CSE 390B, Spring 2022

Building Academic Success Through Bottom-Up Computing

Course Wrap-up & TA-led Activities

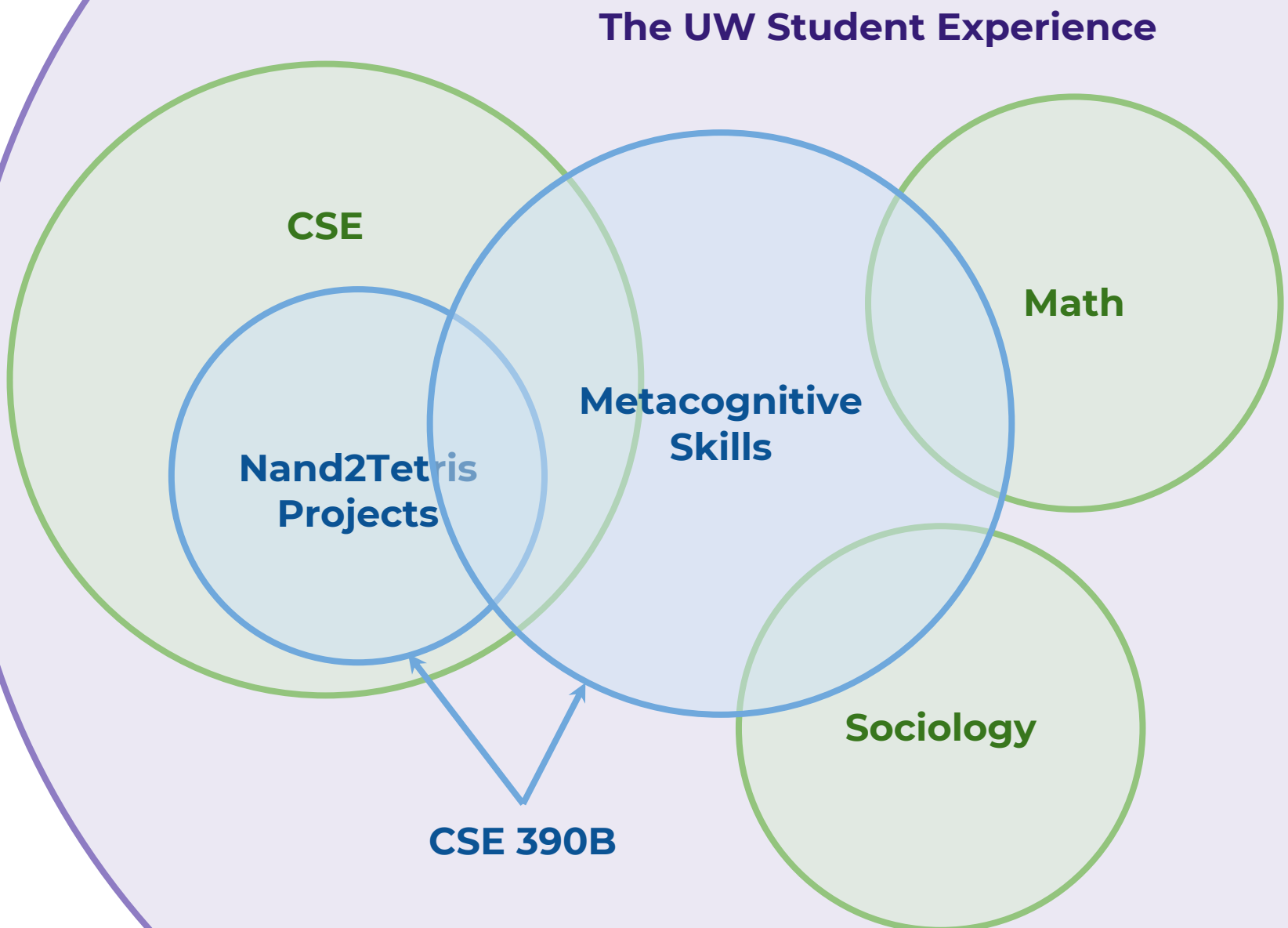
CSE 390B Victory Lap, End of Quarter TA-led Activities

Lecture Outline

- **CSE 390B Victory Lap**
  - **Nand2Tetris Projects and Metacognitive Subjects**

