

CSE 390B, Spring 2023

Building Academic Success Through Bottom-Up Computing

E-Portfolio Workshop & Computer Networks

Final Project Overview, E-Portfolio Workshop, Overview of
Computer Networks

Lecture Outline

- ❖ **Final Project Overview**
 - **E-Portfolio Details and Topics Brainstorming**

- ❖ **E-Portfolio Workshop**
 - Reflection Work Session and Feedback

- ❖ **Overview of Computer Networks**
 - Connecting Computers to The Internet

Final Project E-Portfolio Overview

- ❖ You will create an E-Portfolio that is geared toward a new Allen School student
- ❖ Your E-Portfolio is a culminating project in having you reflect on the **metacognitive skills** you've learned and **providing advice** for entering the program
- ❖ During our final class, you will give a short presentation on your E-Portfolio

Final Project Due Dates

- ❖ Part I: E-Portfolio Outline
 - Due next Tuesday (5/30) at 11:59pm

- ❖ Part II: Final E-Portfolio
 - Due Tuesday of finals week (6/6) at 4:00pm

- ❖ Part III: E-Portfolio Presentations
 - During the scheduled CSE 390B final
 - CSE 390B Final Time: Tuesday, 6/6 from 4:30-6:20pm
 - CSE 390B Final Location: CSE2 G01 (same as usual classroom)

Reflection on Metacognitive Skills

Individually first, take some time to reflect on the following questions, and then discuss in groups:

- ❖ Which two metacognitive topics would you consider including in your E-Portfolio and why?
 - Reflect on which ones you've grown the most in, have impacted you the most, were most challenging to grow in, etc.

- ❖ What are some examples of yourself demonstrating those two metacognitive skills?
 - Please be specific here! Aim to share these skills as if you are telling a story and showing concrete applications of these skills

Reflection on a Technical Skill

Individually first, take some time to reflect on the following questions, and then discuss in groups:

- ❖ What technical topic from CSE 390B would you consider including in your E-Portfolio and why?
 - Reflect on technical skills that helped connect the dots, were most interesting to you, most challenging for you to grasp, etc.

- ❖ What is the impact of having knowledge of that technical skill? In other words, why is that technical skill useful?
 - Please be specific here as well — think about how this technical skill would be useful in an academic or personal setting

Lecture Outline

- ❖ Final Project Overview
 - E-Portfolio Details and Topics Brainstorming
- ❖ **E-Portfolio Workshop**
 - **Reflection Work Session and Feedback**
- ❖ Overview of Computer Networks
 - Connecting Computers to The Internet

E-Portfolio Workshop

- ❖ Individually, spend 15-20 minutes completing the following steps:
 - Aim to finalize the two metacognitive skills, two examples of you applying them, and one technical skill you plan on reflecting on
 - Begin drafting your reflection on a document in paragraph form
- ❖ Goal by the end of class today is to receive feedback on your reflections and complete part one of the final project, the E-Portfolio outline

E-Portfolio Workshop

- ❖ Now, get into groups and complete the following:
 - One group member presents on their reflection so far
 - Each group member listening should provide one question, comment, constructive feedback, or complement to the presenter
 - Repeat until everyone has had a chance to present

Lecture Outline

- ❖ Final Project Overview
 - E-Portfolio Details and Topics Brainstorming
- ❖ E-Portfolio Workshop
 - Reflection Work Session and Feedback
- ❖ **Overview of Computer Networks**
 - **Connecting Computers to The Internet**

Overview of Computer Networks

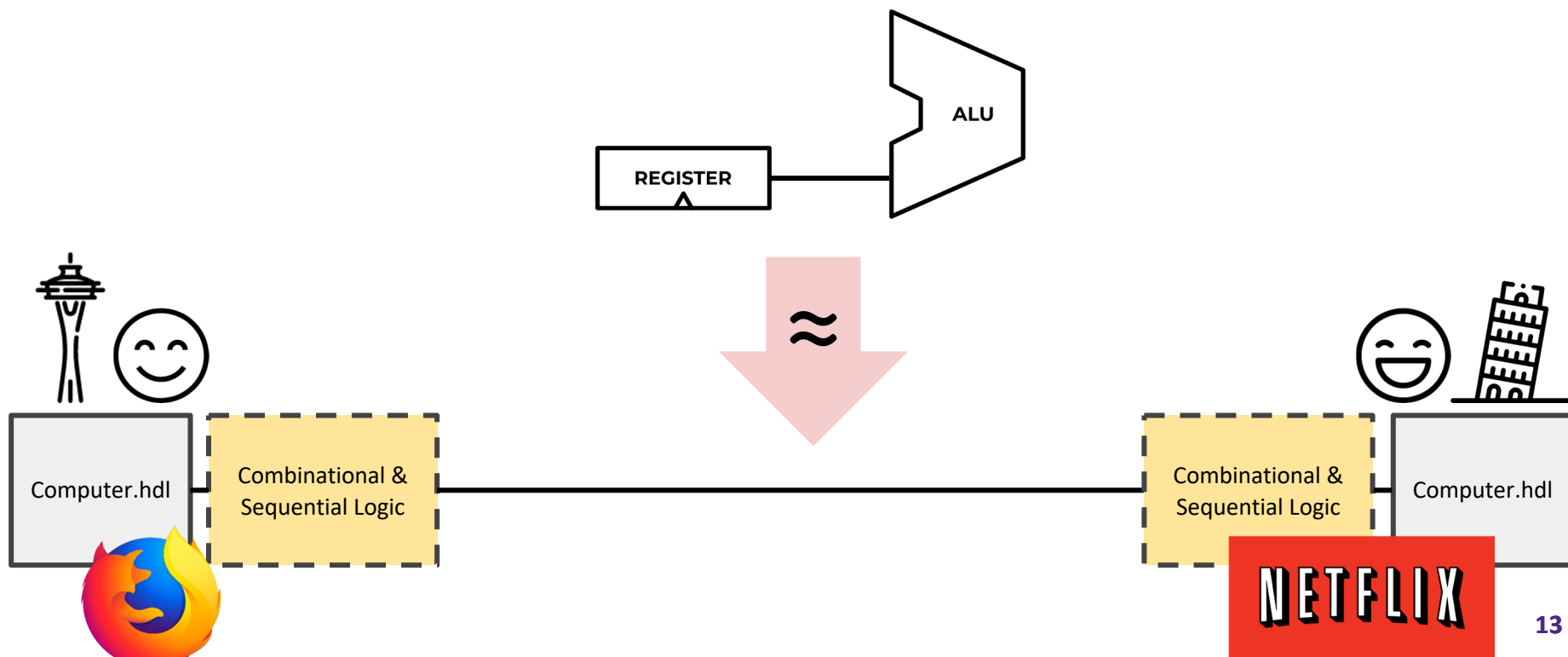
- ❖ We will go over an overview of networks
 - Take CSE 333 (Systems Programming), CSE 461 (Computer Networks), and CSE 452 (Distributed Systems) to learn more
- ❖ Our focus:
 - Brief intro to what connecting to the internet looks like under the hood
 - What that connection might look like implemented in our computer

How Do Computer Networks Work?

- ❖ How do you think we connect computers in different physical locations?
- ❖ What modes of communication do you think computers use to network with one another?

