

x86-64 Programming III

CSE 351 Autumn 2023

Instructor:

Justin Hsia

Teaching Assistants:

Afifah Kashif

Malak Zaki

Bhavik Soni

Naama Amiel

Cassandra Lam

Nayha Auradkar

Connie Lam

Nikolas McNamee

David Dai

Pedro Amarante

Dawit Hailu

Renee Ruan

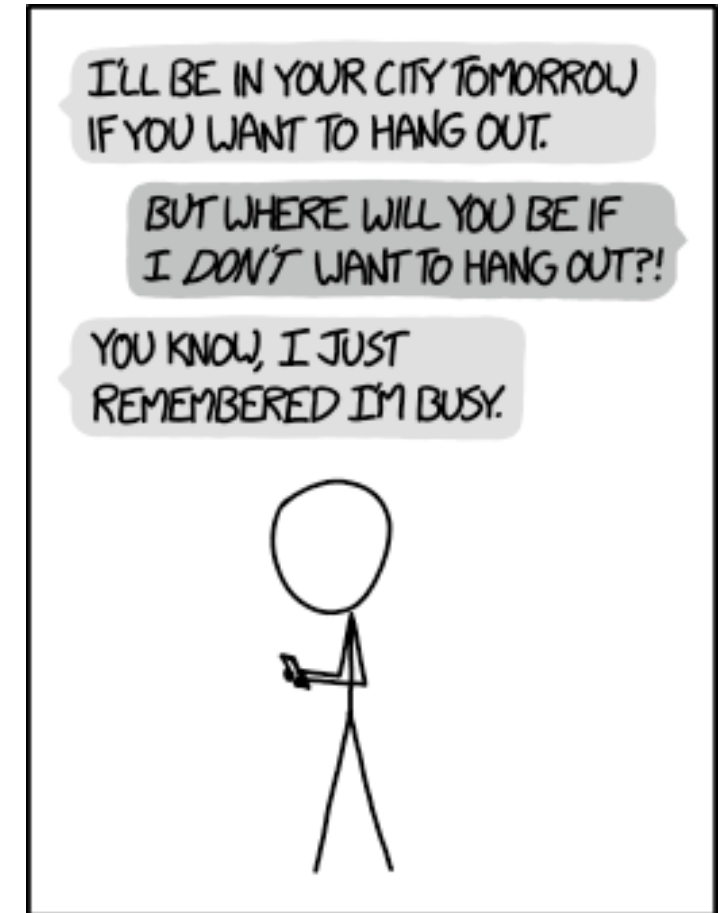
Ellis Haker

Simran Bagaria

Eyoel Gebre

Will Robertson

Joshua Tan



WHY I TRY NOT TO BE
PEDANTIC ABOUT CONDITIONALS.

<http://xkcd.com/1652/>

Relevant Course Information

- ❖ Lab submissions that fail the autograder get a **ZERO**
 - No excuses – make full use of tools & Gradescope’s interface
 - Leeway on Lab 1a won’t be given moving forward
- ❖ Lab 2 (x86-64) released Wednesday
 - Learn to trace x86-64 assembly and use GDB
- ❖ Midterm is in two weeks (take home, 11/2–4)
 - Open book; make notes and use [midterm reference sheet](#)
 - Individual, but discussion allowed via “Gilligan’s Island Rule”
 - Mix of “traditional” and design/reflection questions
 - Form study groups and look at past exams!

Extra Credit

- ❖ All labs starting with Lab 2 have extra credit portions
 - These are meant to be fun extensions to the labs
- ❖ Extra credit points *don't* affect your lab grades
 - From the course policies: “they will be accumulated over the course and will be used to bump up borderline grades at the end of the quarter.”
 - Make sure you finish the rest of the lab before attempting any extra credit

A detailed, colorful micrograph of a microchip die, showing a complex grid of circuitry and various colored regions. The text 'x86-64 Programming III' is overlaid in the center.

x86-64 Programming III

Lesson Summary (1/2)

- ❖ Control flow in x86 determined by Condition Codes
 - Showed **C**arry, **Z**ero, **S**ign, and **O**verflow, though others exist
 - Set flags with arithmetic & logical instructions (implicit) or Compare and Test (explicit)
 - Set instructions read out flag values
 - Jump instructions use flag values to determine next instruction to execute
 - Usually combinations of two instructions with result of first instruction compared against 0 in a way determined by second instruction

- ❖ Labels are a way to refer to specific instruction addresses as jump targets in assembly

Lesson Summary (2/2)

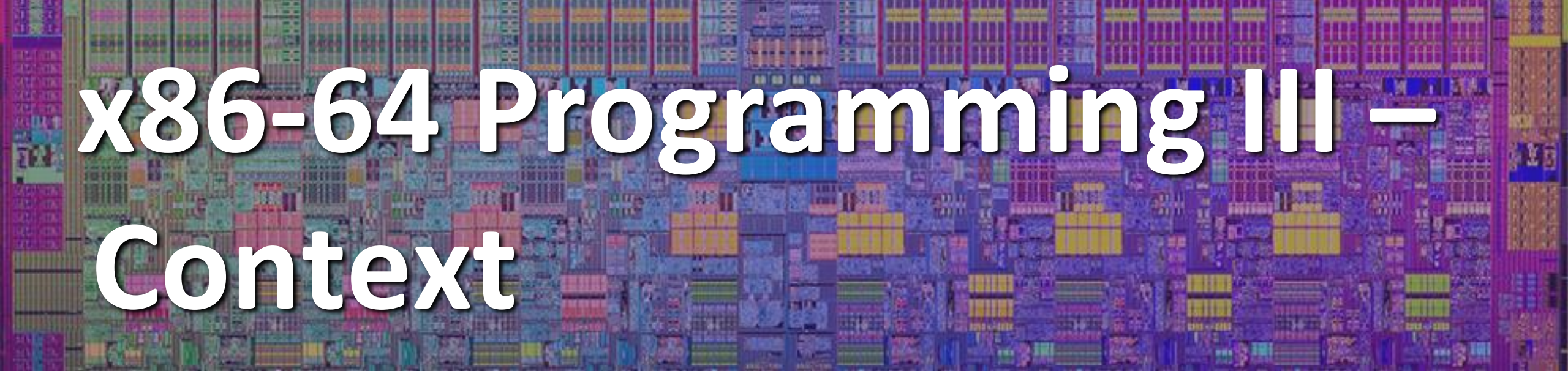
❖ Terminology:

- Condition codes: Carry Flag (CF), Zero Flag (ZF), Sign Flag (SF), Overflow Flag (OF)
- Test (`test`) and compare (`cmp`) assembly instructions
- Jump (`j*`) and set (`set*`) families of assembly instructions
- Label, jump target, program counter

❖ Learning Objectives:

- Without executing, describe the overall purpose of snippets of x86-64 assembly code containing arithmetic, if-else statements, [and/or loops].

❖ What lingering questions do you have from the lesson?

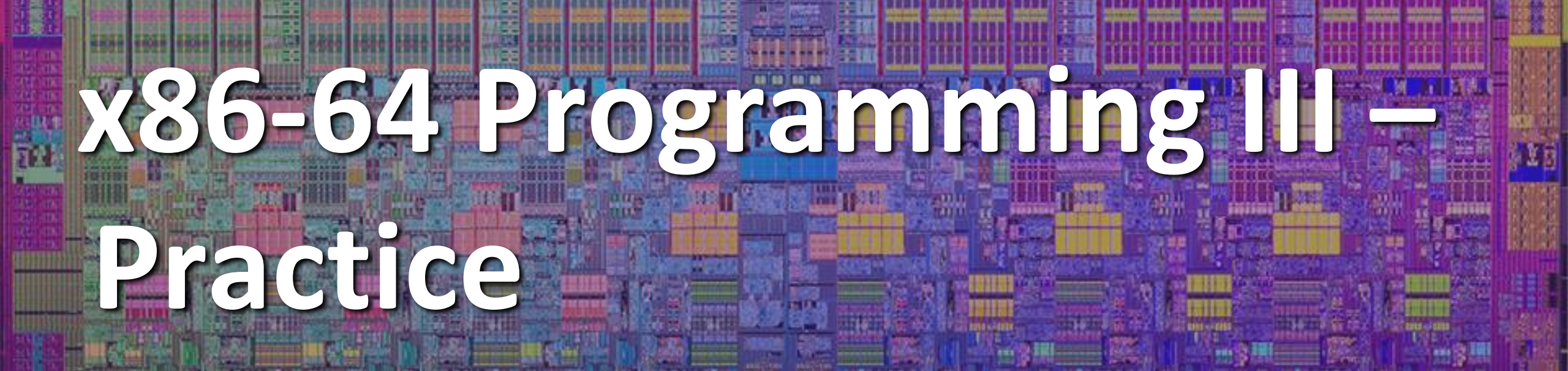
A detailed, colorful micrograph of a microchip die, showing intricate circuit patterns in shades of purple, blue, yellow, and green. The text is overlaid on this background.

x86-64 Programming II – Context

Free and Open Software

- ❖ Alternatives to software made by companies:
 - **Free Software:** Guaranteeing users the freedoms to run, study, modify, and share copies of software; based on the ethical rejection of proprietary software
 - Supported by the Free Software Foundation
 - **Open-Source Software:** Uses open-source licenses, which guarantees access to and the ability to modify the source code under a similar license
 - Steward organization is the Open Source Initiative

- ❖ In 351:
 - Linux is an *open-source* operating system; we currently use **Rocky Linux**
 - The GNU Compiler Collection (GCC) and the GNU Project Debugger (GDB) are *free* software

A detailed, colorful micrograph of a microchip die, showing a complex grid of circuitry and various colored regions (purple, blue, yellow, green, red) representing different functional blocks and interconnects.

x86-64 Programming II – Practice

Group Work Time

- ❖ During this time, you are encouraged to work on the following:
 - 1) If desired, continue your discussion
 - 2) Work on the lesson problems (solutions at the end of class)
 - 3) Work on the homework problems

- ❖ Resources:
 - You can revisit the lesson material
 - Work together in groups and help each other out
 - Course staff will circle around to provide support

Practice Question

Register	Use(s)
%rdi	1 st argument (x)
%rsi	2 nd argument (y)
%rax	return value

- A. `cmpq %rsi, %rdi`
`jle .L4`
- B. `cmpq %rsi, %rdi`
`jg .L4`
- C. `testq %rsi, %rdi`
`jle .L4`
- D. `testq %rsi, %rdi`
`jg .L4`
- E. **We're lost...**

```
long absdiff(long x, long y)
{
    long result;
    if (x > y)
        result = x-y;
    else
        result = y-x;
    return result;
}
```

```
absdiff:
    _____
    _____
                                     # x > y:
    movq    %rdi, %rax
    subq    %rsi, %rax
    ret
.L4:                                     # x <= y:
    movq    %rsi, %rax
    subq    %rdi, %rax
    ret
```