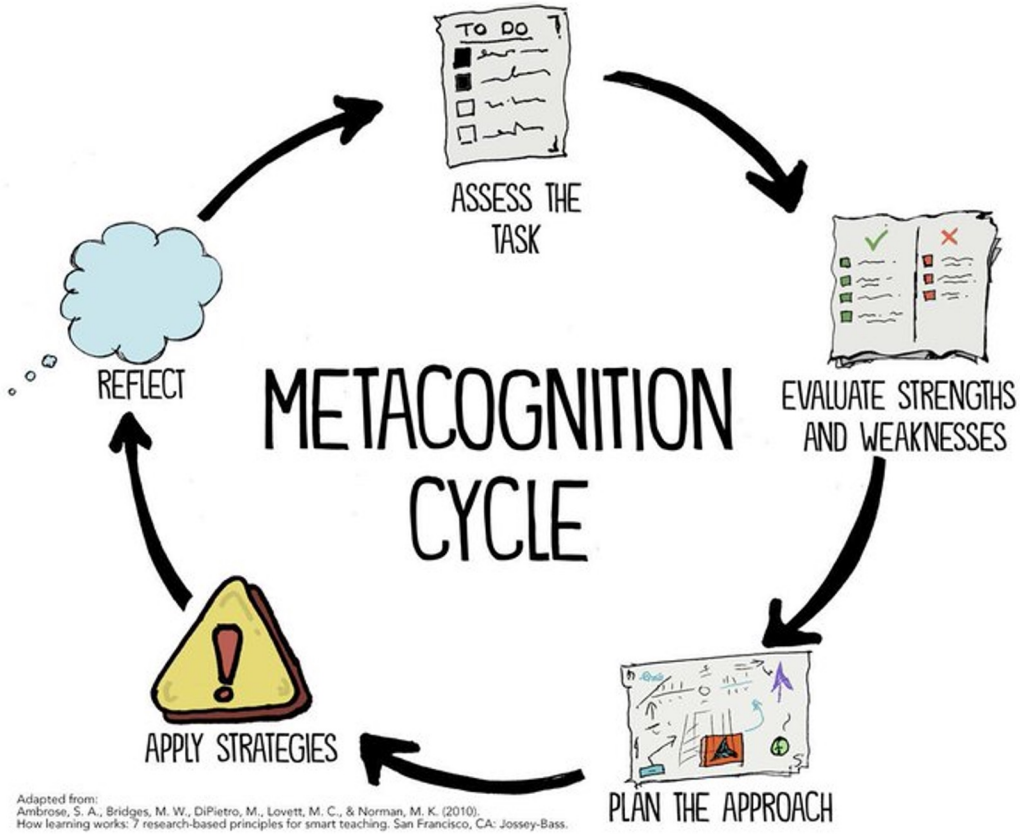
- End of Quarter TA-led Activities
  - TA Ask Me Anything & Jeopardy

Remember this?

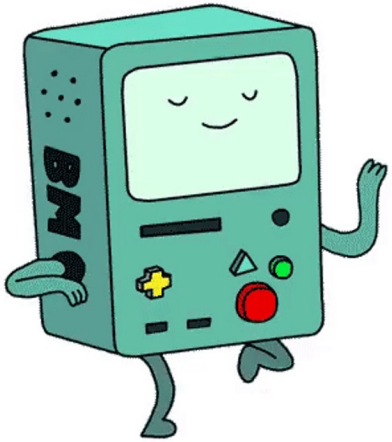


Metacognitive Skills Victory Lap

- Time Management
- Note-Taking
- Annotation
- Exam Preparation & Test-Taking
- Debugging
- Working with Instructors & TAs
- Oral Communication

