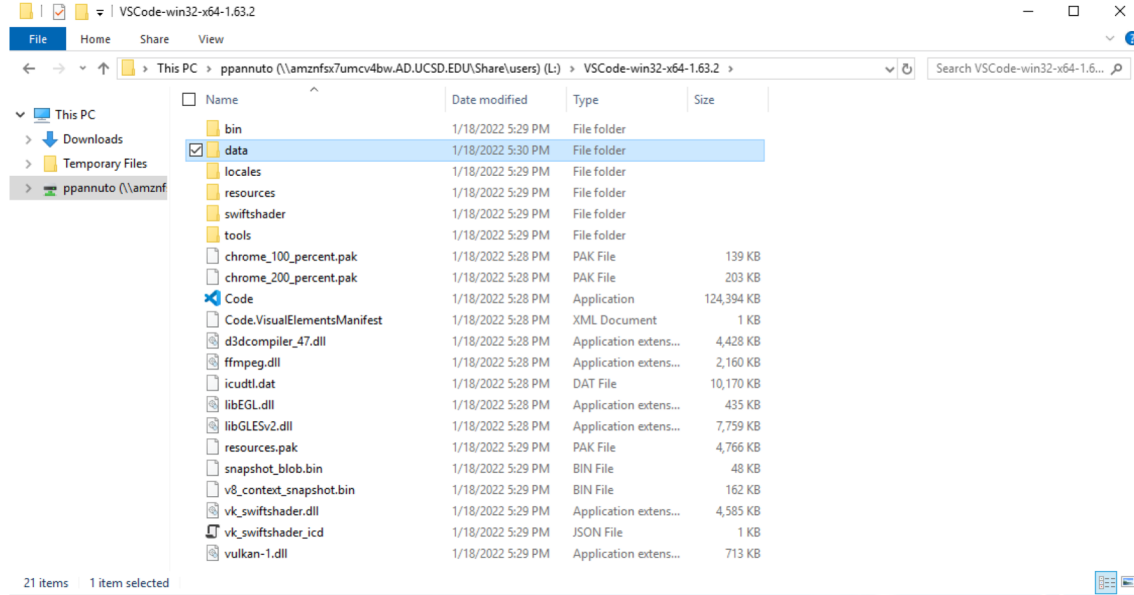
- You can use them, but more modern editors have some helpful tools
- Especially around version control
 - Because you are using version control, right?
 - And I don't mean copying folders `milestone1`, `m1_working`, `m1_real`, `m1_final`, ...

CloudLabs – What’s ephemeral and what’s not

- Spins up a ‘fresh machine’ each login
- But you have a shared L : / drive that persists across sessions
 - So put stuff there
 - But also maybe don’t trust it 100%
 - [What fixes this? **Version control of course!**]
- Take advantage of “portable apps”
 - <https://code.visualstudio.com/docs/editor/portable>
 - <https://git-scm.com/download/win> – Choose “portable” and install in L : /

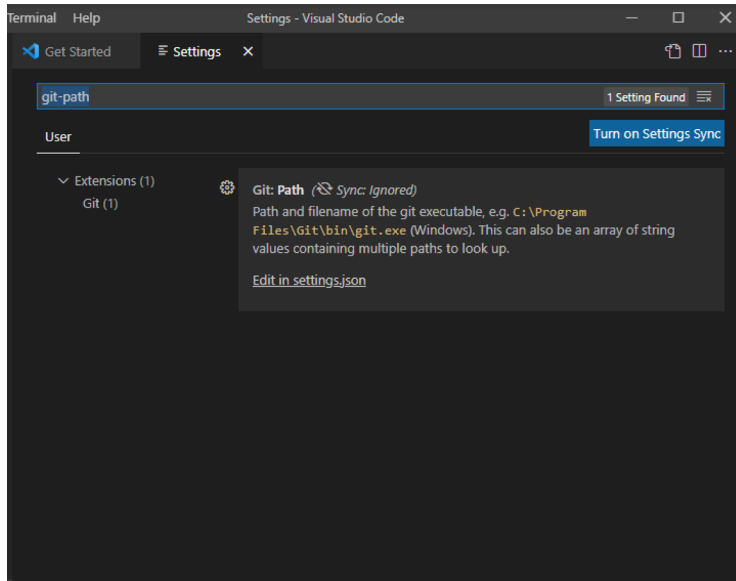
Portable VSCode

- Download the ****zip****, not the installer
- Unzip the folder in your **L:/** drive, and make a folder called **data**



If using portable apps, have to point VSCode to git

- File->Preferences->Settings; search `git-path`; write this json:

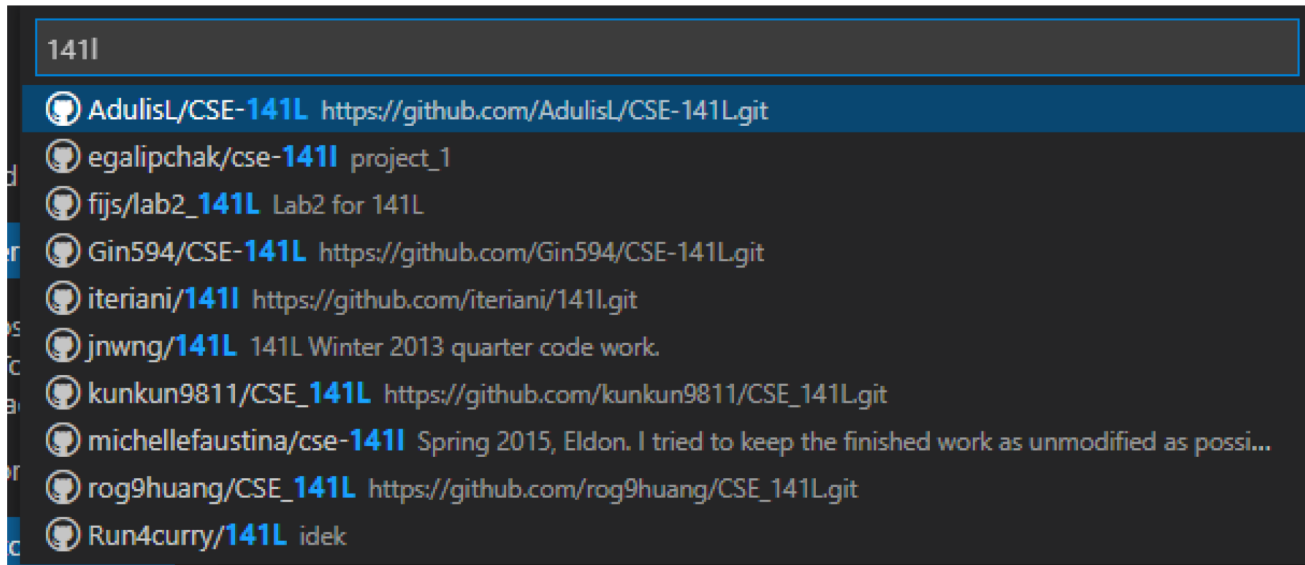


```
Get Started Settings settings.json
L: > VSCode-win32-x64-1.63.2 > data > user-data > User > {} settings.json > ...
1 {
2   // Is git enabled
3   "git.enabled": true,
4
5   // Path to the git executable
6   "git.path": "L:\\PortableGit\\bin\\git.exe"
7
8   // other settings
9 }
```

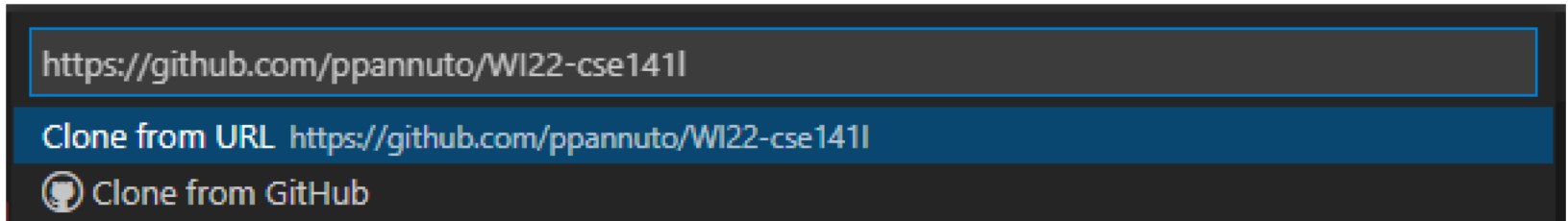
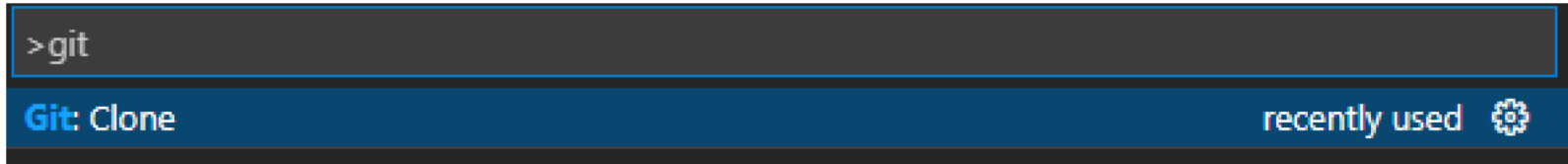
```
{
  "git.enabled": true,
  "git.path": "L:\\YOUR_FOLDER_NAME\\bin\\git.exe"
}
```

Quick reminder: No public repos please

- Q: What's really clear and indisputable evidence of sharing code?
- A: A publicly accessible repository

