Program Counter

Let's make it count this time

Akilesh Praveen — CMSC398E

UMD

April 21, 2020

Agenda

1. Announcements
   - Projects 5, 6, and 7
2. Intro + Background
   - What we need

Announcements

Projects 5, 6, 7

- Projects 5, 6, and 7 are now released on Piazza
- Relevant instructional material is/will be linked
- They can be done in **any order**, but I would suggest doing them in order (5, then 6, then 7)
- We already did a lecture on Project 5, today we'll be talking about **Project 6**

Intro

Intro

- We've built the ALU; the brains of the operation
- Now we need a few more things to take this from just a calculator circuit to an actual computer
  - Ways to **store** programs
  - Ways to **interpret** those programs
  - Ways to **execute** those programs
  - Ways to **store data** for those programs while they're executing
- We're going to use the digital logic circuit theory to build circuits to address all of these! (Projects 5, 6, and 7)

Intro

- Ways to **store** programs - **ROM** *(Project 5)*
- Ways to **interpret** those programs - **389E Assembly** *(Project 5)*
- Ways to **execute** those programs - **Program Counter** *(Project 6)*
- Ways to **store data** for those programs while they're executing - **RAM** *(Project 7)*
- Today, we'll be talking about ways to execute these programs, using a **Program Counter**.