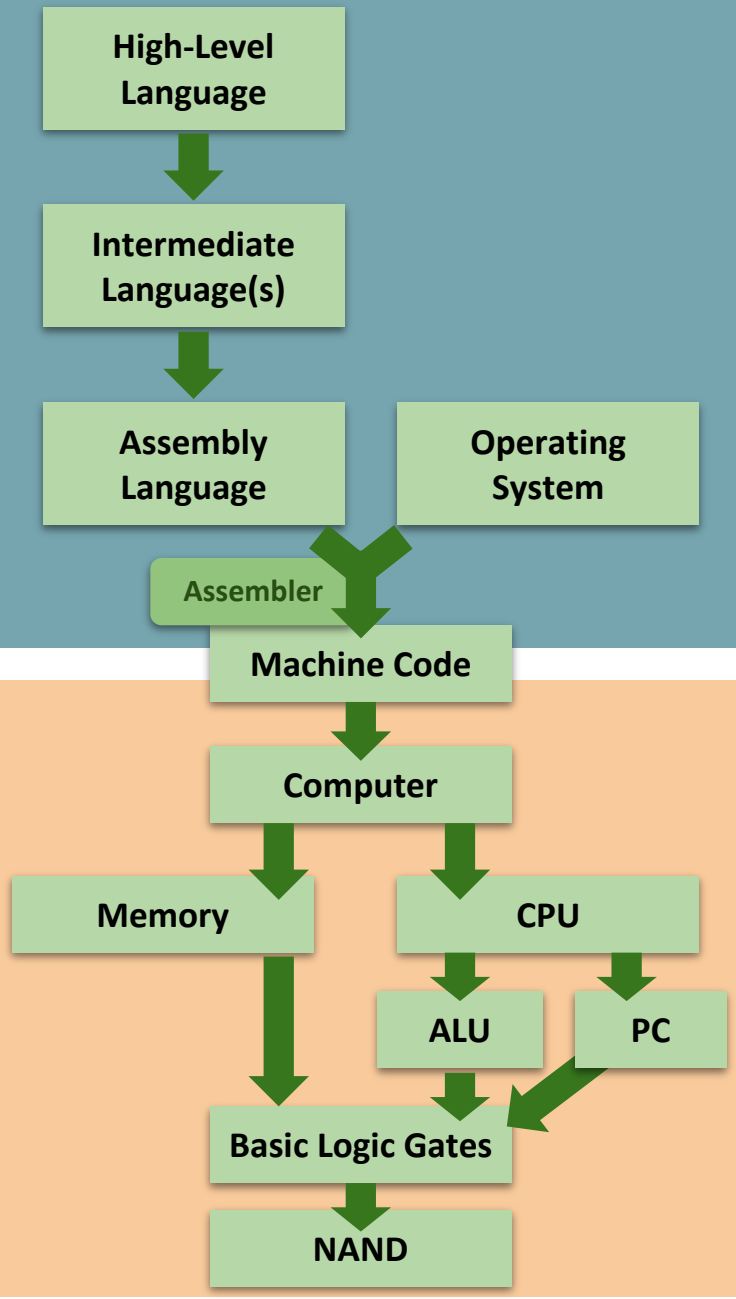


Nand2Tetris Projects



- By **building a computer**, you've accomplished something that very few others have done!
  - Many software writers consider building the computer as Somebody Else's Problem™
  - But so many technical skills and CSE courses tie into this task
  - And even if you only write Java for the rest of time…
    - Understanding the "layer below" makes you a better engineer in the "layer above"!

Roadmap



SOFTWARE

HARDWARE

Project 1 Example: Xor (cont'd)

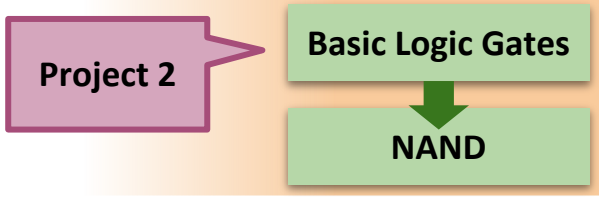
- Step 2: Use truth table to generate a Boolean function
  - Let's use the Boolean function synthesis strategy from the reading
  - Row 2 = NOT(A) AND B
  - Row 3 = A AND NOT(B)
  - A XOR B = Row 2 OR Row 3