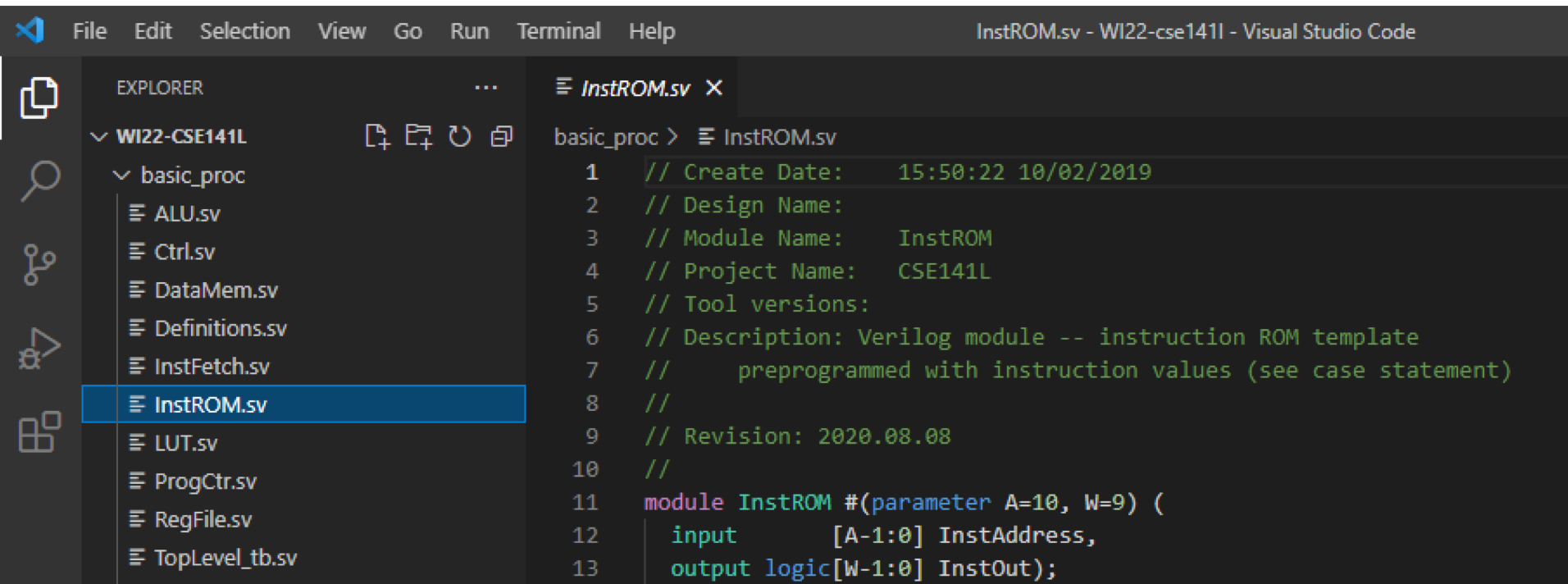


The “github” built-in won’t find private repos, so you have to type the URL by hand



- You can use the default ‘credential helper’ (“manager-core”); authorize it to github; and then it will clone

Now you're set up to code!