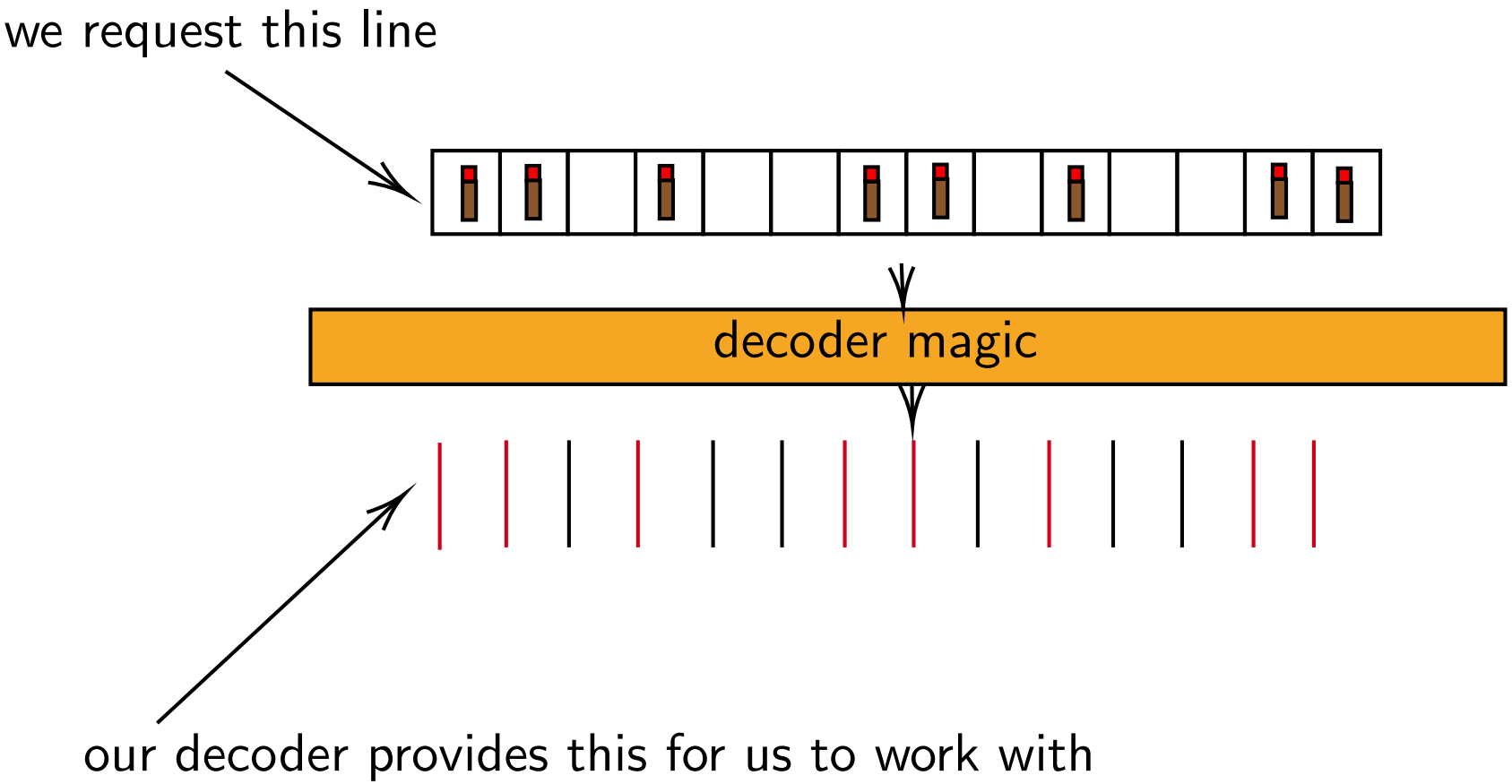
Background

The Story So far

- So far, let's take a look at the system we've got

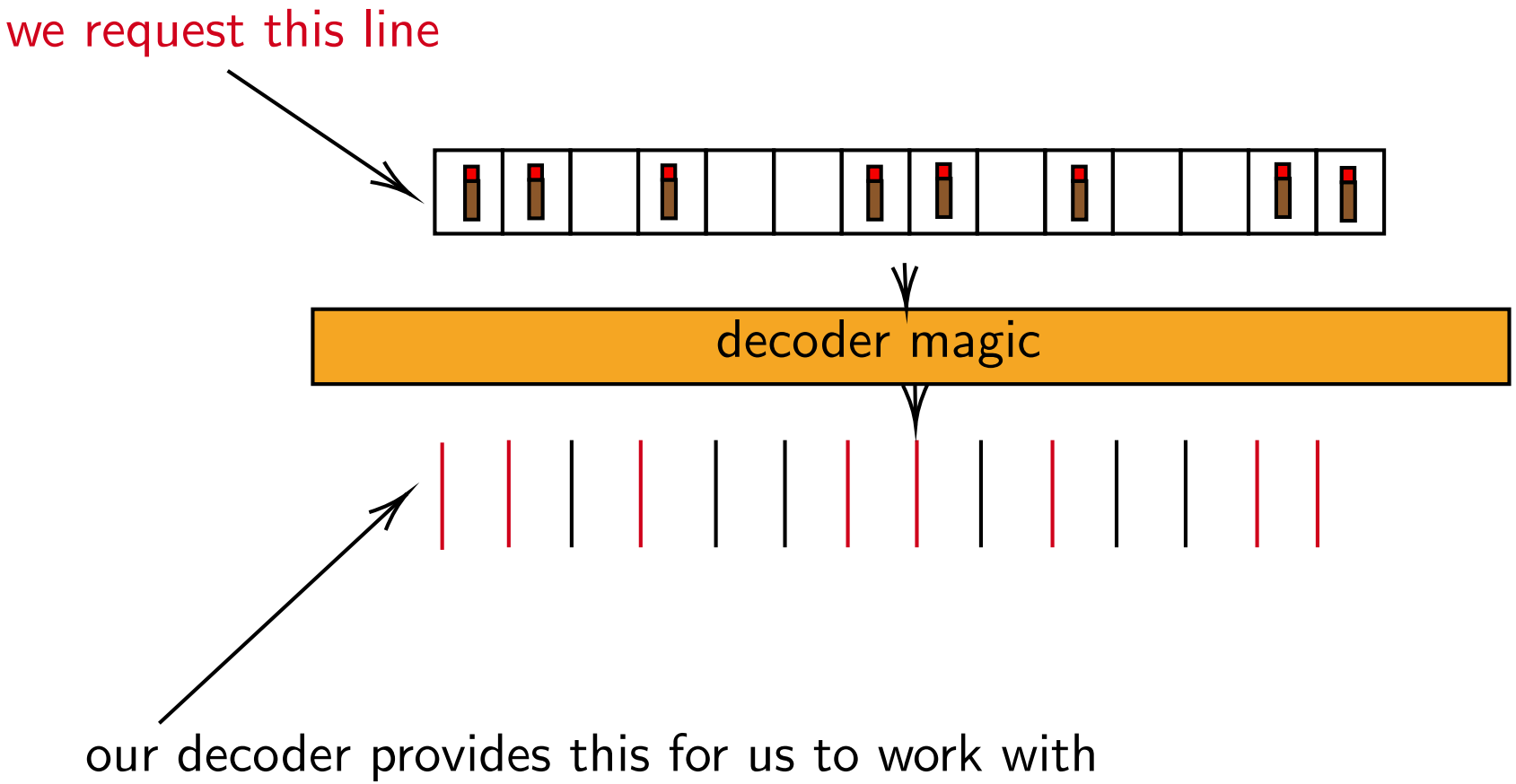
The Story So far

- So far, let's take a look at the system we've got
- I've taken the liberty of black boxing the components



The Story So Far

- Here's what we care about right now.



Requirements

- So we know that we need a way to **request lines**.
- We also know that such a request has to be input in **binary**.
- *If you don't know this already, make sure you know why.*

Sequentiality

- Ok, let's think about the **order** in which we request them
- After all, after we request one line, we're going to need a way to request the next.
  - Then the next after that, then the next after that, etc.
- This is where the idea of sequential logic comes in.

Sequentiality

- And, we aren't just thinking in terms of moving one by one.
- After all, is 1 the only amount we're ever going to be incrementing by?
- What about statements like `BRANCHEQ` in our 389E Assembly? How will we handle that?

Requirements

- Let's not get too confused with it all right now, we're just outlining specifications
- We've pondered enough, let's quantify our goals
  - What we've **got so far**
  - What we **need**

So What've We Got?

- Excellent ROM Architecture that takes in a **binary number** as a signal and produces the corresponding **line of code**

What Do We Need?

- We need a circuit that produces **line numbers** in **binary**.
- It needs to work **sequentially** (Produce a signal for 1, then 2, then 3, etc.)
- Additionally, it needs to be able to handle `BRANCHEQ` as well.
  - That is, it needs to be able to **increment and decrement by arbitrary quantities**, i.e. **jump** from one line to another.
  - For example, if we needed to 'jump' from line 2 to 5, this could be handled by incrementing our counter by 3.

In Relation To..

- Here's where our new circuit will belong among our other planned components. Again, we're working on the orange one.