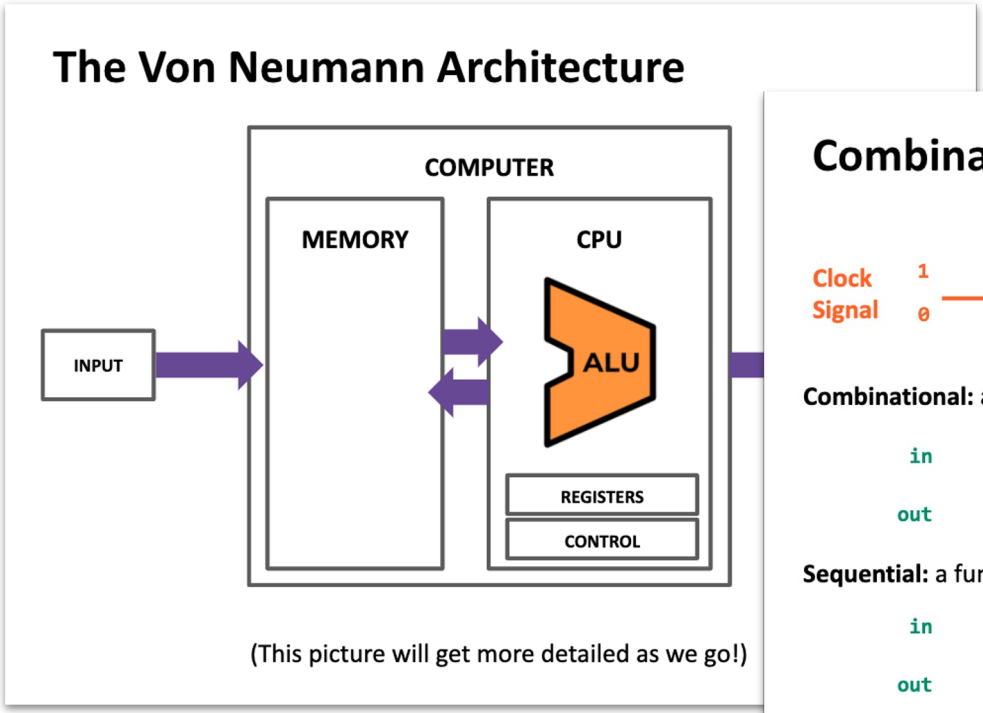
= (NOT(A) AND B) OR (A AND NOT(B))

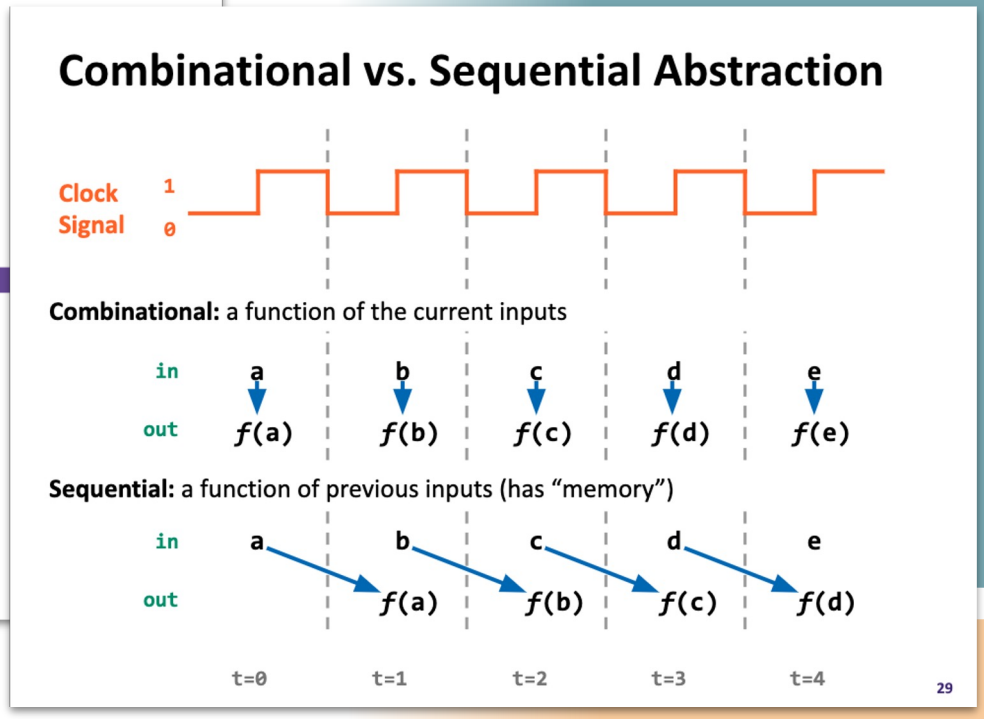
A	B	F	
0	0	0	(Row 1)
0	1	1	(Row 2)
1	0	1	(Row 3)
1	1	0	(Row 4)