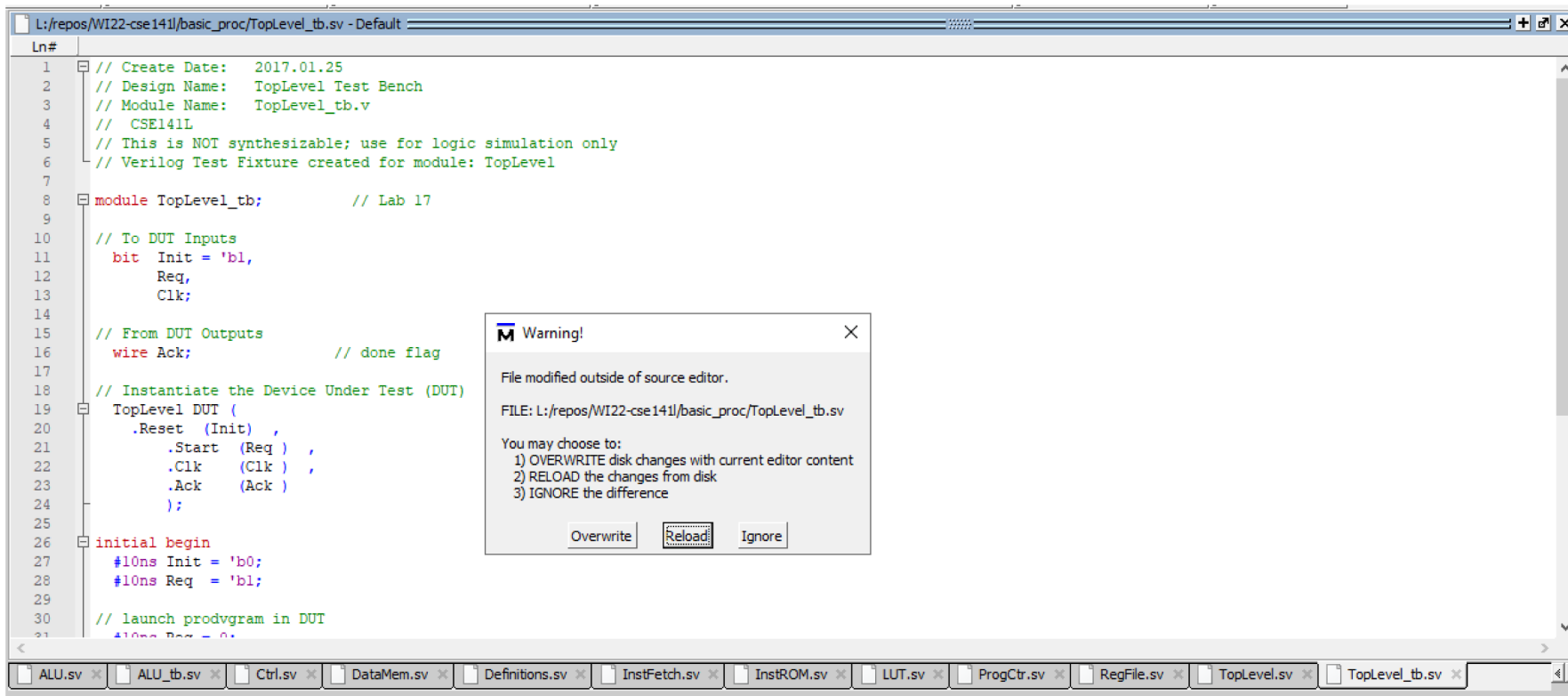
The screenshot shows the Visual Studio Code interface. The Explorer panel on the left displays the project structure for 'WI22-CSE141L', with 'basic_proc' expanded and 'InstROM.sv' selected. The main editor window shows the code for 'InstROM.sv'.

```
File Edit Selection View Go Run Terminal Help InstROM.sv - WI22-cse141l - Visual Studio Code

EXPLORER
WI22-CSE141L
  basic_proc
    ALU.sv
    Ctrl.sv
    DataMem.sv
    Definitions.sv
    InstFetch.sv
    InstROM.sv
    LUT.sv
    ProgCtr.sv
    RegFile.sv
    TopLevel_tb.sv

basic_proc > InstROM.sv
1 // Create Date: 15:50:22 10/02/2019
2 // Design Name:
3 // Module Name: InstROM
4 // Project Name: CSE141L
5 // Tool versions:
6 // Description: Verilog module -- instruction ROM template
7 //     preprogrammed with instruction values (see case statement)
8 //
9 // Revision: 2020.08.08
10 //
11 module InstROM #(parameter A=10, W=9) (
12     input [A-1:0] InstAddress,
13     output logic[W-1:0] InstOut);
```

Careful, you (can) have two editors open



The screenshot shows a Verilog source file editor with a warning dialog box overlaid. The dialog box contains the following text:

```
Warning!  
File modified outside of source editor.  
FILE: L:/repos/WI22-cse141/basic_proc/TopLevel_tb.sv  
You may choose to:  
1) OVERWRITE disk changes with current editor content  
2) RELOAD the changes from disk  
3) IGNORE the difference  
[Overwrite] [Reload] [Ignore]
```

The background code in the editor is as follows:

```
Ln#  
1 // Create Date: 2017.01.25  
2 // Design Name: TopLevel Test Bench  
3 // Module Name: TopLevel_tb.v  
4 // CSE141L  
5 // This is NOT synthesizable; use for logic simulation only  
6 // Verilog Test Fixture created for module: TopLevel  
7  
8 module TopLevel_tb; // Lab 17  
9  
10 // To DUT Inputs  
11 bit Init = 'b1,  
12 Req,  
13 Clk;  
14  
15 // From DUT Outputs  
16 wire Ack; // done flag  
17  
18 // Instantiate the Device Under Test (DUT)  
19 TopLevel DUT (  
20 .Reset (Init) ,  
21 .Start (Req) ,  
22 .Clk (Clk) ,  
23 .Ack (Ack) ,  
24 );  
25  
26 initial begin  
27 #10ns Init = 'b0;  
28 #10ns Req = 'b1;  
29  
30 // launch prodvgram in DUT  
31 #10ns Req = 0;
```

[Demo Plan]

- Create new project from basic_proc
- Compile
 - Show warnings; explore ALU 1 vs 1'b1
- Go to waveform viewer
 - Show results
 - Work examples / take Q's