Multiplexers & Demultiplexers

Why use many wire when few wire do trick

Akilesh Praveen - CMSC389E

UMD

March 5, 2020

Agenda

1. Announcements
   - Project 4
2. Multiplexers & Demultiplexers
   - Introduction
   - What are Muxes and Demuxes?
   - Muxes and Demuxes In-Depth
   - Applications
   - Project 4

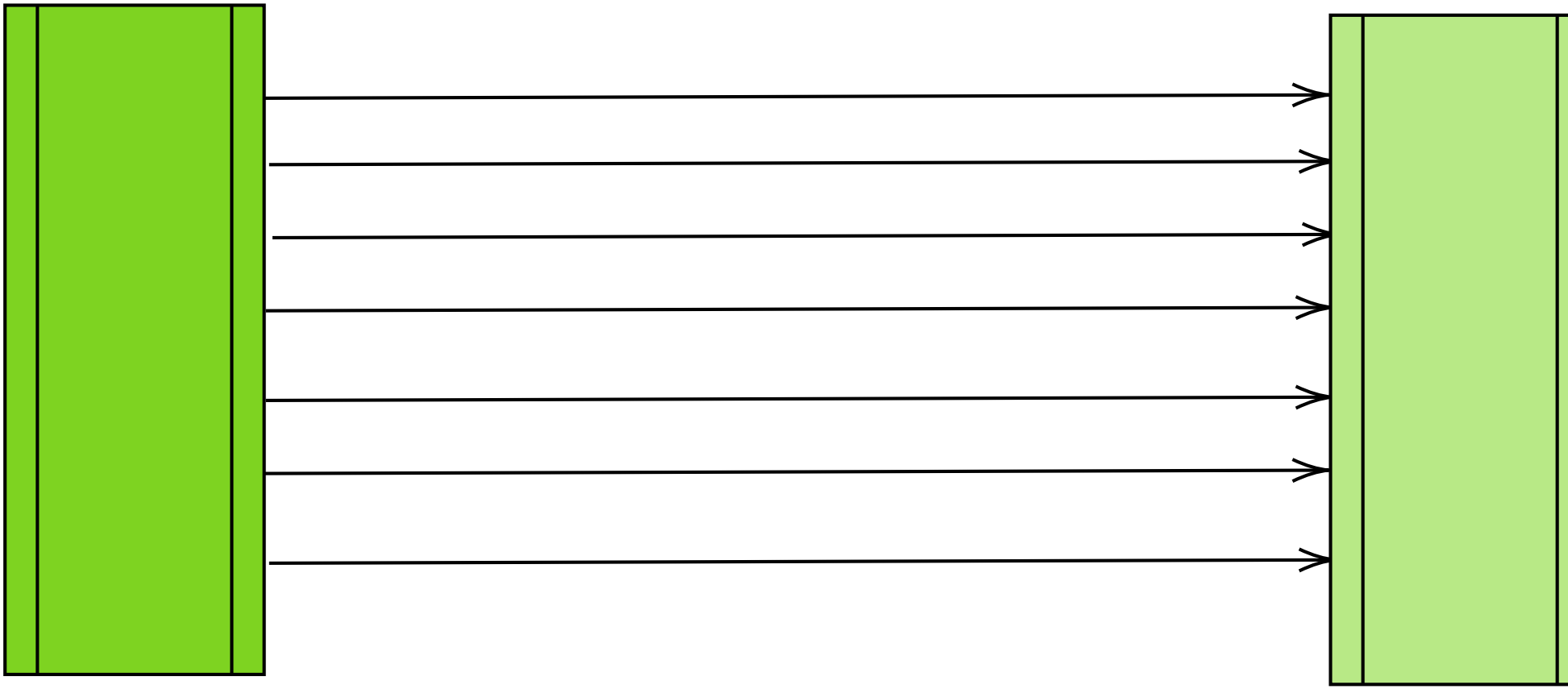
Announcements

Project 4

- Project 4's been released, and the name of the game is multipliers.
- You'll find everything you need under the 'week 5' section on the course website
- Today's lecture will give you background knowledge that will help you implement multiplexers and demultiplexers.
- More info to come at the end of lecture

Muxes and Demuxes

An example



- We've started creating circuits that have a sizable amount of wires associated with them
- It turns out, extending **all** of these wires is pretty expensive and space-consuming, given just how many wires we have to build

Parallel Signals

- This phenomenon is known as having many **parallel signals**.
- They use a ton of wires!
- I'm sure everyone is well aware of how much more effort it is to place one extra line of wires in Minecraft, which makes you wonder how much harder it is to do in real life
- If we're going to be extending these wires over long distances, are there more efficent ways to handle that many signals?

Parallel Signals

- Yes!!!
- Your answer can be found in **Multiplexers** and **Demultiplexers**.

Multiplexers & Demultiplexers

- A **Multiplexer** is often called a 'mux'.
- A **Demultiplexer** is often called a 'demux'.

Multiplexers & Demultiplexers

- A **Multiplexer** is often called a 'mux'.
- A **Demultiplexer** is often called a 'demux'.
- The question is- what are they?