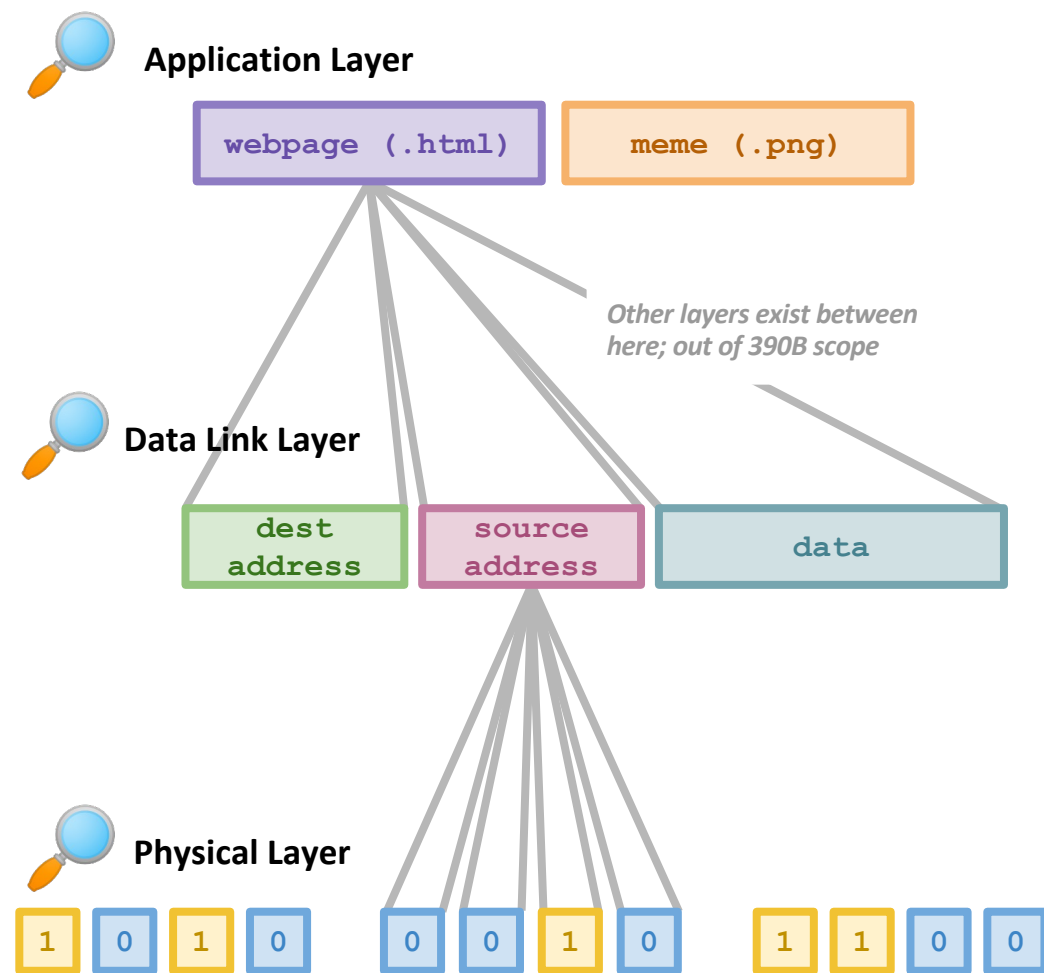
Networks = Really, Really Long Wires

- ❖ At a fundamental level, there's nothing magic about the Internet—it's the same concepts we used to build our CPU, just with longer wires
 - Still 1s and 0s, still just combinational + sequential logic



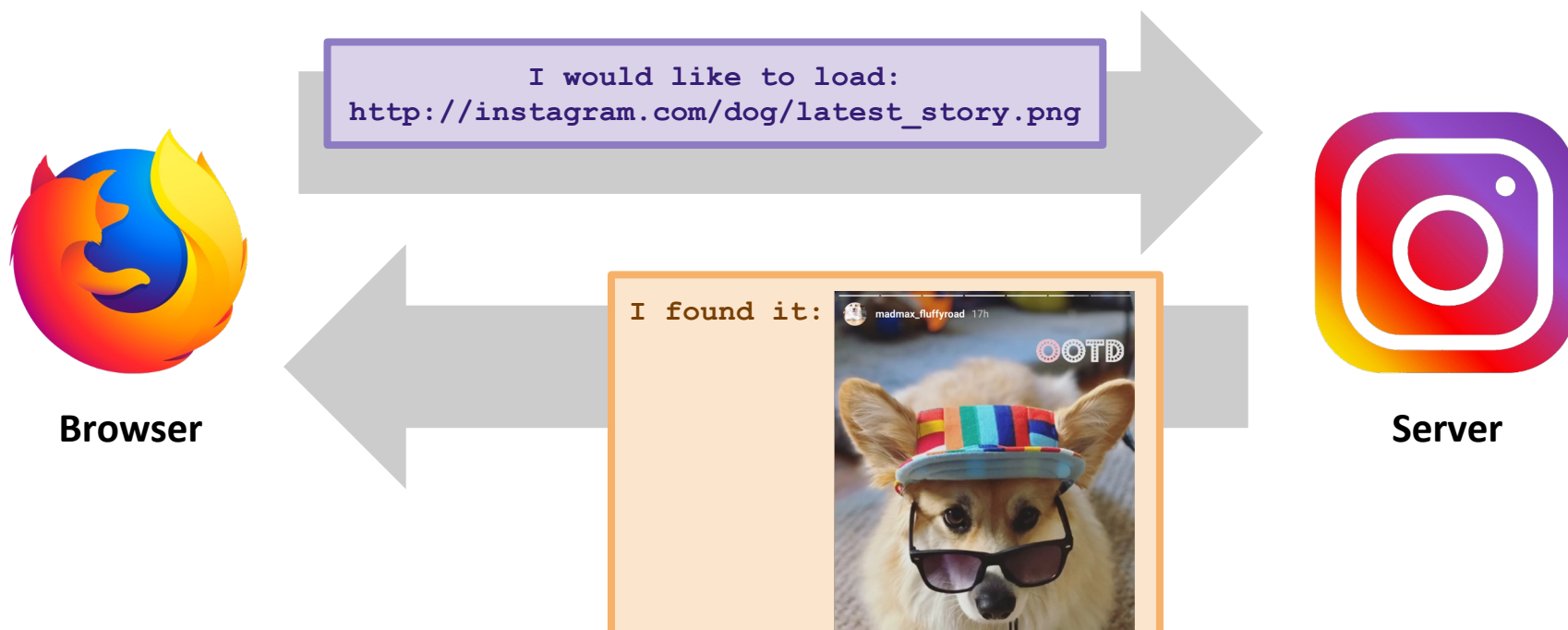
Thinking about the Network: Layers

- ❖ To manage the complexity, we think about the network in layers
- ❖ It's all 0s and 1s, but each layer is a different way of “framing” or thinking about those 0s and 1s
 - Each layer zooms out a little more