F = A XOR B

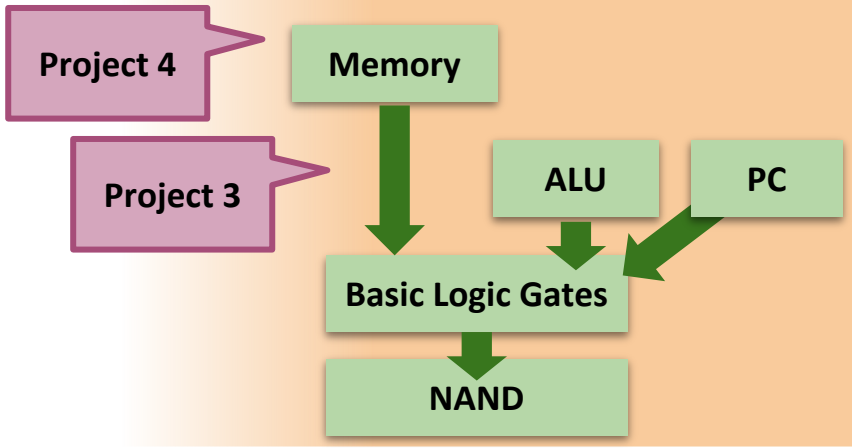
- Boolean function synthesis
- Practice with HDL—an unfamiliar, declarative style of programming







- Components found in real-world computers: ALU, PC, Memory, etc.
- Learning a mental model for sequential logic



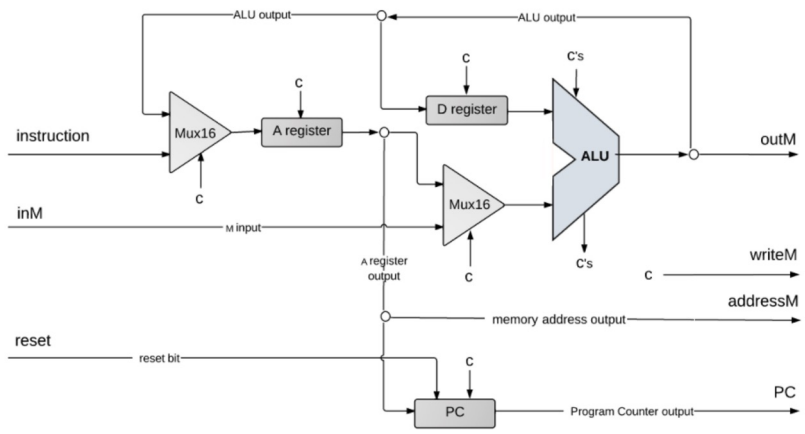
Hack: C-Instructions

Symbolic: `dest = comp ; jump`

Binary: `1 1 1 a c1 c2 c3 c4 c5 c6 d1 d2 d3 j1 j2 j3`

- Family: C-Instruction
- unused
- Comp: ALU Operation (a bit chooses between A and M)
- Dest: Where to store result
- Jump: Condition for jumping

