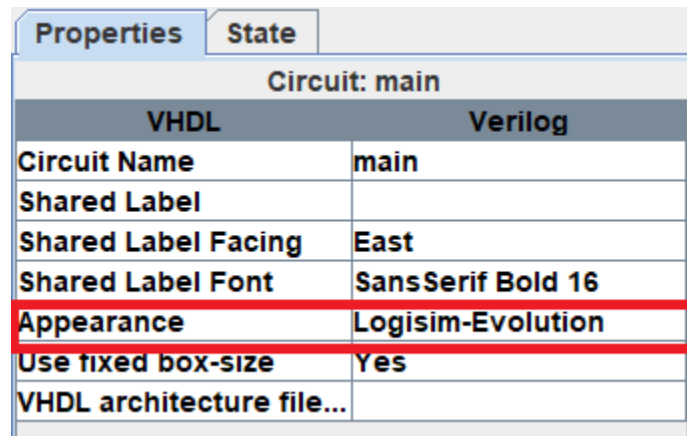


# ECS 154A: Homework 2

## Logisim

1. Use the [version from the class Google Drive of Logisim Evolution](#). Other versions may not work correctly.
2. Do not rename the files you receive. If you do so you will automatically fail the tester when you submit.
3. Put your solution for each problem into implementation subcircuit
4. Do not rename the implementation subcircuit anything else. If you do so you will automatically fail the tester when you submit.
5. Do not change the appearance of the implementation subcircuit from what it is set as. Doing so will cause you to automatically fail the tester when you submit.
  - a. That is this field right here



Properties		State
Circuit: main		
VHDL		Verilog
Circuit Name	main	
Shared Label		
Shared Label Facing	East	
Shared Label Font	SansSerif Bold 16	
Appearance	Logisim-Evolution	
Use fixed box-size	Yes	
VHDL architecture file...		

6. Do not move the pins inside of the implementation subcircuit as that affects the appearance of the circuit on the outside as you saw in discussion. Doing so will cause you to automatically fail the tester when you submit.
  - a. If you want to “move the pins” instead connect tunnels to the pins and move the tunnels around.
7. Do not name any of the subcircuits in your solution main. Doing so will cause you to automatically fail the tester when you submit.
8. You **can** create as many other subcircuits as you want in your solution. Just make sure your solution ends up in the implementation subcircuit

## Restrictions

For all problems in this homework, you may only use

- All of the components under the wiring folder
- AND, OR, NOT, and XOR gates

Unless a problem specifies otherwise.

## Problem 1: adder4.circ (33 points)

Implement a 4-bit **prefix** adder. The adder you implement must be a **prefix** adder. If you implement any other adder, such as a ripple carry adder, you will **not receive any points**. The automated checking cannot tell the difference between adders so the TAs will look at your submission after the due date to validate that you implemented the correct type of adder. Again, if you did not, you will **not receive any credit for this problem**.

It should go without saying that you cannot use the builtin adder for this problem not only because it would defeat the purpose of this problem but also because it isn't one of the generally allowed components.

### Inputs

Pin	Size (in bits)	Explanation
Ain	4	The A in $A + B$
Bin	4	The B in $A + B$
Cinin	1	The incoming carry bit

### Outputs

Pin	Size (in bits)	Explanation
Sum_Out	4	The sum of $A + B$
Carry_Out	1	The carry coming out of the most significant bit from doing $A + B$

## Problem 2: shifter.circ (33 points)

Build a circuit that can logical/arithmetic left/right shift a value a variable amount.

### Additional Components

For this problem and this problem only, you can also use

- Muxes

## Inputs

Pin	Size (in bits)	Explanation
IsRightShiftIn	1	If 0 do a left shift. If 1 do a right shift.
IsArithmeticIn	1	If 0 do a logical shift. If 1 do an arithmetic shift
ShiftAmountIn	2	How much to shift NumberIn by
NumberIn	3	The bit string to shift

## Outputs

Pin	Size (in bits)	Explanation
ShiftedNumberOut	3	The result of the shift performed on NumberIn

## Hints

1. Don't focus heavily on Kmaps for this problem. An easier solution can be found by thinking of this problem at a higher level
2. Focus on each bit of the output one by one when trying to find an answer to this problem
3. Treat the multi-bit inputs as a bunch of bits instead of as numbers
4. Use muxes to solve this problem

## Problem 3: bitcounter.circ (34 points)

Create a circuit that counts the number of 1's or 0's in an 8 bit bit-string.