Need necessitates solution

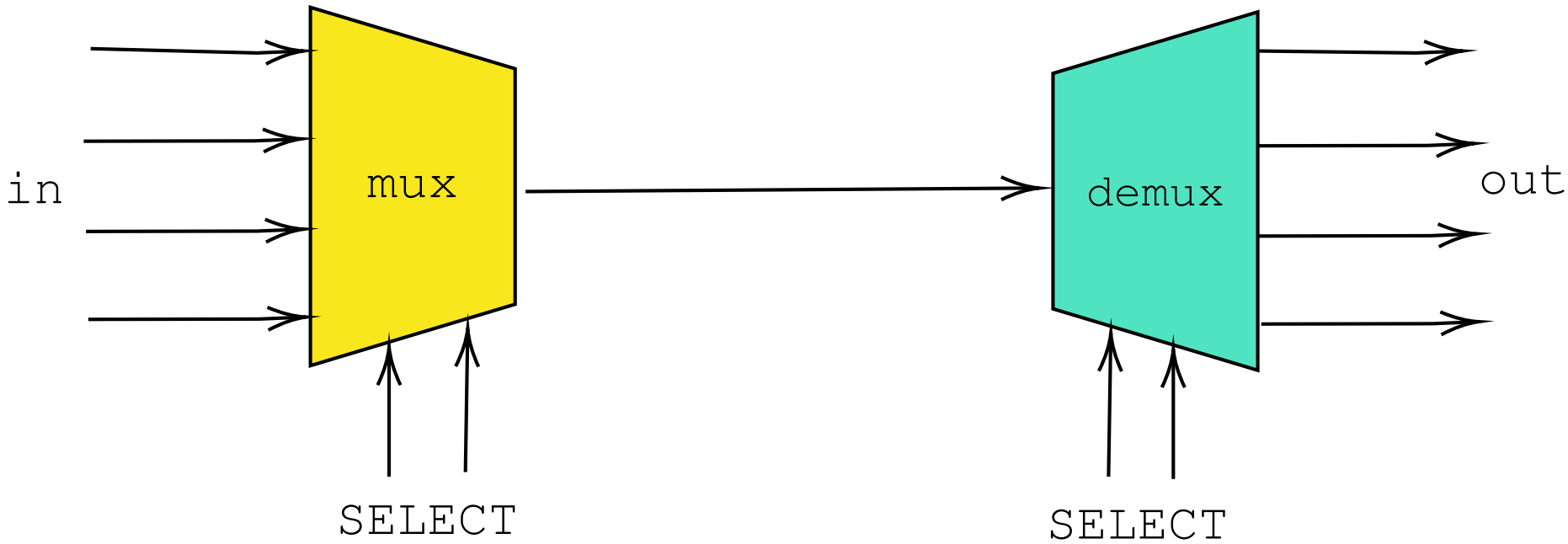
- The need for **Multiplexers** arose due to the presence of a certain problem in digital circuit design.
- Think back to our bus designs, and our encoders and decoders
- It turns out, there are a lot of different circuits that would require us to have a fair amount of parallel wires.
- In other words, it's a **common problem** in digital logic design to have to deal with a multitude of **parallel signals**.

What are they?

- A **Multiplexer**, or **Mux**, takes a bunch of parallel wires and condenses all of them down to a singular `OUT` wire and a few `SELECT` wires.
- A **Demultiplexer**, or **Demux**, takes a singular `IN` wire and a few `SELECT` wires, and sends the `IN` signal down the appropriate output wire.

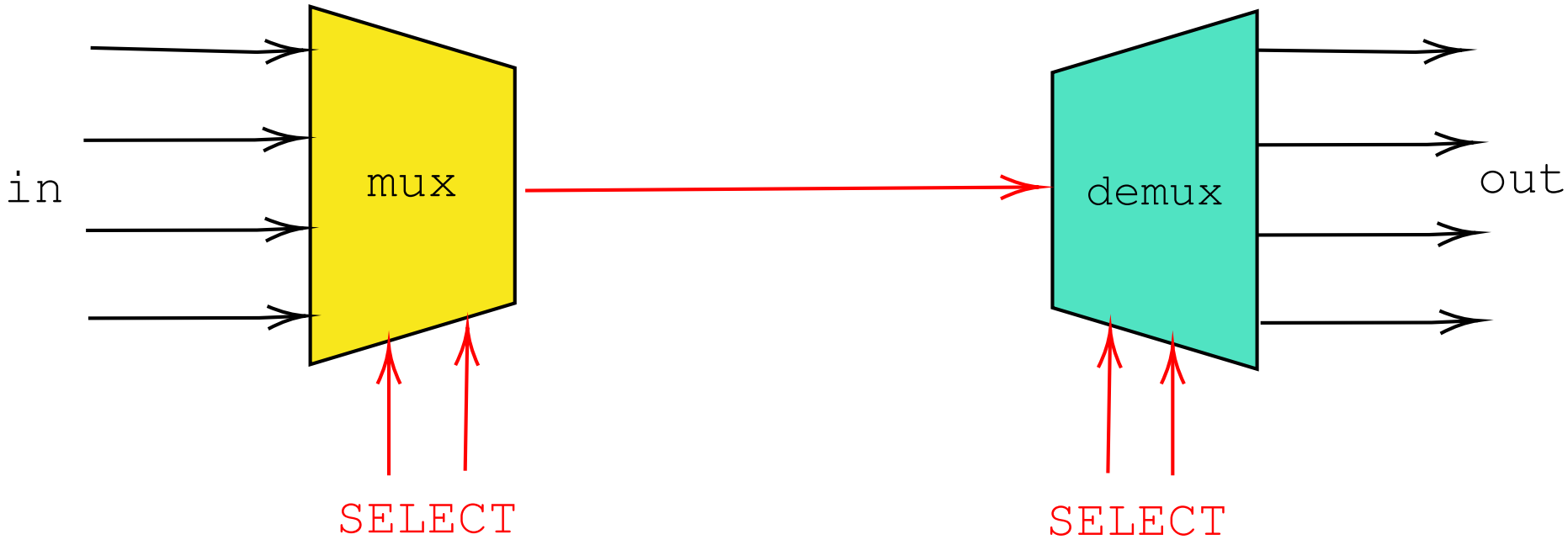
What are they?