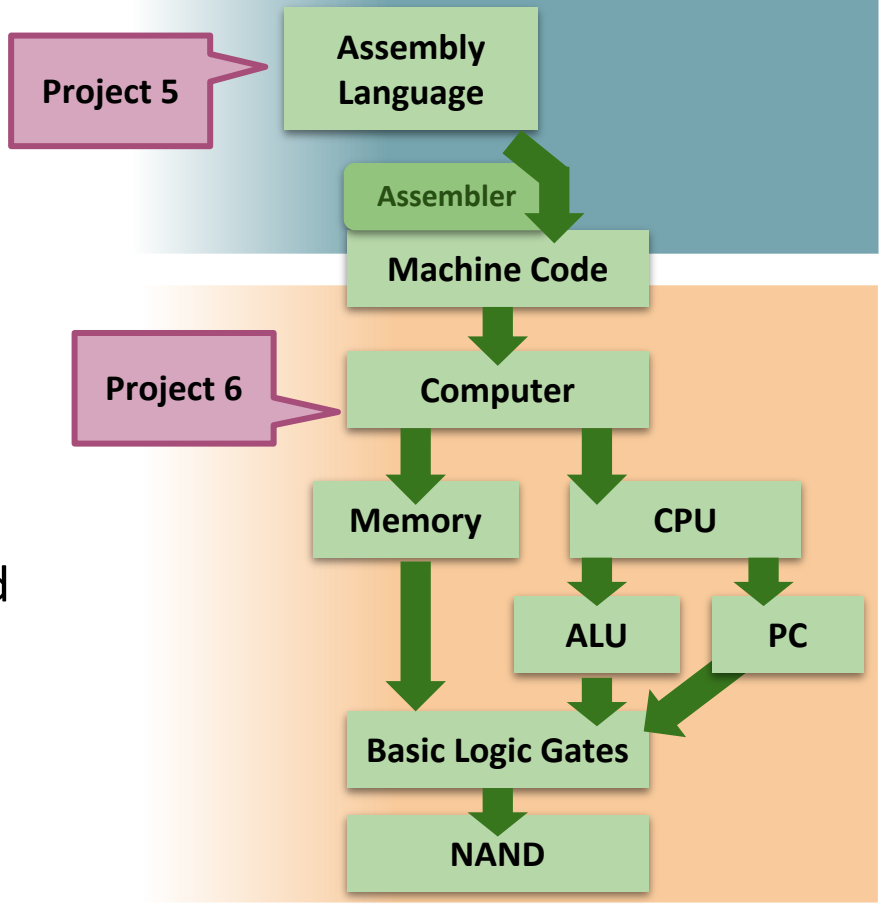
Hack CPU Implementation



(each "C" symbol represents a control bit)

26

- The connection between software and hardware through binary instructions
- What must happen in a clock cycle to process one instruction



The Compiler: Implementation

```
public int fact(int n) {
  if (n == 0) {
    return 1;
  } else {
    return n * fact(n - 1);
  }
}
```
High-Level Language

```
(fact)
  @R0
  M=M+1
  @R1
  D=A
  @ifbranch
  D;JEQ
```
Assembly Language

Scanner → Parser → Type Checker → Optimizer → Code Generator

Break string into discrete **tokens**: IF (ID(n) == NUM(0) etc.

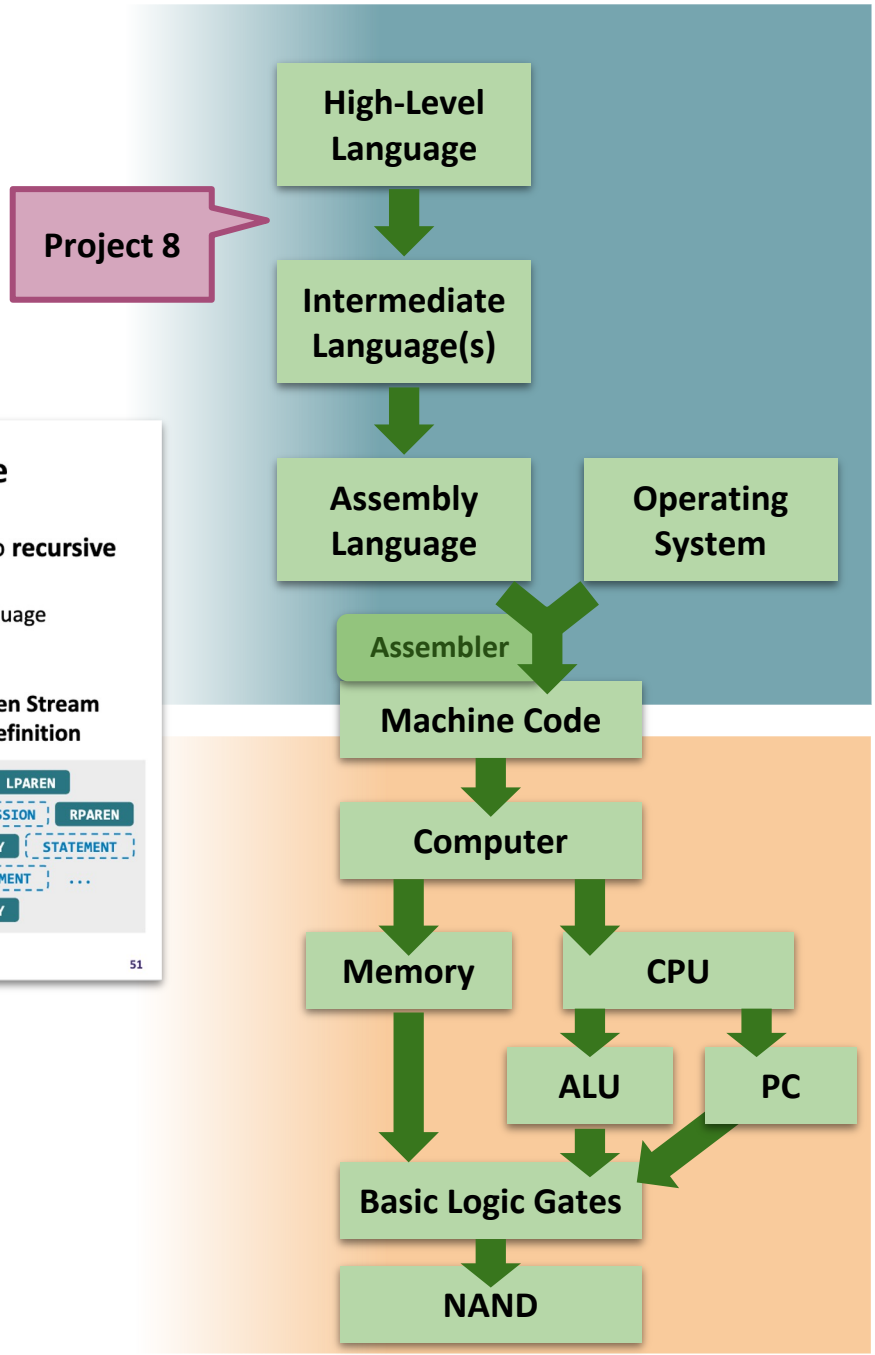
Arrange tokens into **syntax tree**: + (x, 10)