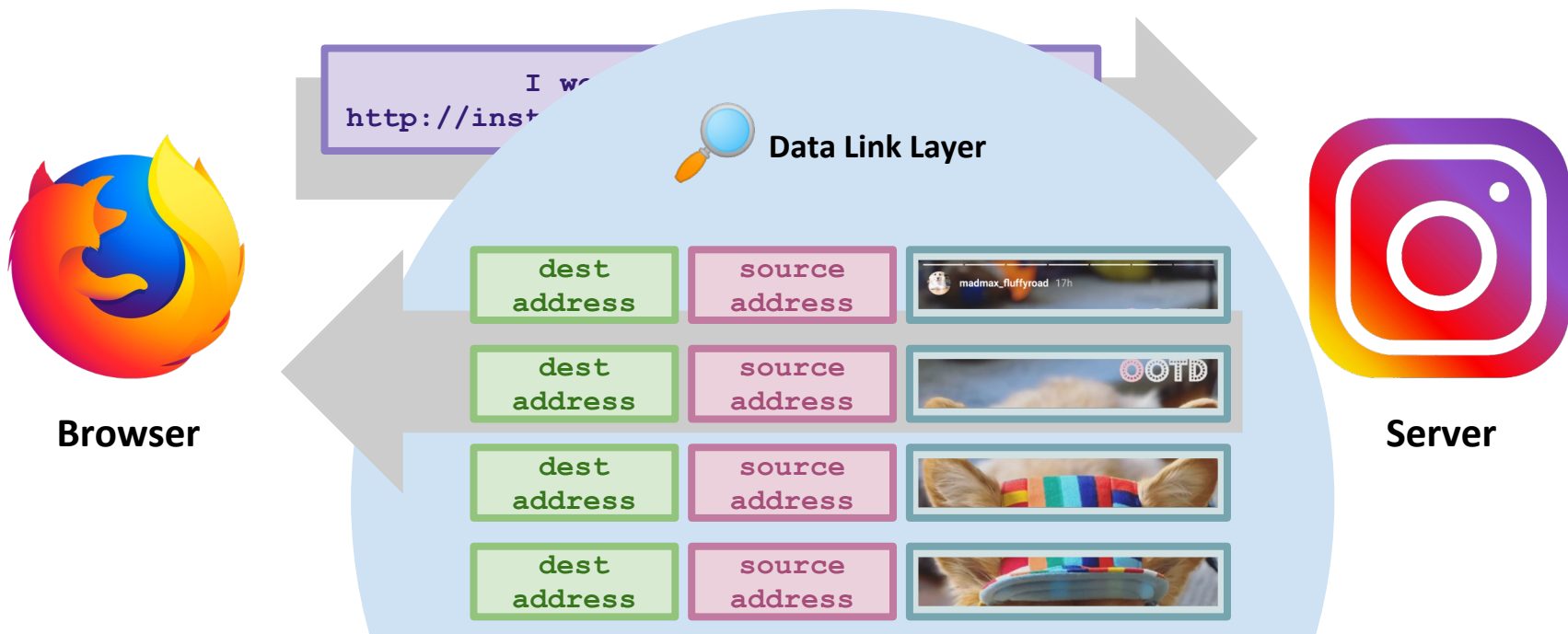
Application Layer

- ❖ Conceptually the “top” layer: looking at internet traffic as direct communication between applications
- ❖ Common use: HTTP (HyperText Transfer Protocol)
 - Your browser sends an HTTP request to a server
 - The server sends back an HTTP response with data attached



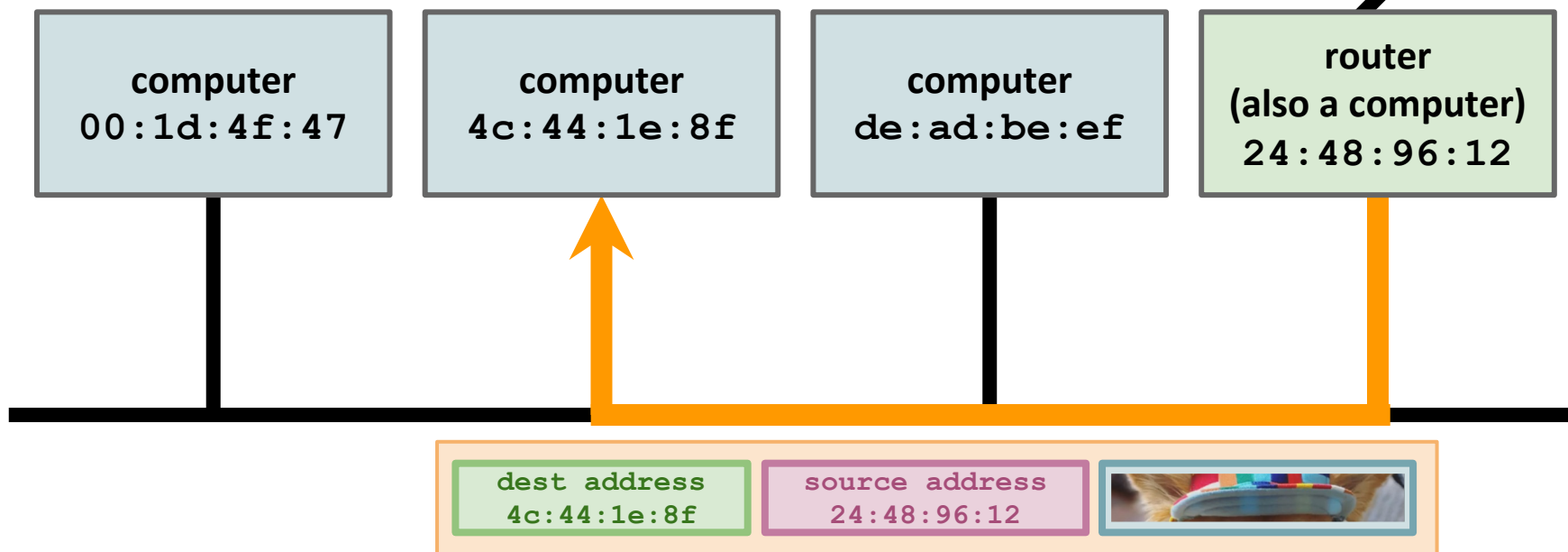
Application Layer

- ❖ Conceptually the “top” layer: looking at internet traffic as direct communication between applications
- ❖ Common use: HTTP (HyperText Transfer Protocol)
 - Your browser sends an HTTP request to a server
 - The server sends back an HTTP response with data attached



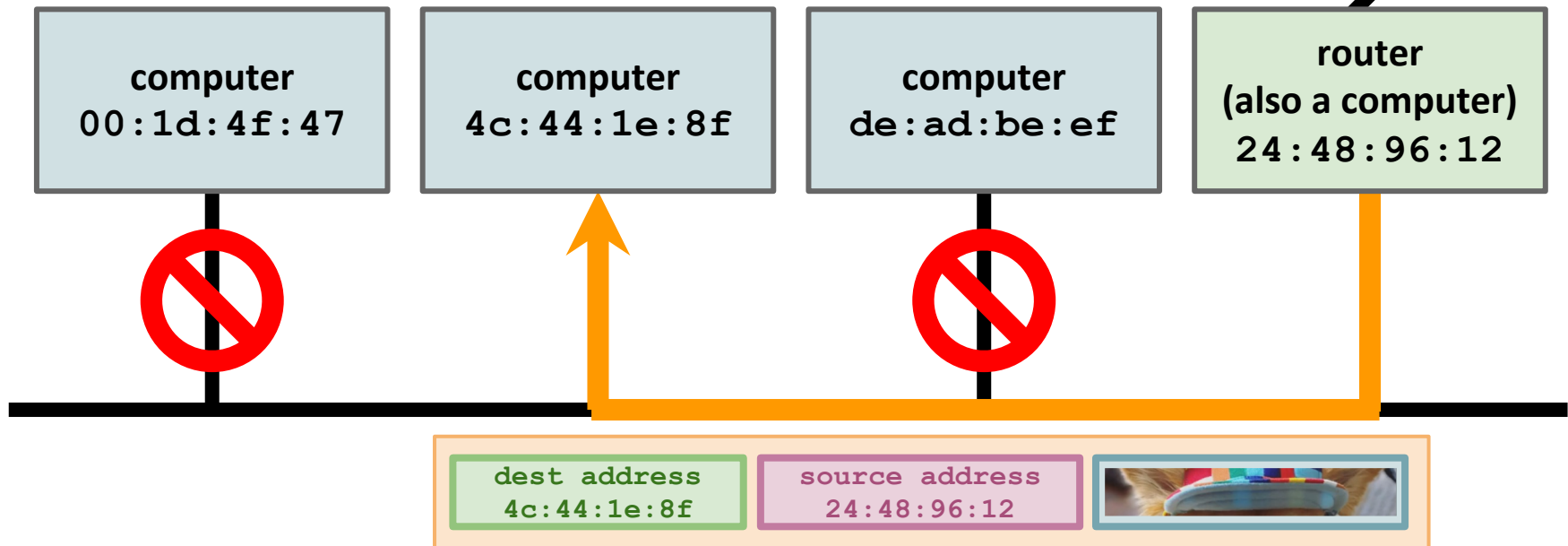
Data Link Layer

- ❖ A computer network is simply multiple computers connected by a single wire
- ❖ Why is it better to send smaller chunks of data?