- A **Multiplexer**, or **Mux**, takes a bunch of parallel wires and condenses all of them down to a singular `OUT` wire and a few `SELECT` wires.
- A **Demultiplexer**, or **Demux**, takes a singular `IN` wire and a few `SELECT` wires, and sends the `IN` signal down the appropriate output wire.



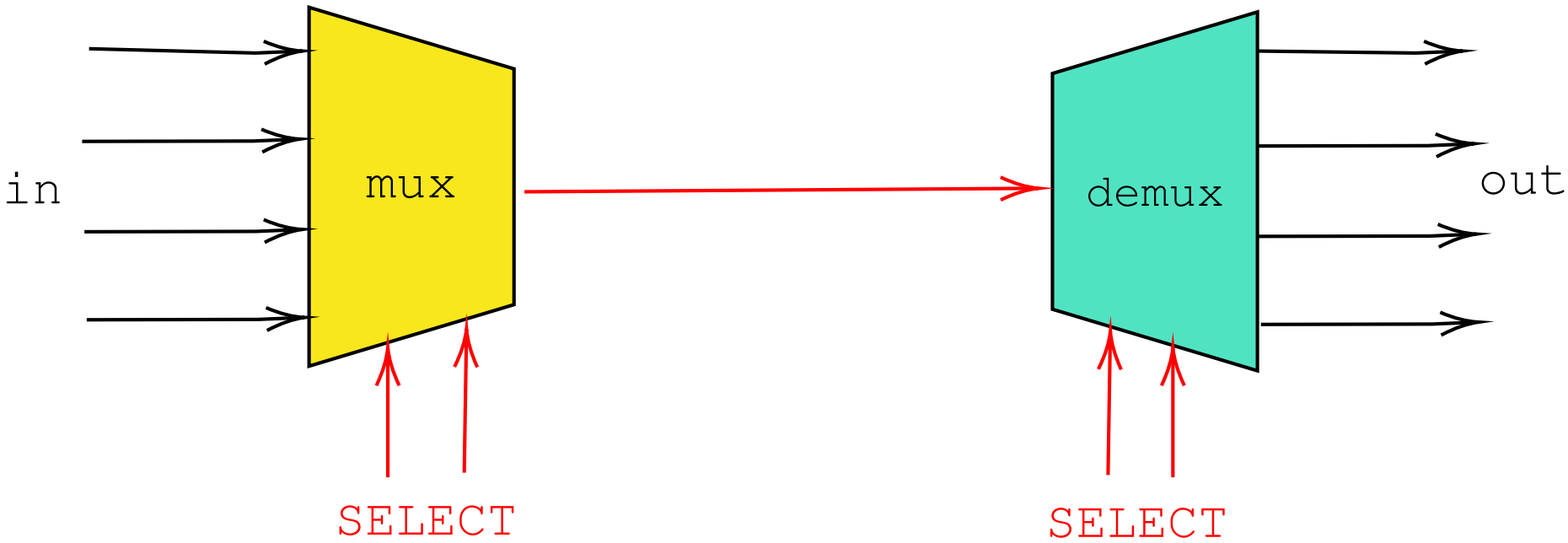
What can we observe?

- In the earlier example, instead of having to carry 4 wires across the distance spanned by the mux and demux, we only needed the singular output wire and the selector wires
- In other words, we only need to worry about carrying the wires highlighted in red across the distance spanned by the mux and demux



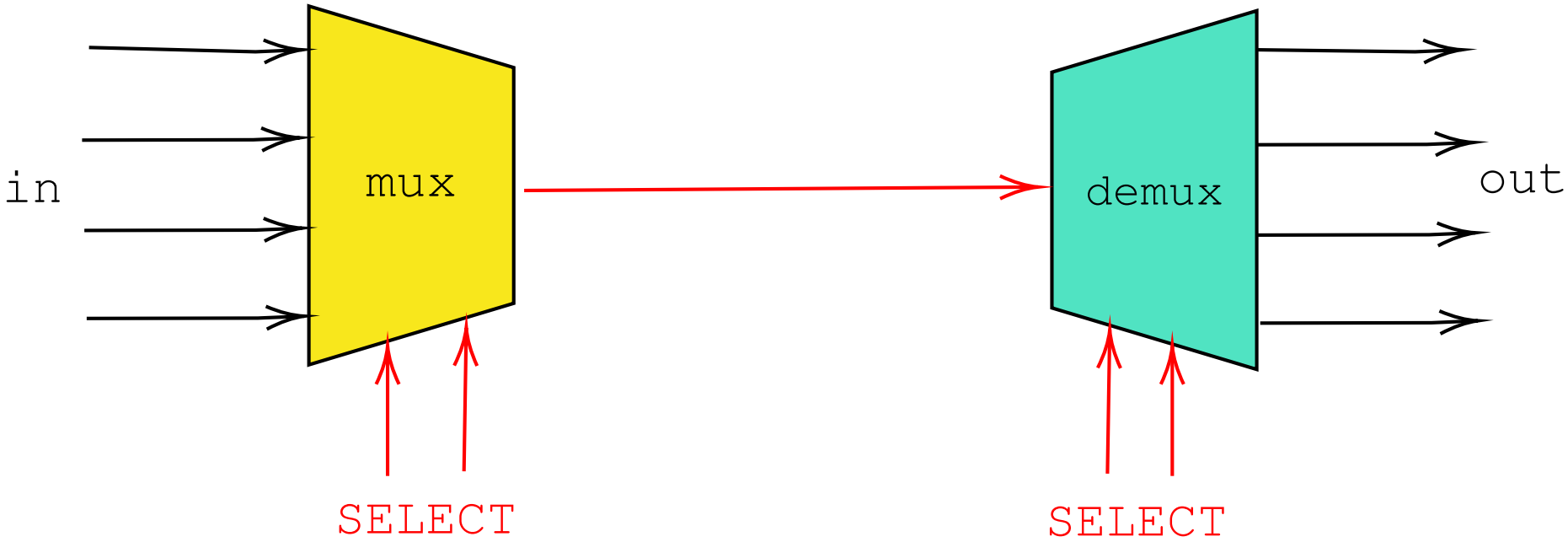
What can we observe?

