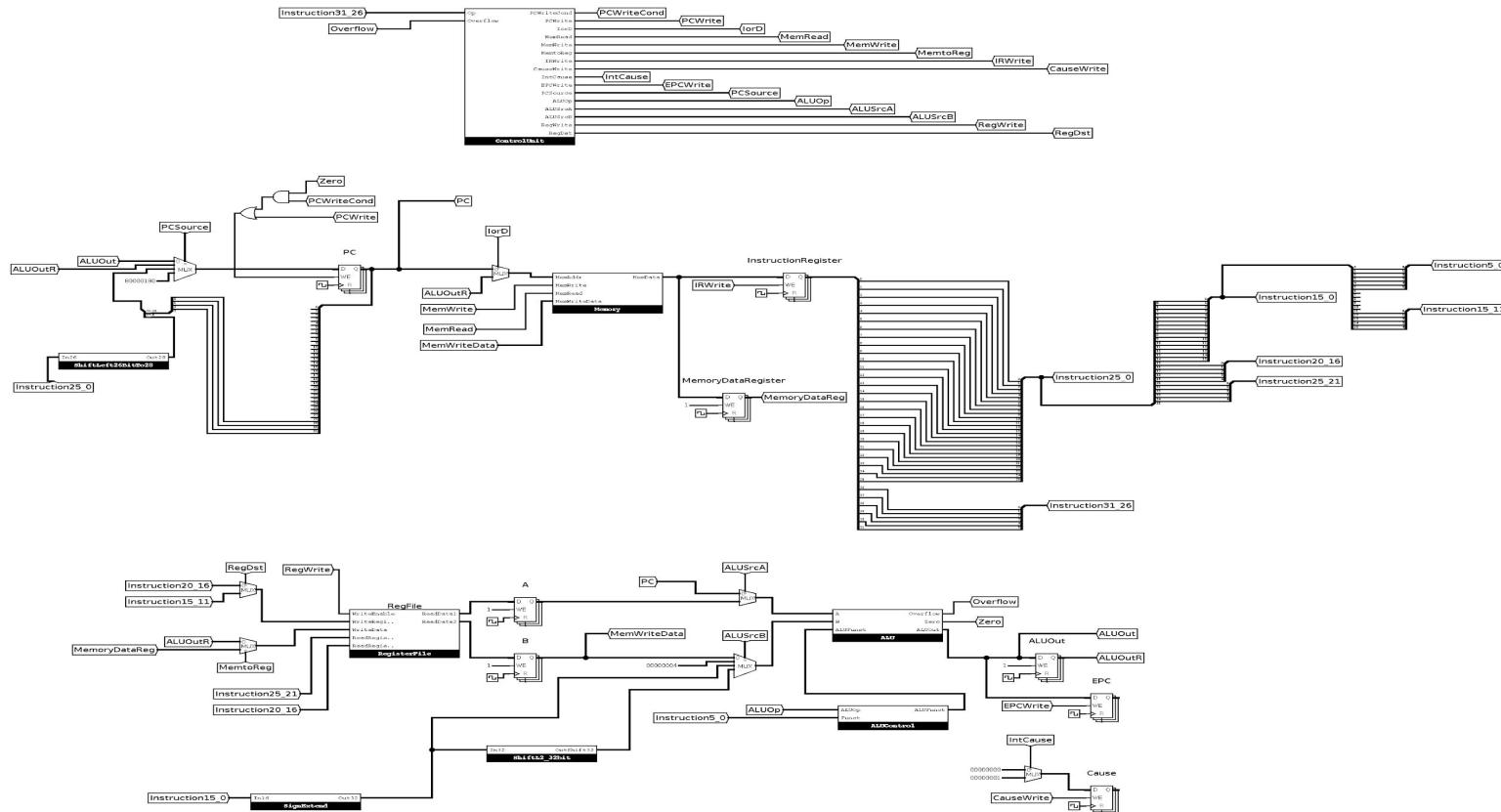


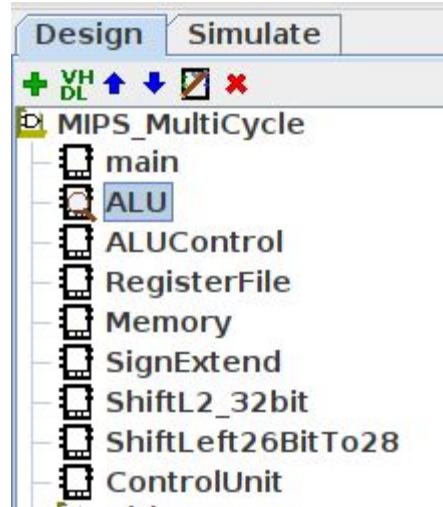
# Multicycle datapath in Logisim

# General structure



# Sub-components

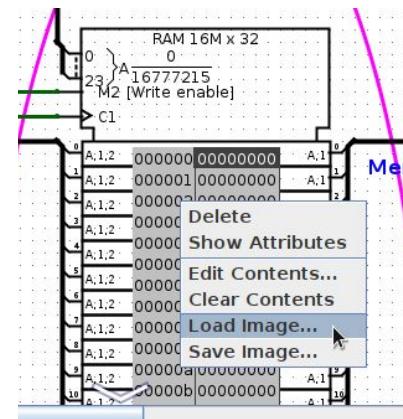
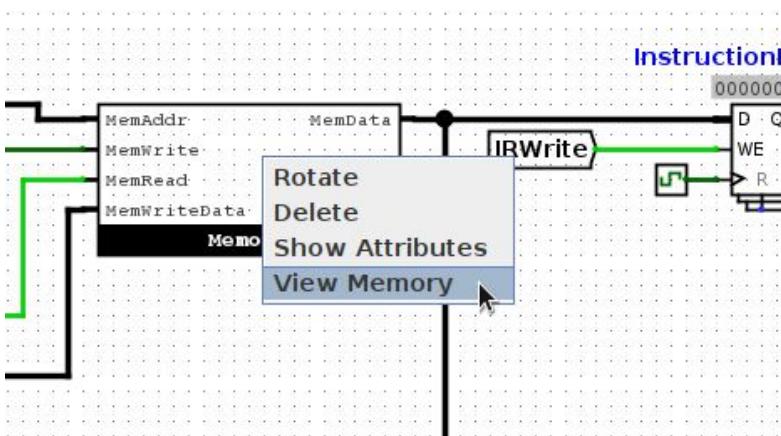
You can see the various components of the datapath using the menu on the left



# Load a program (1)

Go in the “main”, right click on “Memory” -> “View Memory”

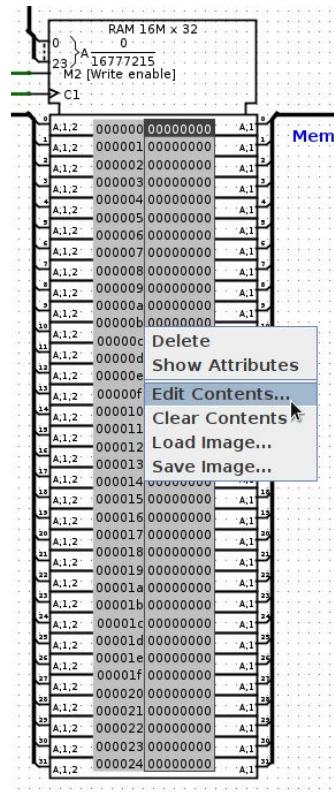
Right click on RAM -> “Load image..” and select the .hex file to store in the memory



## Load a program (2)

You can also manually insert instructions and data in hexadecimal.

**WARNING:** the addresses in the memory are word aligned, you cannot address by bytes



A screenshot of a memory dump window titled 'Logisim-Evolution: mem'. The window displays a grid of memory addresses from 0x000000 to 0x000024. Red arrows point to the first two bytes at addresses 0x000000 and 0x000001, which are both set to the value '0xC0'. The rest of the memory is filled with zeros.