### Additional Components

For this problem and this problem only, you can also use

- All of the components under Plexers
- All of the components under Arithmetic **except for Bit Adder**

### Inputs

Pin	Size (in bits)	Explanation
BitsIn	8	The bit string whose 1's or 0's your circuit is to count
CountZerosIn	1	If 0 count the number of 1's in BitsIn. If 1 count the number of 0's in BitsIn

### Outputs

Pin	Size (in bits)	Explanation
BitCountOut	4	The number of 1's/0's in BitsIn

### Credit

Credit for this problem goes to Justin Perona.

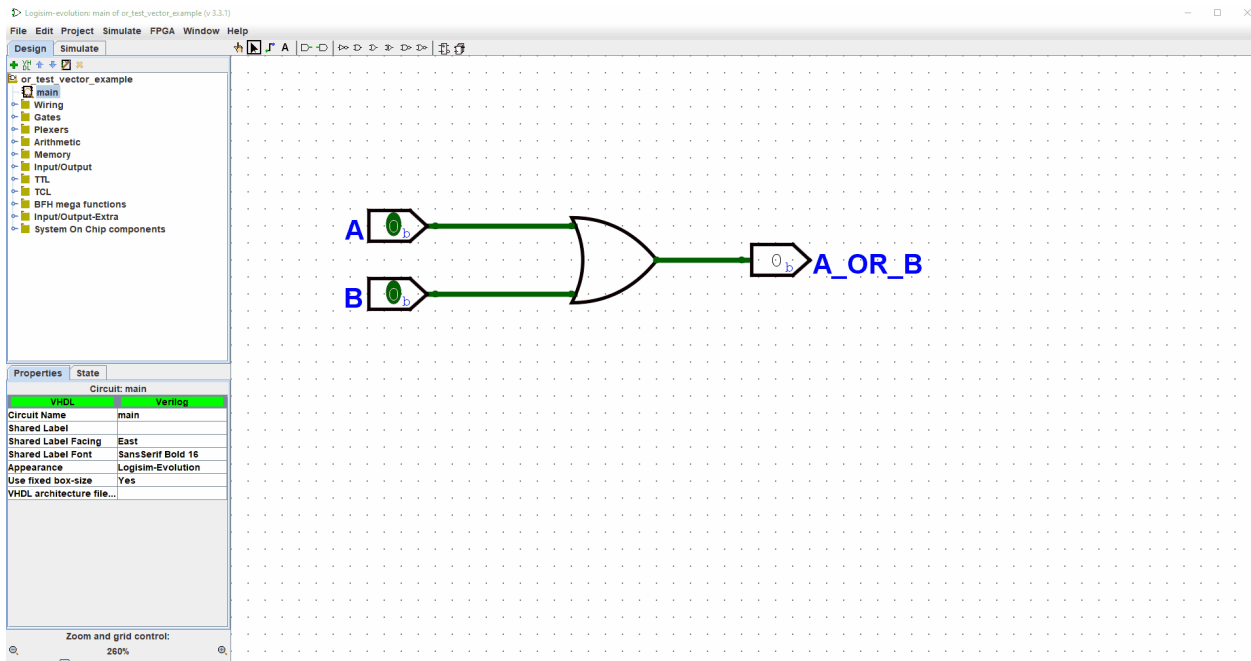
### Hints

1. Don't focus heavily on Kmaps for this problem. An easier solution can be found by thinking of this problem at a higher level

# Testing

Testing for this assignment should be fairly straight forward. All you have to do is load the included test vector for each problem into the implementation subcircuit. It will then tell you for each input whether the corresponding output is correct. If the test vector says everything is good you should pass all of the test cases when you submit as long as you didn't break any of the rules at the beginning of this document.

To run a test vector, select Simulate -> Test Vector. Then in the pop-up window select Load Vector.



# Submitting

## Submit to

Logisim Homework 1 on GradeScope.

## What to Submit

Submit a zip file that contains the following .circ files

1. adder4.circ
2. shifter.circ
3. bitcounter.circ

Inside of each .circ file leave a comment with you and your partner's names.

Make sure that you submit a zip that contains the files and **NOT** the **folder containing the files**. Check out the animation below for what to submit.