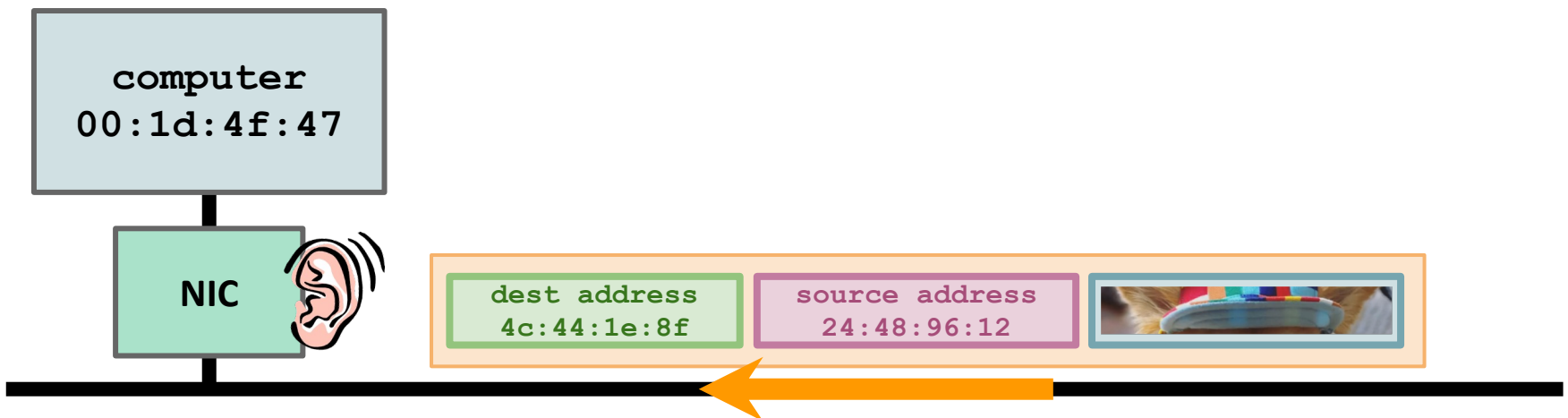
Data Link Layer

- ❖ Every computer will “hear” the message
- ❖ How do the other computers know to ignore an incoming packet of data?



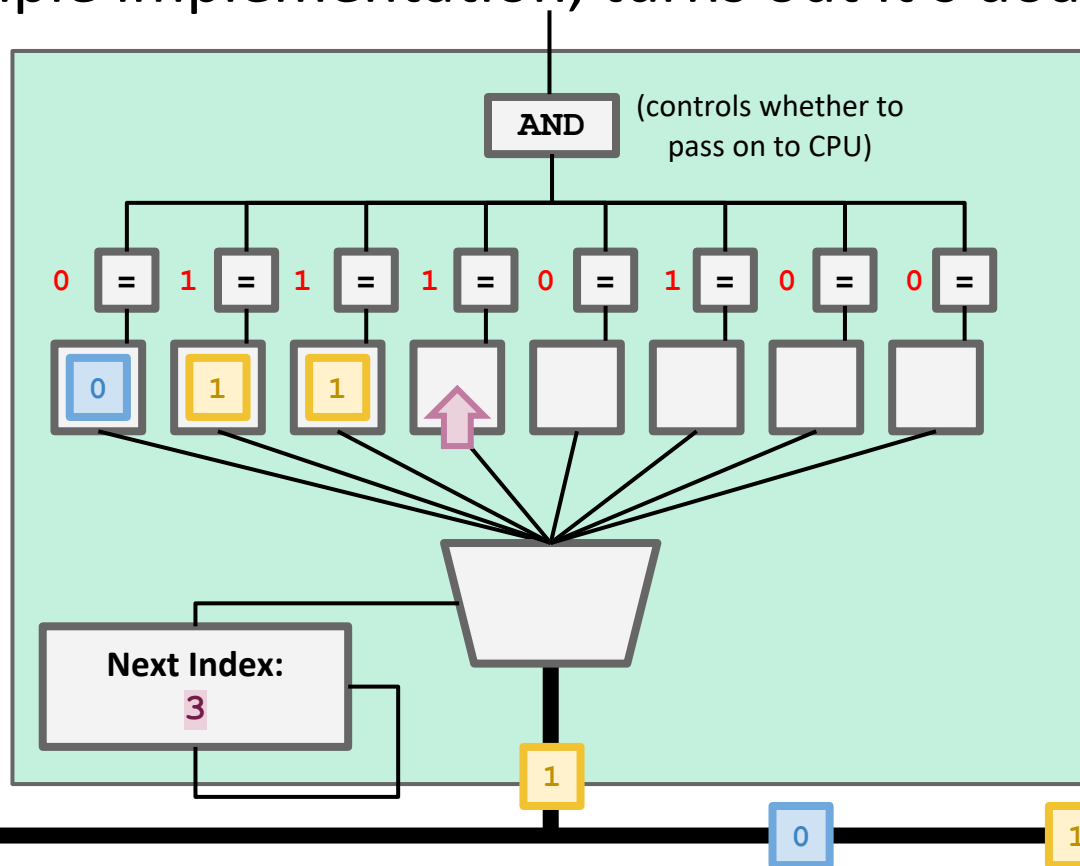
NIC (Network Interface Card)

- ❖ We don't want the CPU to waste time always listening to the network wire, especially when it's not even the destination computer
- ❖ Solution: **the NIC**—a new piece of the computer dedicated to dealing with the network wire
 - Listens to the network wire until it hears a destination address, checks if it matches this computer, and only sends to CPU if so