- The OUT wire is simple enough- that's the signal that we ultimately want to transfer across
- As for the SELECT wires, they're a little more nuanced
- Can anybody guess why (if we have 4 inputs/outputs) we need two SELECT wires for our mux/demux combo?

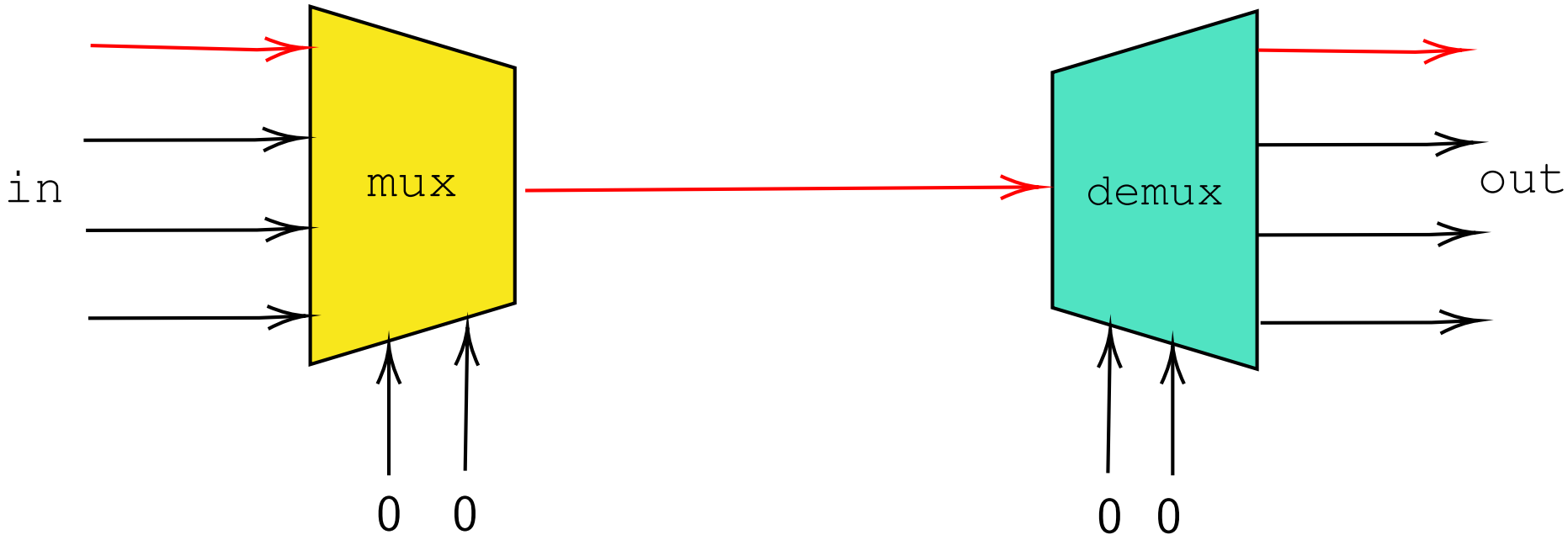


What can we observe?

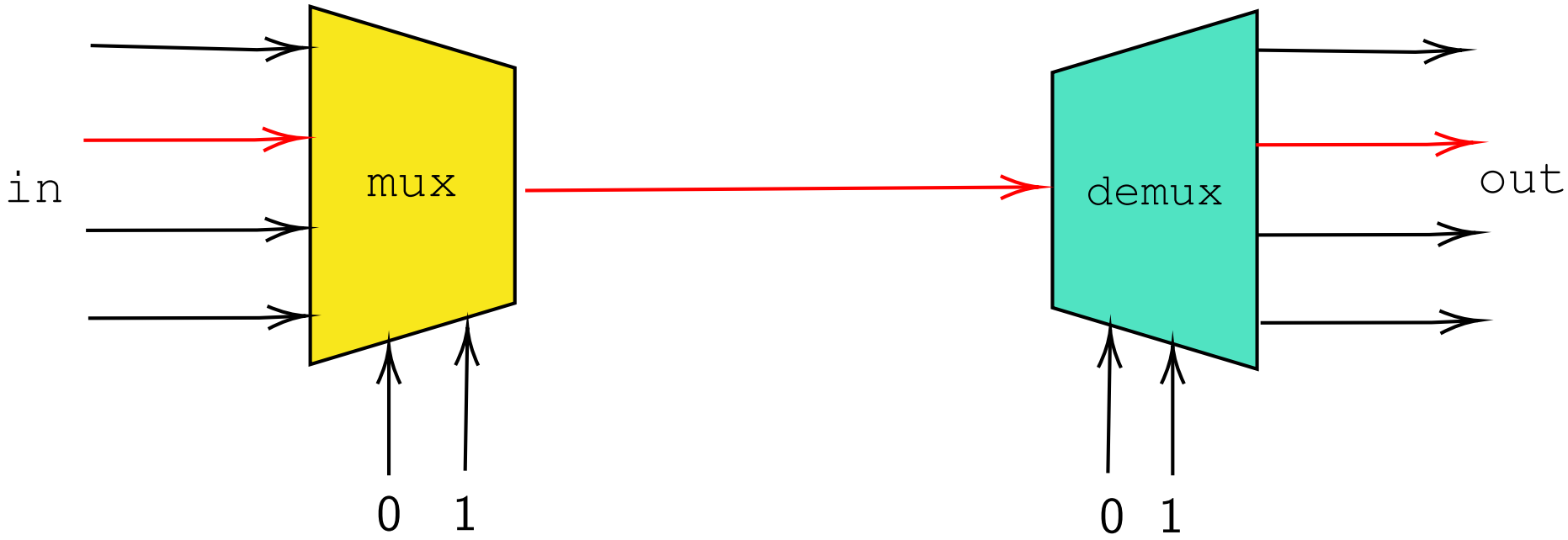
- If we want to signify which of the four inputs are being transmitted across the OUTPUT channel, we can go ahead and represent which one it is by using binary



