



Lab 3 Preparation

What this lab requires

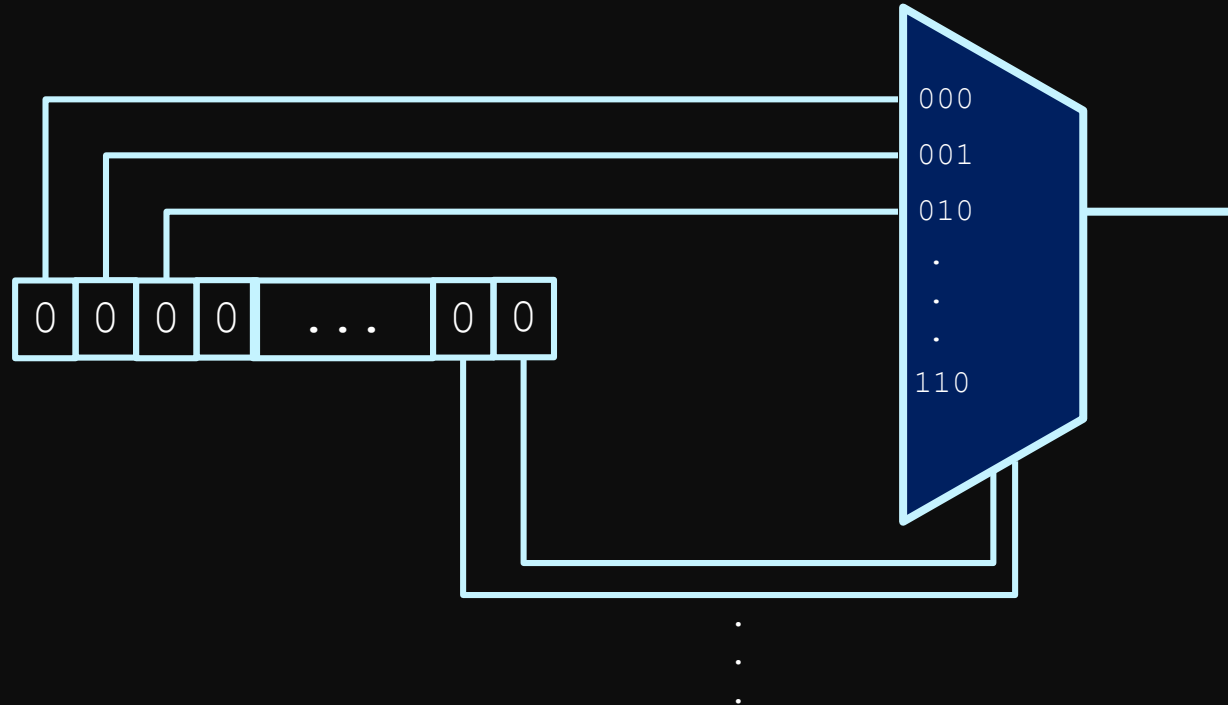
- More practice with Logisim, and explore some new components.
 - Arithmetic operators!
- Implementing some logical devices.
 - Full adder circuit, ALU.
- Learning some hierarchical design.
 - Mux + adders = ALU.

Part 1: Mux + Splitter

- In components, you can find mux under the category **Plexer**. Just drag and drop on your canvas.
- Key points:
 - In Properties of the multiplexer:
 - Adjust the number of select bits to change the size of the multiplexer.
 - Adjust the number of data bits to match the size of the input data.

Part 1: Mux + Splitter

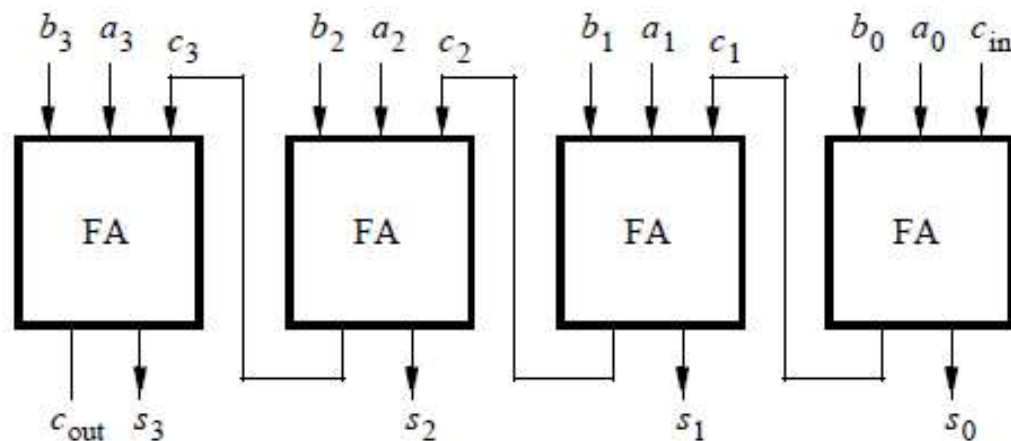
- What you're supposed to do:
 - ▣ Multi-bit input into 7-to-1 mux.