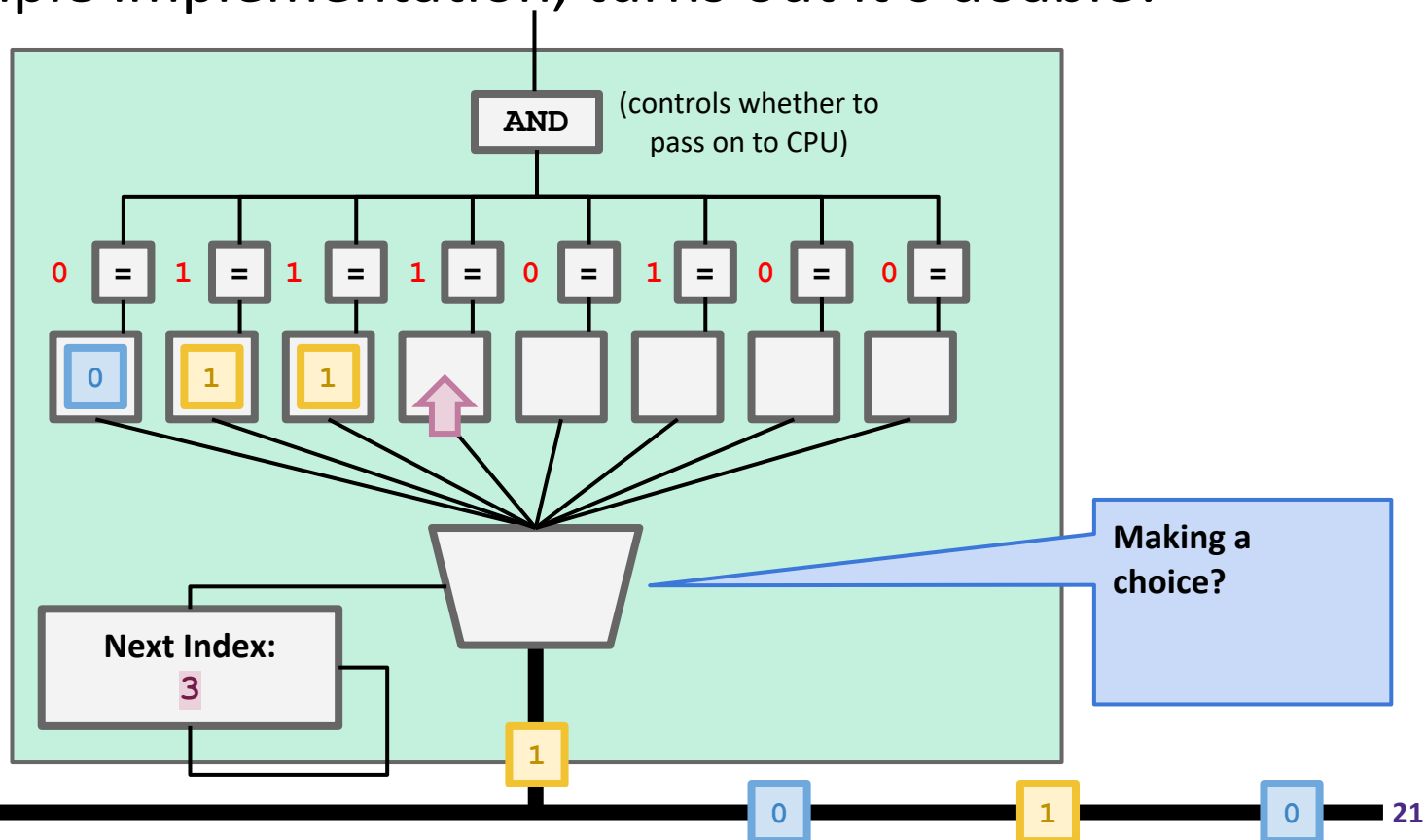
NIC Implementation in Your Computer

- ❖ Before, we would have to accept NIC as “magic”
- ❖ Now, we can imagine exactly how to build this chip, and for a simple implementation, turns out it’s doable!



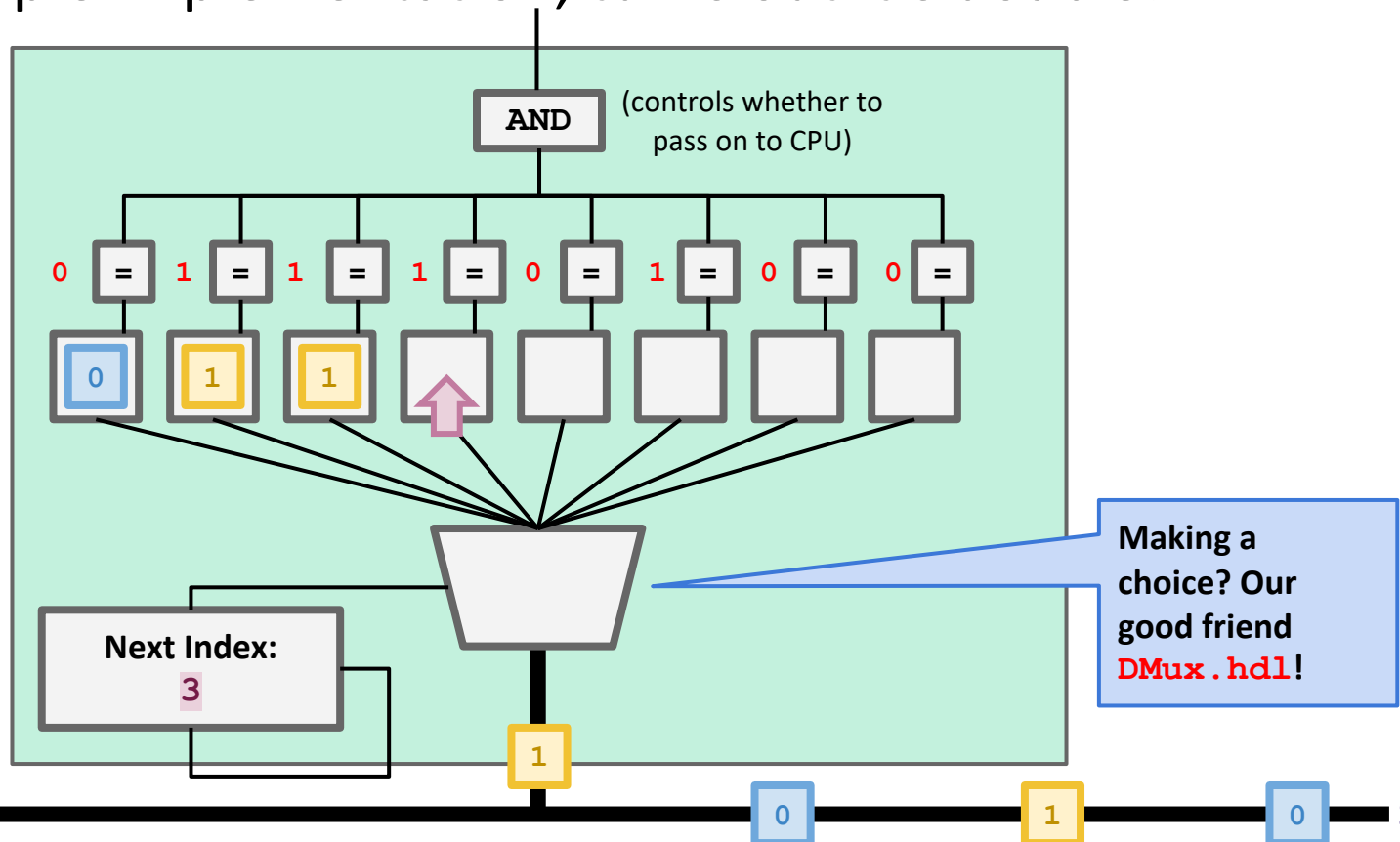
NIC Implementation in Your Computer

- ❖ Before, we would have to accept NIC as “magic”
- ❖ Now, we can imagine exactly how to build this chip, and for a simple implementation, turns out it’s doable!