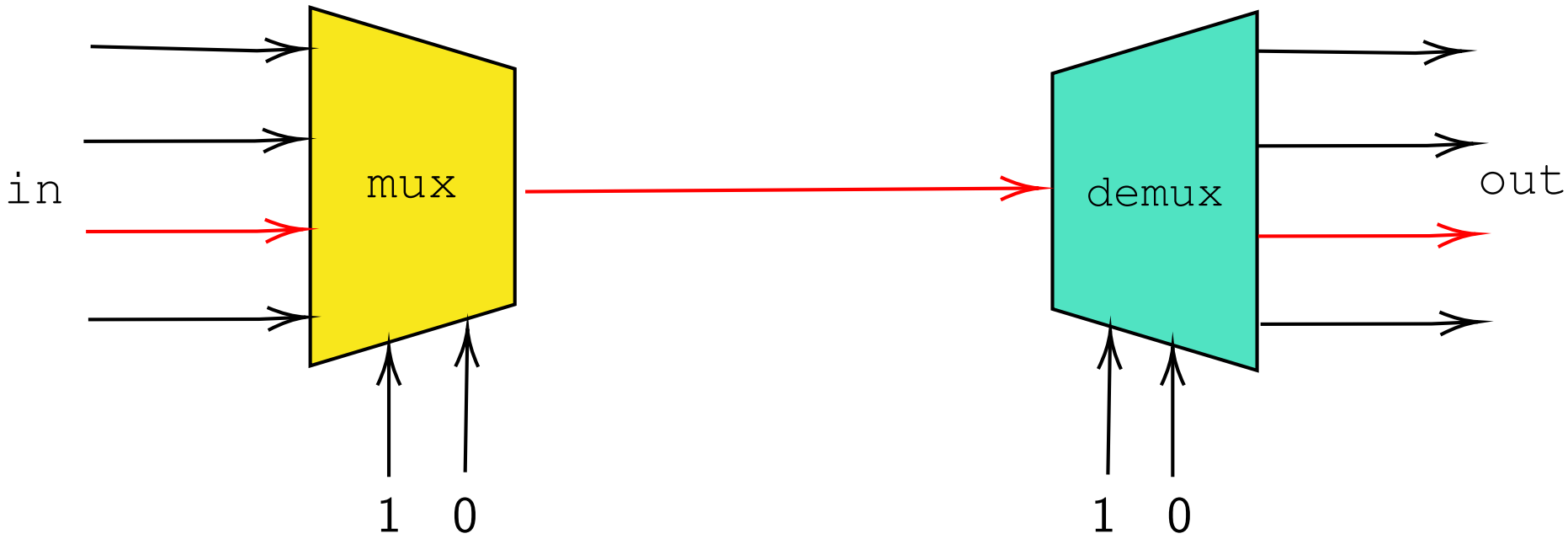
4×1 Multiplexer & Demultiplexer Example



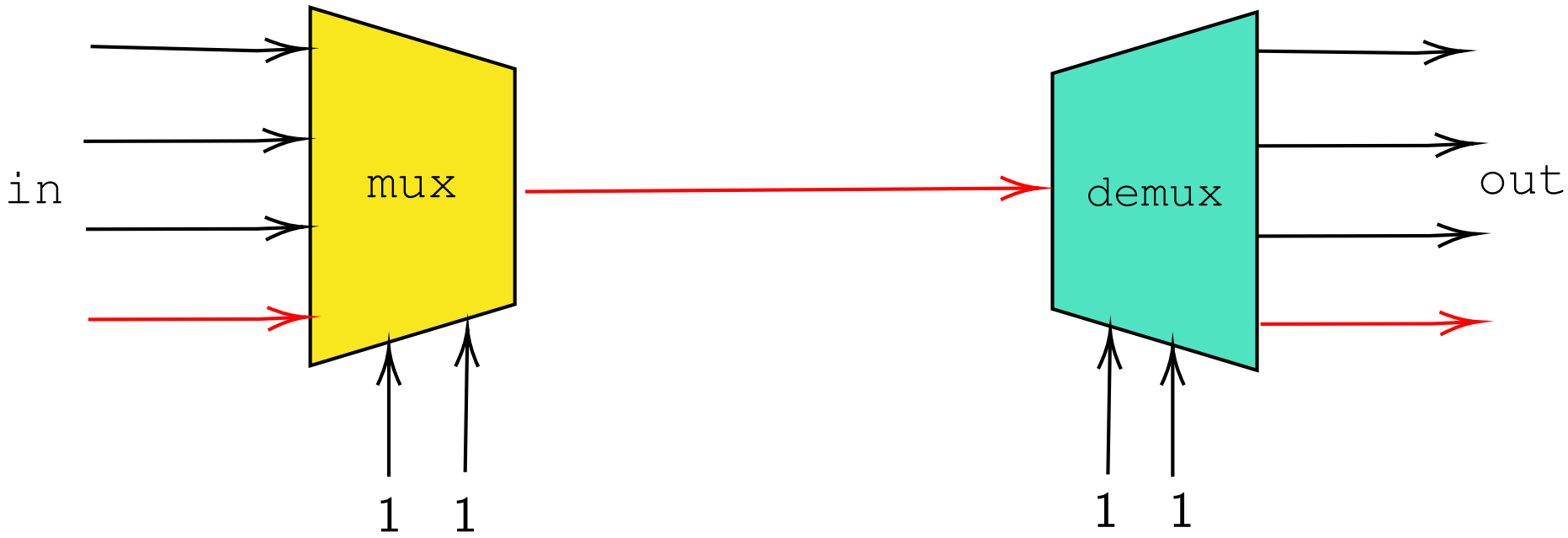
4×1 Multiplexer & Demultiplexer Example



4×1 Multiplexer & Demultiplexer Example



4×1 Multiplexer & Demultiplexer Example