Part 2: Ripple Carry Adders

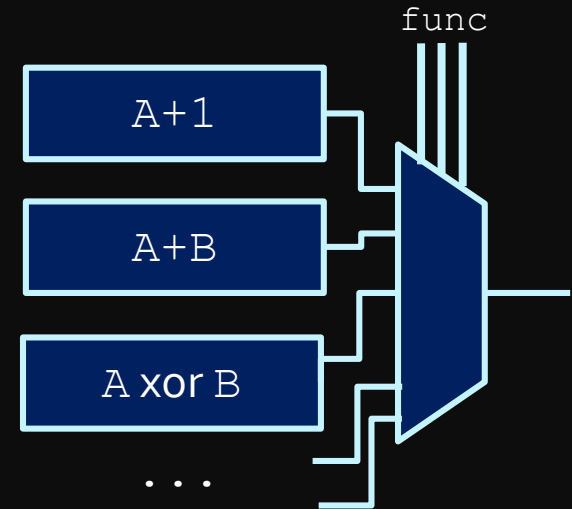
- Implement a Ripple Carry Adder by connecting (chaining) four full-adders together.
 - Must use hierarchical design!

b	a	c_i	$c_o \ s$	
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



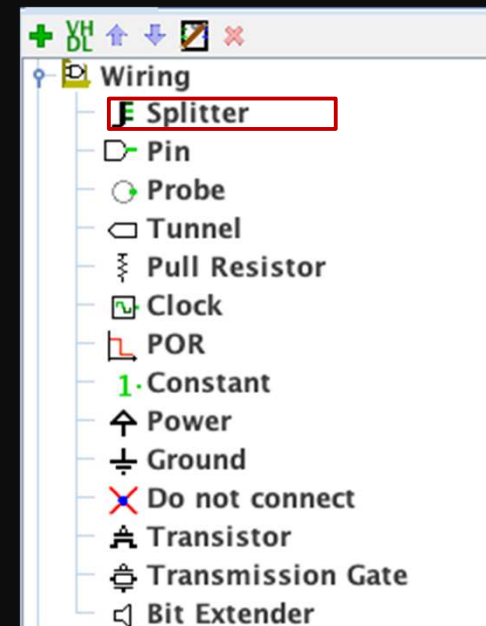
Part 3: ALU

- The ALU (**Arithmetic Logic Unit**) uses a mux to choose a single output value from a series of modules.
 - Each of these multiplexer inputs is connected to a module that performs a single arithmetic or logical operation.
- Once each operation module has been implemented with general inputs A and B , connect these ALU inputs to the switches and the ALU outputs to both the LEDs and 7-segment display (in hexadecimal).
 - Try to you reuse the modules you made in Lab 2.
 - The handout includes syntax for some new (and potentially useful) operators



Splitter in Logisim

- Splitter can be very useful in Logisim. You can find them under Wiring in components.
- They can be used in two ways:
 - Separating a multi-bit signal,
 - Concatenating signals together.



Bitwise Operations

■ Bitwise Operations

- If you use a bitwise operation with two n-bit operands, the result is also an n-bit vector.
- For example:

0101 & 0011 → 0001

- More general mathematical notation:
- $(X_{n-1}X_{n-2} \dots X_1X_0 \text{ \& } Y_{n-1}Y_{n-2} \dots Y_1Y_0)$ results in $W_{n-1}W_{n-2} \dots W_1W_0$ where W_i is $(X_i \text{ \& } Y_i)$ for every i in $[0, n-1]$.

■ In Logisim, this can be achieved using Splitter.

- For a bitwise AND:
 - Split the signal into separate bits,
 - Use AND gates on each pair of bits,
 - Concatenate the result bits together

Reduction Operators

- **Reduction Operations** have the same approach as bitwise operations, but
 - They take **a single** multi-bit operand, and
 - The result is a **single-bit output**.
- In Logisim, we achieve this using splitters as well.
 - For reductive AND:
 - Split the signal into individual bits
 - Connect these bits to the inputs of a single AND gate

Replication and Concatenation

- The binary value 011 (3 in decimal) is the same as 0011 or 000000011.
 - Adding zeros in the most significant bits of a positive or an unsigned number does NOT change the number being represented!
 - Example:
 - If the output of a module is 3-bits and you want to feed it to a 5-bit input of another module, you'd need to use both replication & concatenation!
- In Logisim, this is achieved using **Bit Extender** under Wiring in components.