Describing a Programming Language

- These broad categories lend themselves well to **recursive definitions**
  - Easily express all possible configurations of the language constructs

Symbolic Example	General Definition of an `if` Statement	Token Stream Definition
`if (x == 0) { x = y; }`	`if ( EXPRESSION ) { STATEMENT STATEMENT ... }`	IF LPAREN EXPRESSION RPAREN LCURLY STATEMENT STATEMENT ... RCURLY

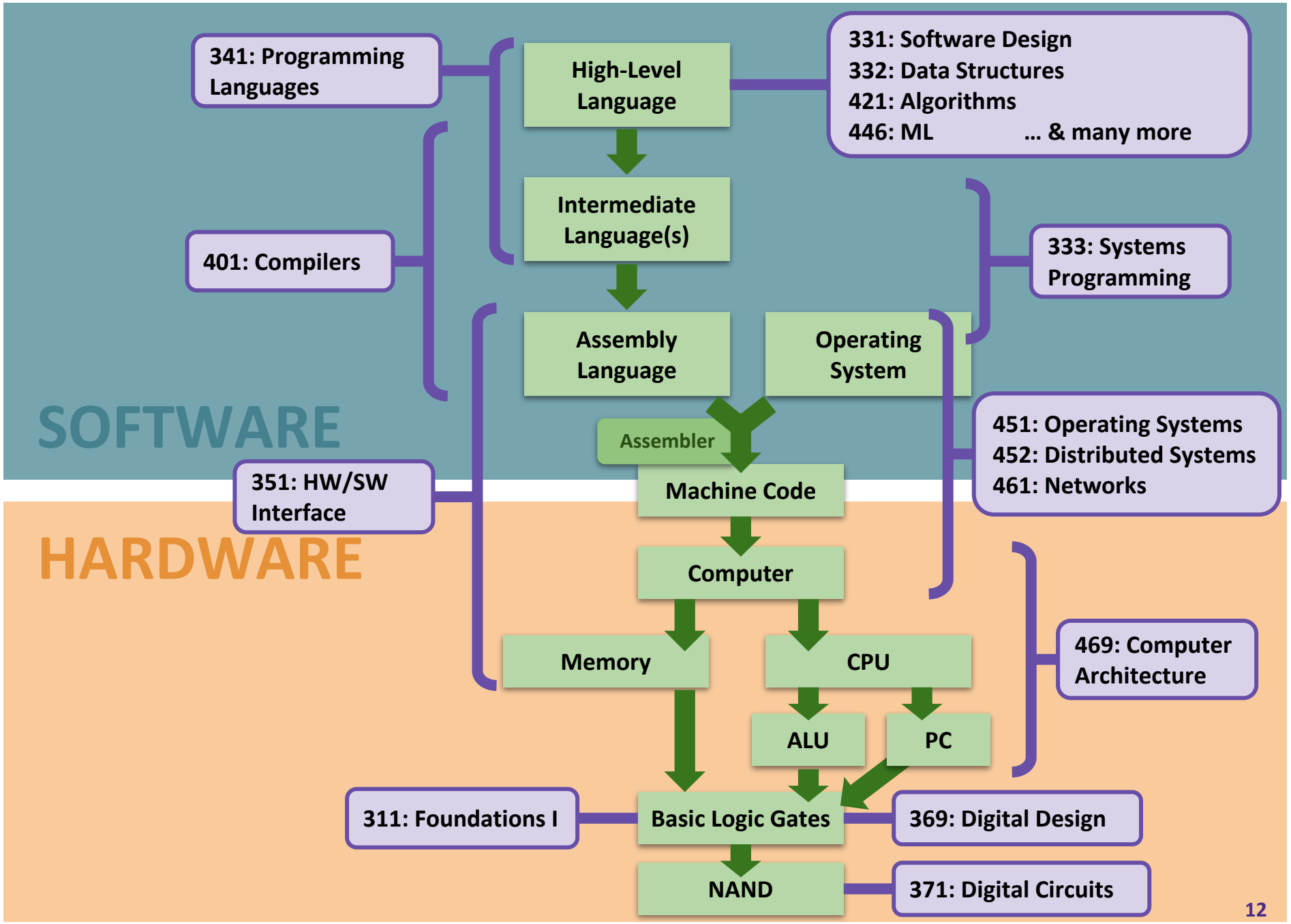
51



- What happens when you click that green run button?
- Programs can read in programs and then spit out equivalent programs

Takeaways: Why Build a Computer?

- **A significant engineering effort**
  - You practiced so many skills and programmed with so many different languages, tools, & paradigms—and you can do it again!
- We hope this was a **demystifying experience**
  - To not see CSE courses as isolated but as interconnected puzzle pieces
- We hope you had **fun** in this learning journey!
  - The computing field is broad and has much for you to explore
  - We are hopeful you found a topic you want to pursue further, both technically and metacognitively


341: Programming Languages
High-Level Language
331: Software Design
332: Data Structures
421: Algorithms
446: ML
… & many more
Intermediate Language(s)
401: Compilers
333: Systems Programming
Assembly Language
Operating System
SOFTWARE
Assembler
451: Operating Systems
452: Distributed Systems
461: Networks
351: HW/SW Interface
Machine Code
HARDWARE
Computer
Memory
CPU
469: Computer Architecture
ALU
PC
311: Foundations I
Basic Logic Gates
369: Digital Design
NAND
371: Digital Circuits

Five-minute Break!

- Feel free to stand up, stretch, use the restroom, drink some water, review your notes, or ask questions