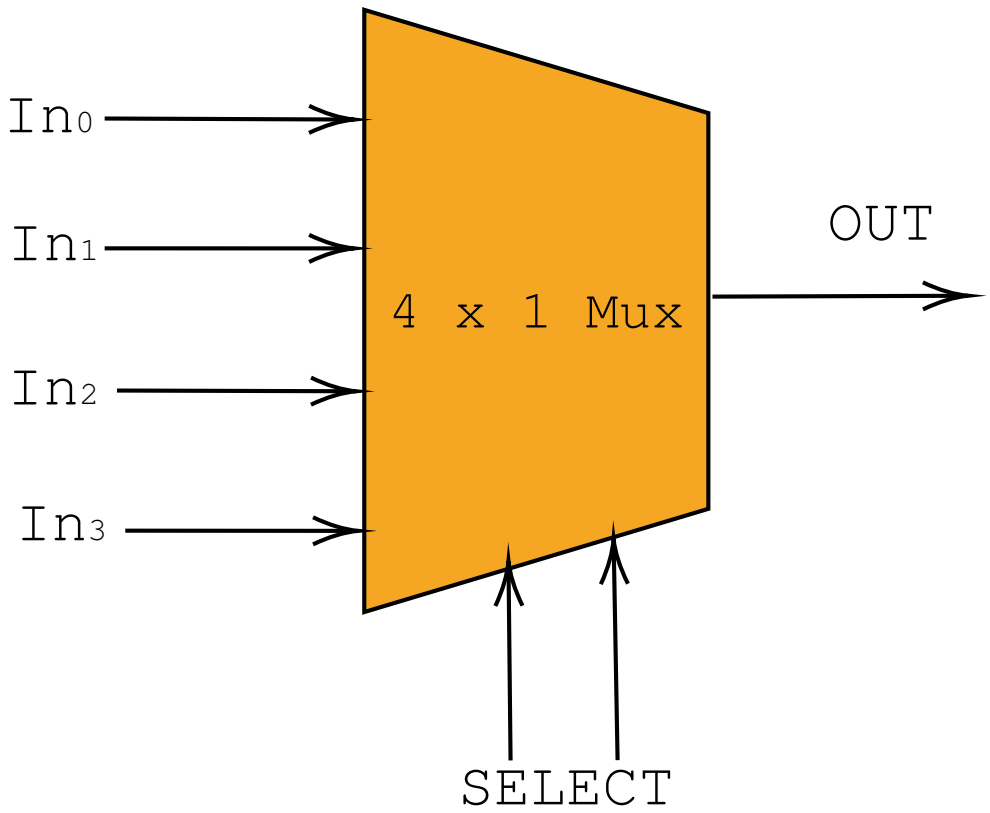
Immediate Observations

- We need less wires to carry signals across long distances (this is good!)
- We can only carry one signal across this channel at once (this may be bad!)
- In summary- it's all about tradeoffs