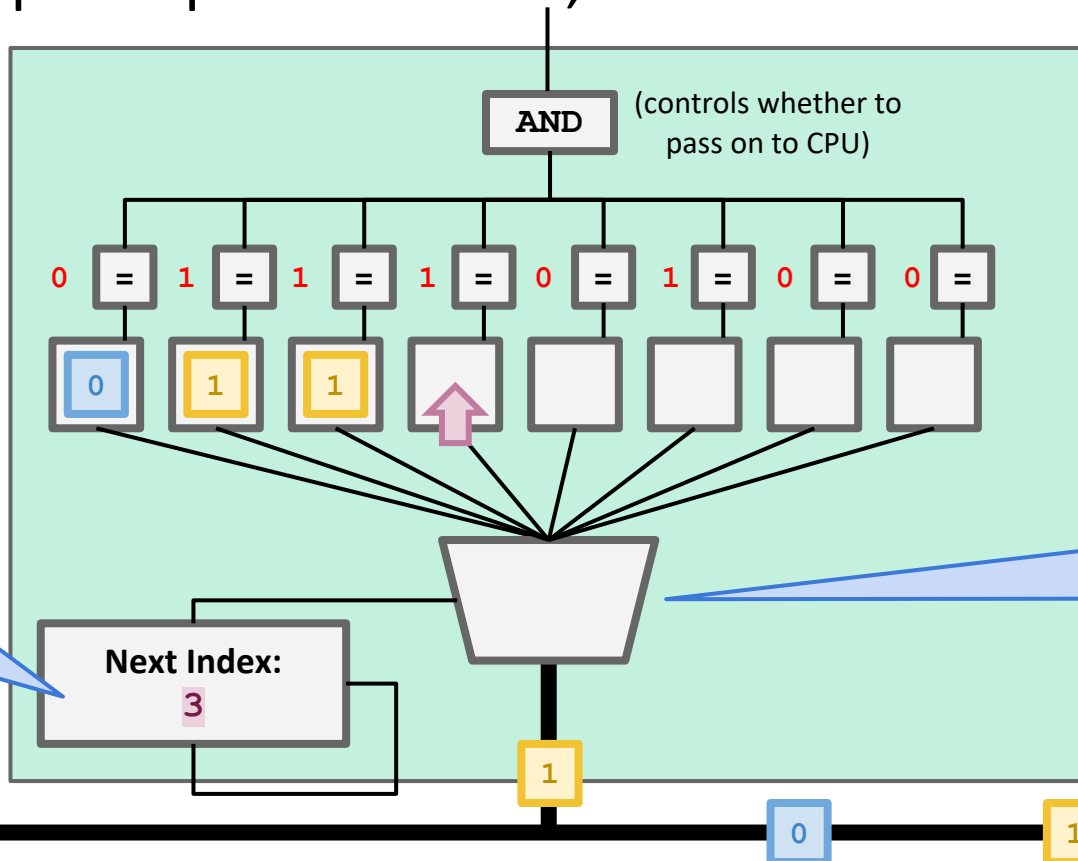
NIC Implementation in Your Computer

- ❖ Before, we would have to accept NIC as “magic”
- ❖ Now, we can imagine exactly how to build this chip, and for a simple implementation, turns out it's doable!



NIC Implementation in Your Computer

- ❖ Before, we would have to accept NIC as “magic”
- ❖ Now, we can imagine exactly how to build this chip, and for a simple implementation, turns out it’s doable!

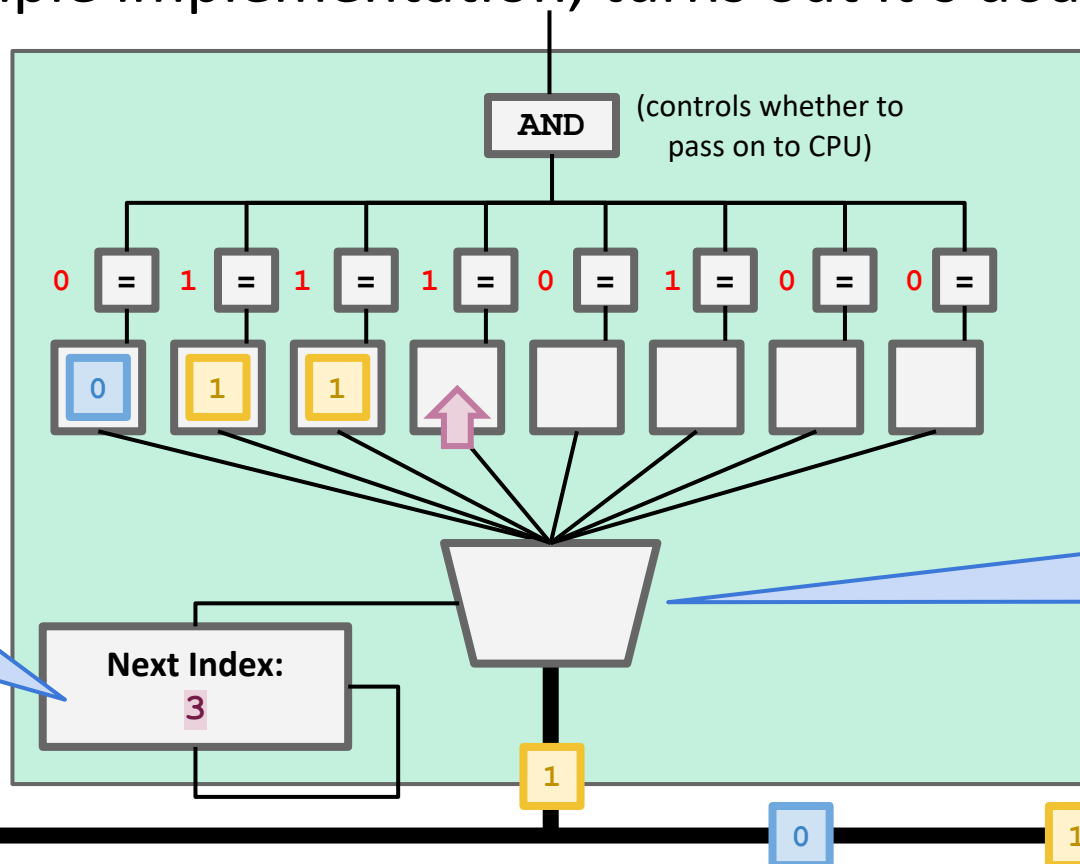


Constantly incrementing an address?

Making a choice? Our good friend **DMux.hdl!**

NIC Implementation in Your Computer

- ❖ Before, we would have to accept NIC as “magic”
- ❖ Now, we can imagine exactly how to build this chip, and for a simple implementation, turns out it’s doable!

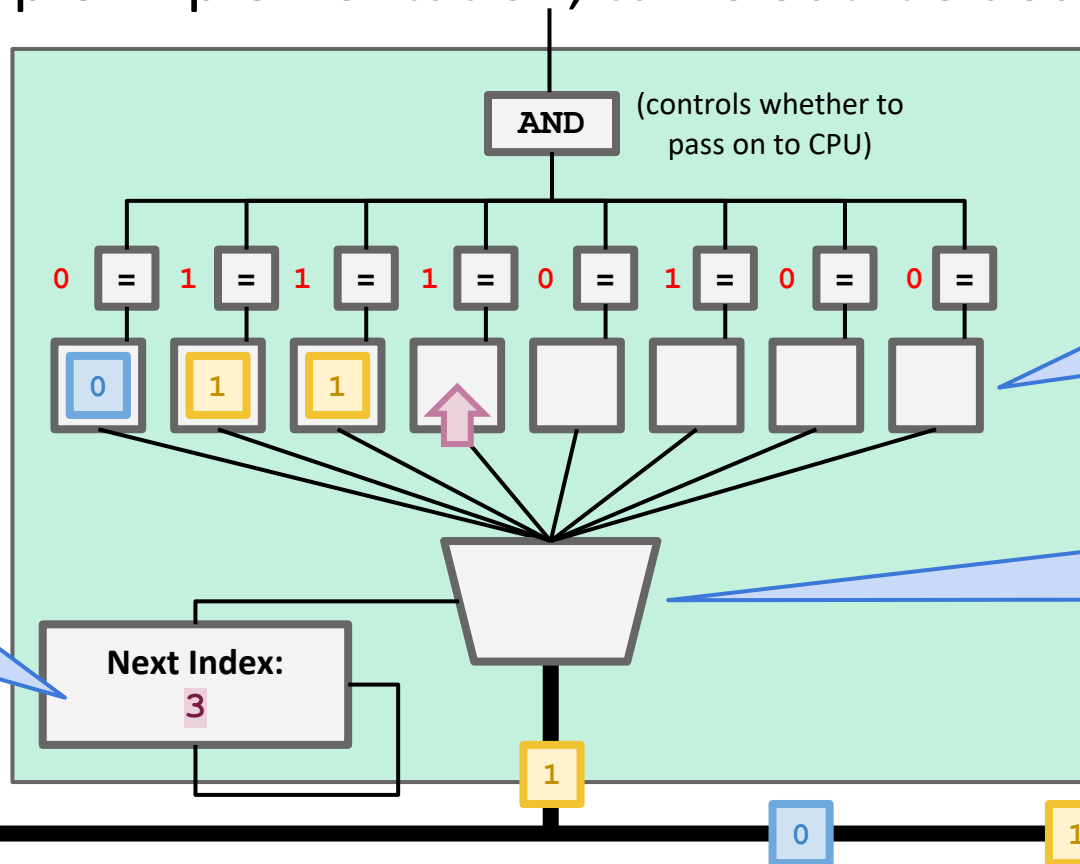


Constantly incrementing an address? Sounds like `PC.hdl!`

Making a choice? Our good friend `DMux.hdl!`

NIC Implementation in Your Computer

- ❖ Before, we would have to accept NIC as “magic”
- ❖ Now, we can imagine exactly how to build this chip, and for a simple implementation, turns out it’s doable!