- We'll be back at:
- Research shows mid-lecture breaks reduce the decline of attention in the middle of lecture (Olmsted, 1999)

Olmsted III, John. "The Mid-Lecture Break: When Less Is More." *Journal of Chemical Education* (1999). https://pubs.acs.org/doi/abs/10.1021/ed076p525.

Lecture Outline

- CSE 390B Victory Lap
  - Nand2Tetris Projects and Metacognitive Subjects

- **End of Quarter TA-led Activities**
  - **TA Ask Me Anything & Jeopardy**

TA Ask Us Anything!

- Ask us about…
  - Classes
    - Recommendations for easy, hard, useful, etc. classes
    - What classes go well with each other
  - Extracurricular activities
    - TAing
    - Allen School RSOs
    - UW RSOs
  - Internships
  - How we feel about CS…

Jeopardy!

- Organize into teams of 3-4 students
- The first team to raise their hand and answer the question wins that round and chooses the next question

Lecture 20 Wrap-up

- Office Hours and Student-TA 1:1 meetings end this week
  - Course staff open to meeting during finals week by appointment

- Project Reminders
  - **Final Project, Part I: Outline of E-Portfolio due tonight (6/2) at 11:59pm PDT**
    - Submission can be a text entry box, a website URL, a media recording, a file upload, or a draft of your E-Portfolio
  - Final Project, Part II: Final E-Portfolio Submission due next Tuesday at 4pm PDT (right before we meet for final presentations)
  - If you have any uncompleted projects, the last day to turn them in is this Friday (6/3) at 11:59pm PDT

Lecture 20 Wrap-up

- Please fill out the CSE 390B course evaluation if you have not already
  - Linked here: https://uw.iasystem.org/survey/260462

- Looking forward to your final presentations next Tuesday!