Muxes & Demuxes In-depth

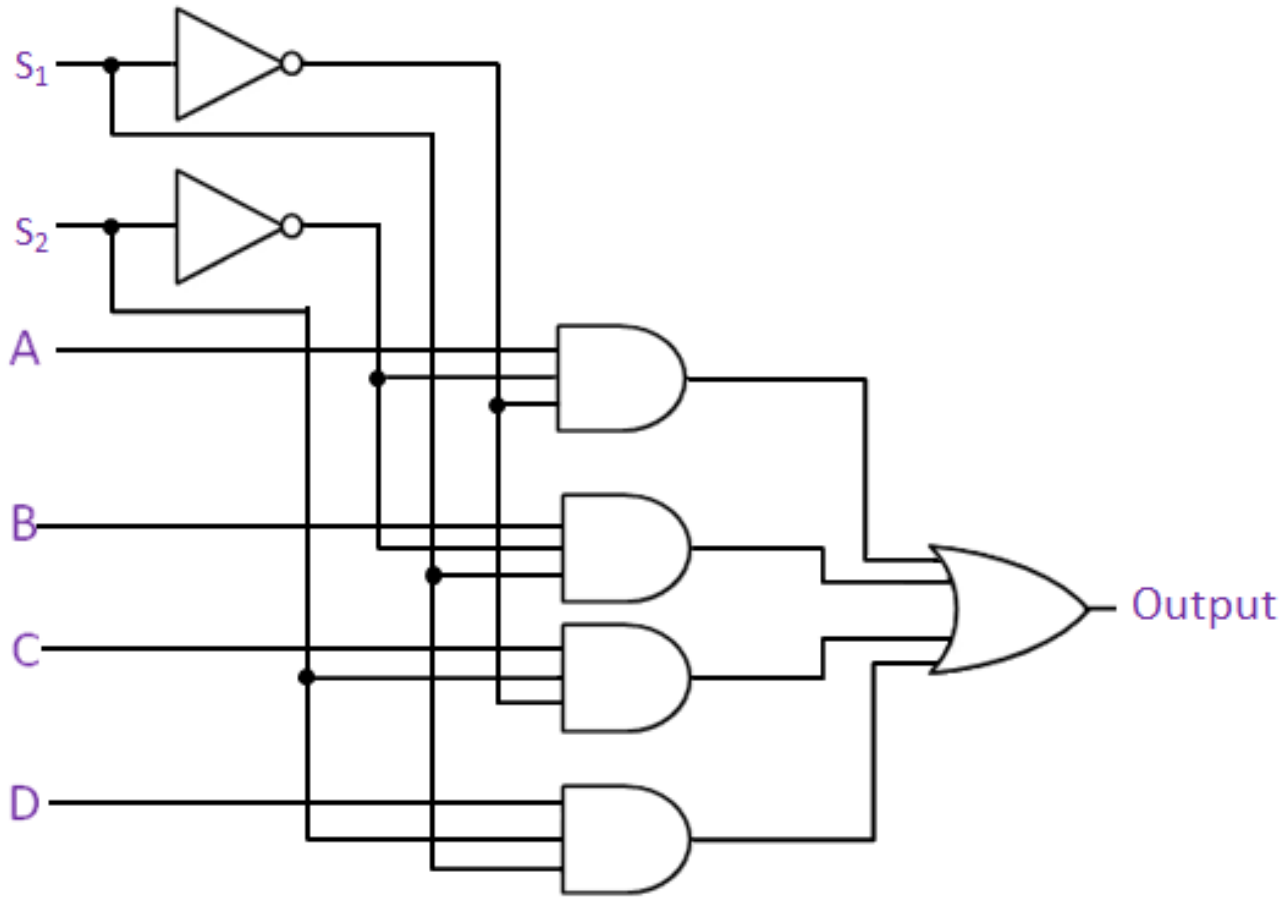
Multiplexer

- Now that we understand multiplexers conceptually, can you think of how you'd build it with the logic gates we've learned?