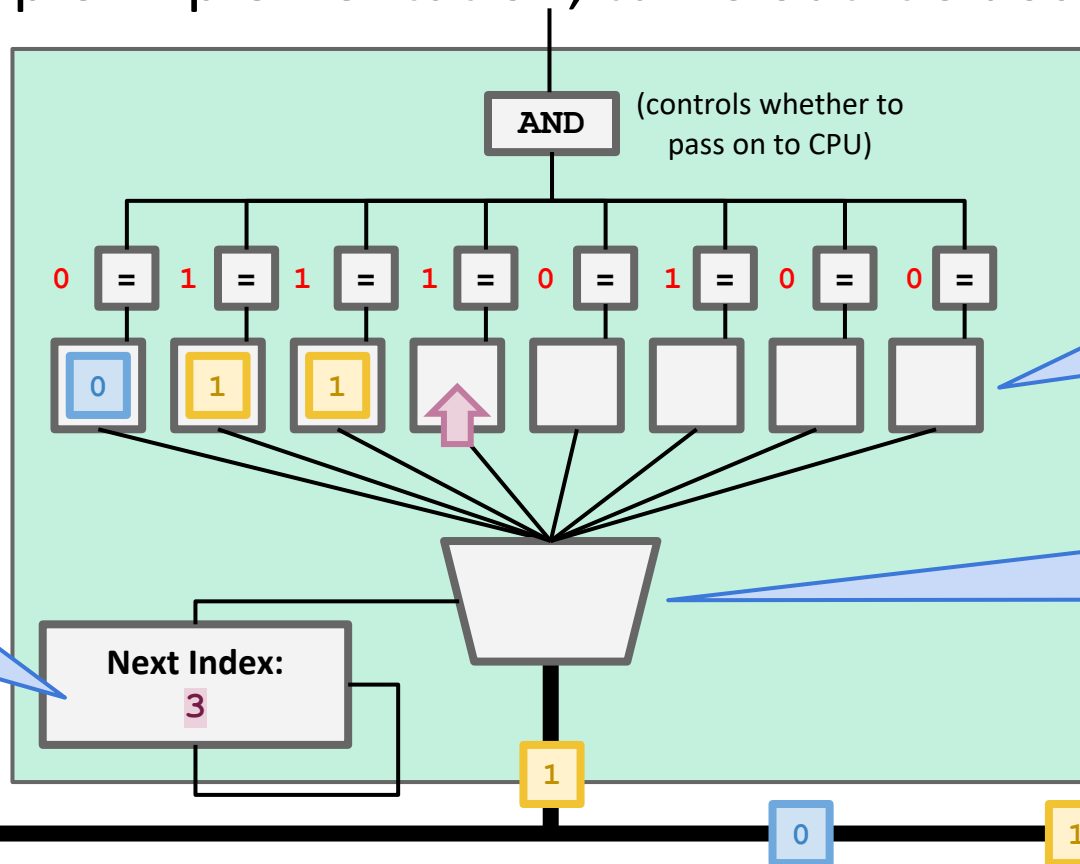
Constantly incrementing an address? Sounds like `PC.hdl!`

Selectable read/write?

Making a choice? Our good friend `DMux.hdl!`

NIC Implementation in Your Computer

- ❖ Before, we would have to accept NIC as “magic”
- ❖ Now, we can imagine exactly how to build this chip, and for a simple implementation, turns out it’s doable!



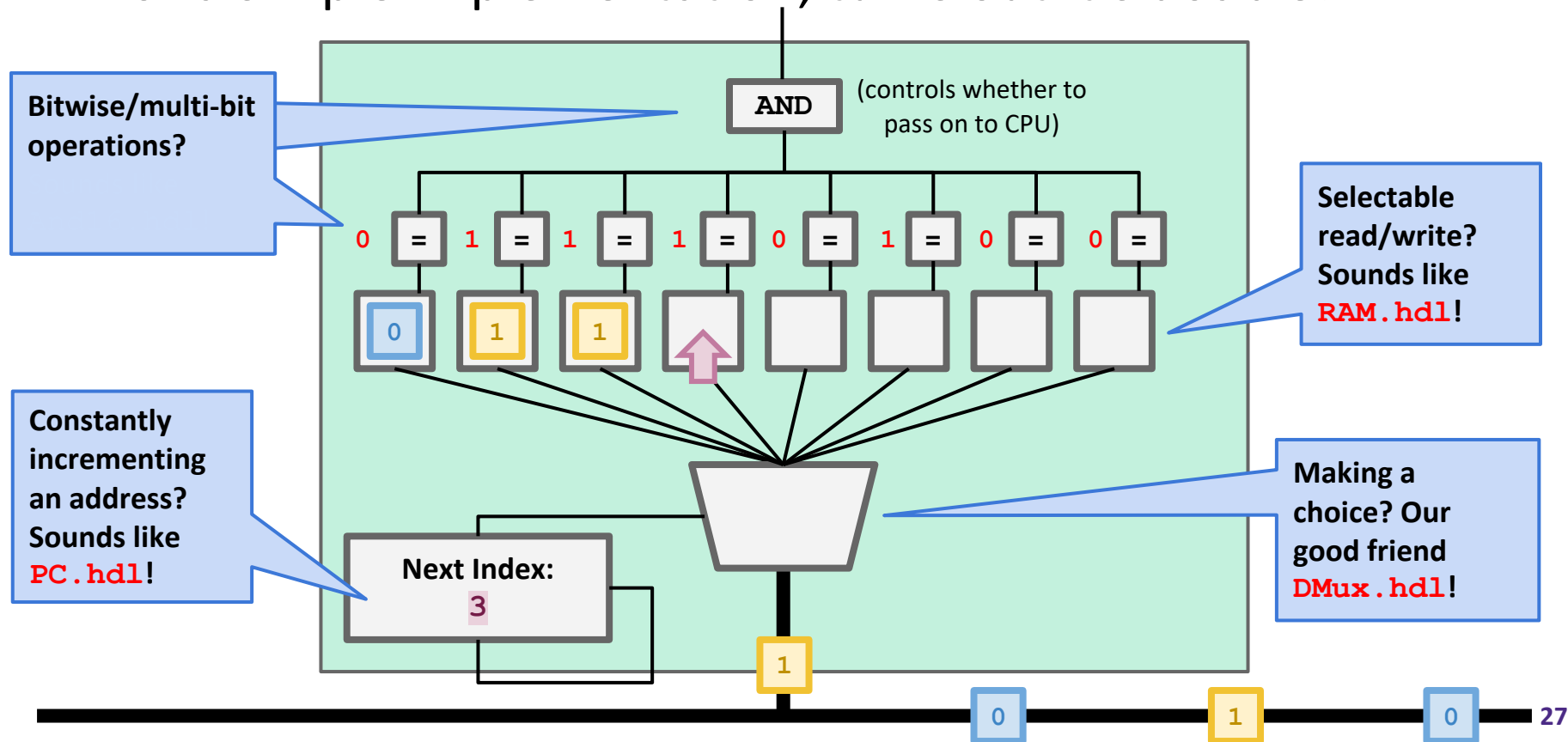
Constantly incrementing an address?
Sounds like **PC.hdl!**