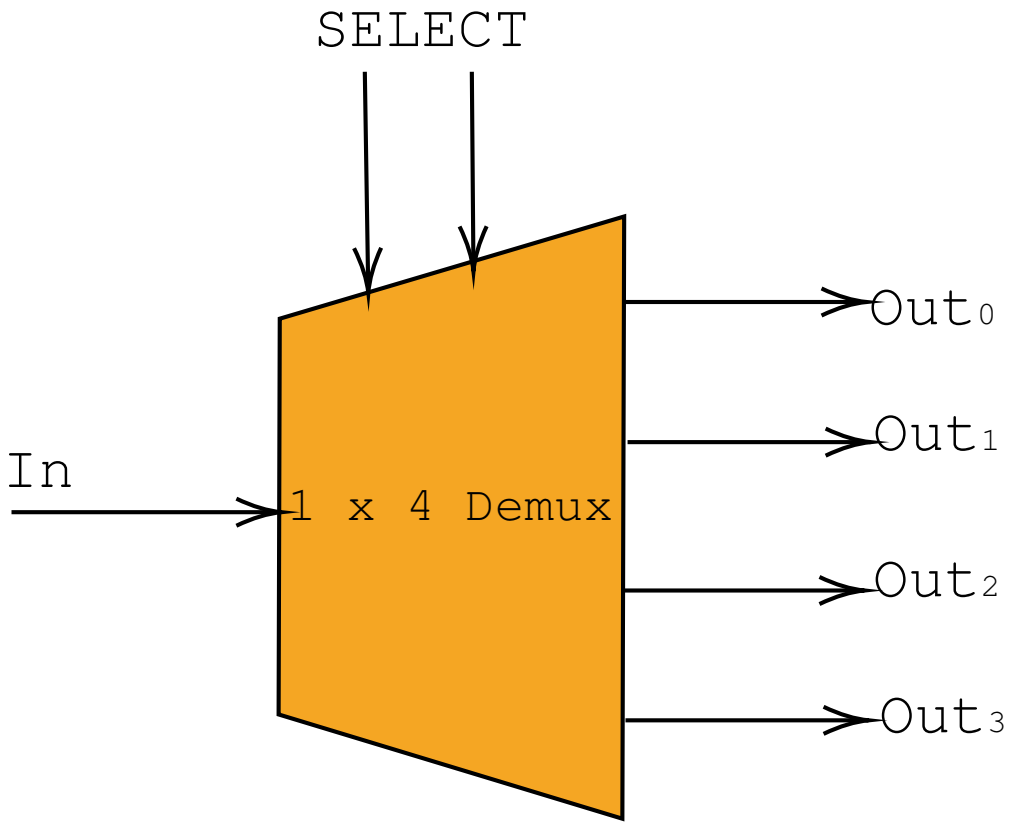
Multiplexer

- Here's the actual logical circuit



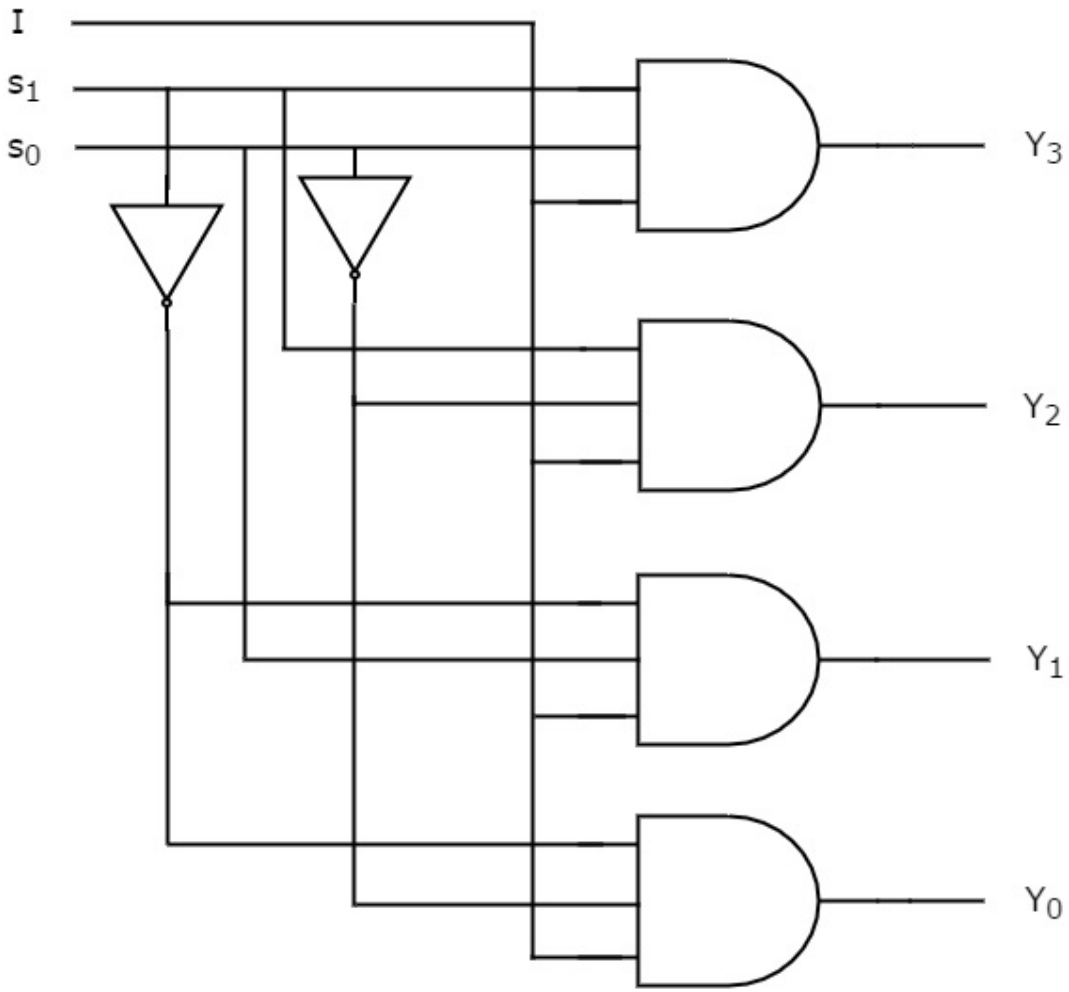
Demultiplexer

- Here's the complement, a 1×4 Demultiplexer. As you can imagine, demistifying the internal logic components is pretty similar to what we did for the mux.



Demultiplexer

- Here's the actual logical circuit



Applications

Applications of Multiplexers

- So how **useful** are these, anyway?
- As it turns out, there are clear mathematical reasons why

Applications of Multiplexers

- Consider the computer
- CPUs need to leverage **muxes** and **demuxes** to process instructions and data efficiently
- In particular, think of the ALU (what your projects have been, but way bigger)

Applications of Multiplexers

- We provide them for you in the projects, but consider where the inputs for these ALU operations come from
- If you think back to 216, the actual data comes from memory locations
- To keep the math from becoming utter spaghetti, let's talk about **32-bit** systems for now.

Applications of Multiplexers

- Knowing that an ALU performs crucial operations that depend on the inputs it gets, let's figure out how we'd access those inputs
- First of all, does anyone know how many memory addresses a 32-bit system would have?

Applications of Multiplexers

- Knowing that an ALU performs crucial operations that depend on the inputs it gets, let's figure out how we'd access those inputs
- First of all, does anyone know how many memory addresses a 32-bit system would have?

$$2^{32} = 4294967296$$

Applications of Multiplexers

- What would the inside of our computers be like if we had a little wire connecting each of those individual memory addresses to the ALU?

Applications of Multiplexers

Applications of Multiplexers

- Instead of having around 4 million individual wires, we can use a multiplexer
- How much would that cut us down to? (Remember, 32-bit architecture)

Applications of Multiplexers

- Instead of having around 4 million individual wires, we can use a multiplexer
- How much would that cut us down to? (Remember, 32-bit architecture)
- **We'd only need 1 output wire and 32 selector wires! (zoo wee mama)**
- Keep in mind that modern ALUs can do way more than just add, subtract, and do simple logic gate operations, so this sort of implementation is majorly useful for us
- P.S. How do you think it gets put back in memory?

Applications of Multiplexers

- Just food for thought- multiplexing and demultiplexing is hugely useful for us in other realms as well
- How do you think phone calls are made on landlines? Is there really a wire that connects your landline to every other phone in the world?
- Remember cable TV? (Emphasis on cable)
- As it turns out, multiplexing and demultiplexing has proved very useful for us within the digital logic realm and outside of it

Project 4