Selectable read/write?
Sounds like **RAM.hdl!**

Making a choice? Our good friend **DMux.hdl!**

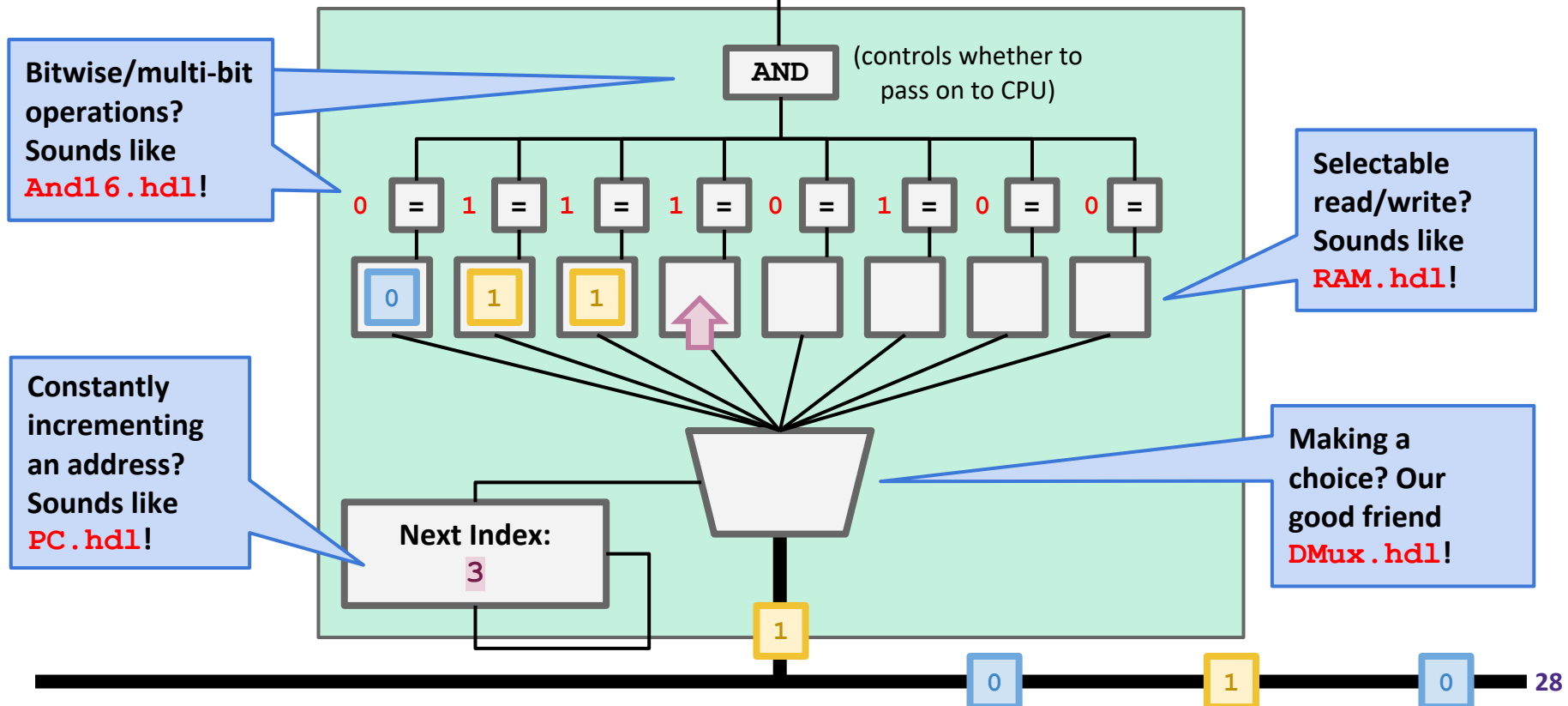
NIC Implementation in Your Computer

- ❖ Before, we would have to accept NIC as “magic”
- ❖ Now, we can imagine exactly how to build this chip, and for a simple implementation, turns out it’s doable!



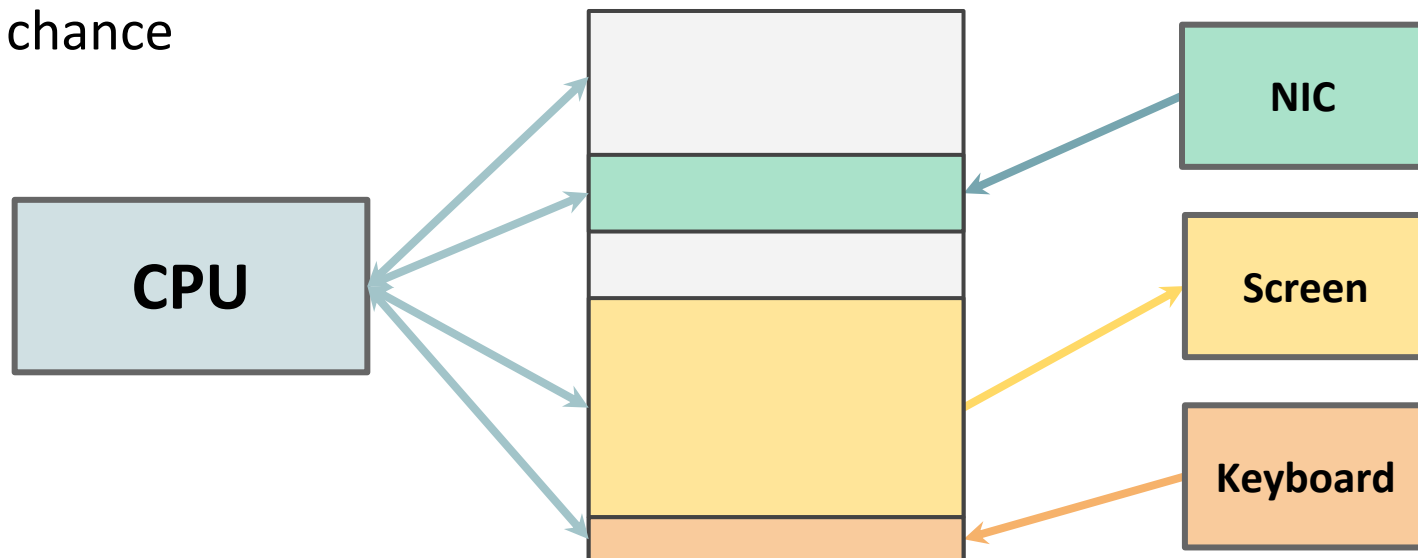
NIC Implementation in Your Computer

- ❖ Before, we would have to accept NIC as “magic”
- ❖ Now, we can imagine exactly how to build this chip, and for a simple implementation, turns out it’s doable!



Connecting NIC to Memory

- ❖ The keyboard and screen communicated with the CPU via memory maps—agreed-upon regions of RAM that can be read/written by the hardware of the devices themselves
- ❖ The NIC could be implemented in the same way
 - Every time the right address is detected, copy the following data into part of RAM where the CPU can retrieve it once it gets a chance



Takeaways: Computer Networks

- ❖ The network is fundamentally the same hardware we've been looking at
- ❖ Its incredible power comes from scale: how much data and how many computers it connects
 - To manage this complexity, we think of it in layers
- ❖ Interfacing with the network can be done with specialized hardware
 - This frees the CPU from monitoring constantly
 - Access data only when needed

Post-Lecture 18 Reminders

❖ Project Reminders

- **Project 7, Part II: Professor Meeting Report due tonight (5/25) at 11:59pm**
- Project 8: Debugging & Implementing a Compiler due next Tuesday (5/30) at 11:59pm
- Final Project, Part I: E-Portfolio Outline due next Tuesday (5/30) at 11:59pm

❖ Anam has office hours after class in CSE2 121

- Feel free to post your questions on the Ed board as well