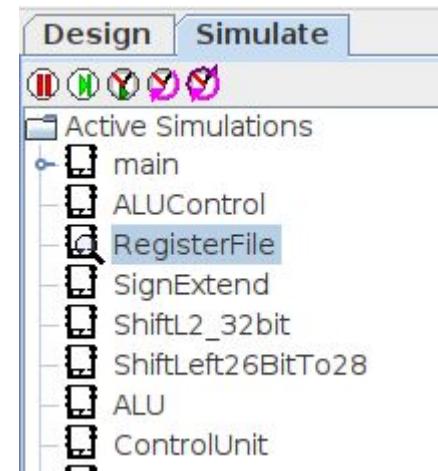
Address	Value
0x000000	0xC0
0x000001	0xC0
0x000002	0x00
0x000003	0x00
0x000004	0x00
0x000005	0x00
0x000006	0x00
0x000007	0x00
0x000008	0x00
0x000009	0x00
0x00000A	0x00
0x00000B	0x00
0x00000C	0x00
0x00000D	0x00
0x00000E	0x00
0x00000F	0x00
0x000010	0x00
0x000011	0x00
0x000012	0x00
0x000013	0x00
0x000014	0x00
0x000015	0x00
0x000016	0x00
0x000017	0x00
0x000018	0x00
0x000019	0x00
0x00001A	0x00
0x00001B	0x00
0x00001C	0x00
0x00001D	0x00
0x00001E	0x00
0x00001F	0x00
0x000020	0x00
0x000021	0x00
0x000022	0x00
0x000023	0x00
0x000024	0x00

# Execute a program

Verify that “Run simulation” is ticked.

Click on “Tick Full Cycle” (or press F2) to do one clock cycle.

You can verify the datapath steps by clicking on the components in the “simulate” menu



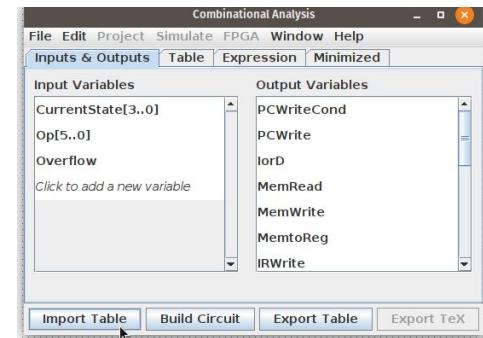
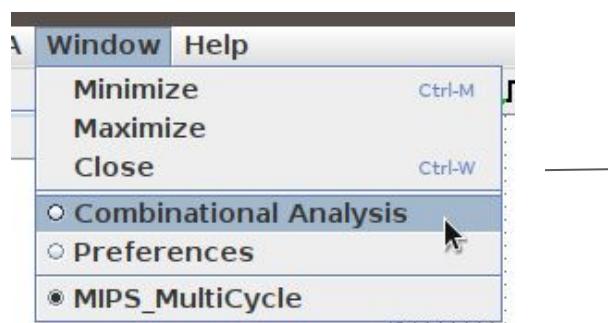
# Tips & Tricks for modifying the datapath

# Control Unit (1)

The control unit is generated by a Logisim Evolution tool, that allows us to create a circuit based on its truth table

You can import truth tables from the .txt files in the repo (the same is true for the ALU Control)

Being the control unit a FSM we also need, in addition to inputs and outputs, a state of the machine. We achieve this by adding an input called "CurrentState" and an output called "NextState".



## Control Unit (2)

In the “Table” tab you can see the truth table of the Control Unit.

The bit indicated by “-” are “don’t cares”.

CurrentState[3..0]	Op[5..0]	Overflow
0 0 0 0	- - - - -	-

## Control Unit (3)

We know that the first 2 steps are the same for all the instructions.

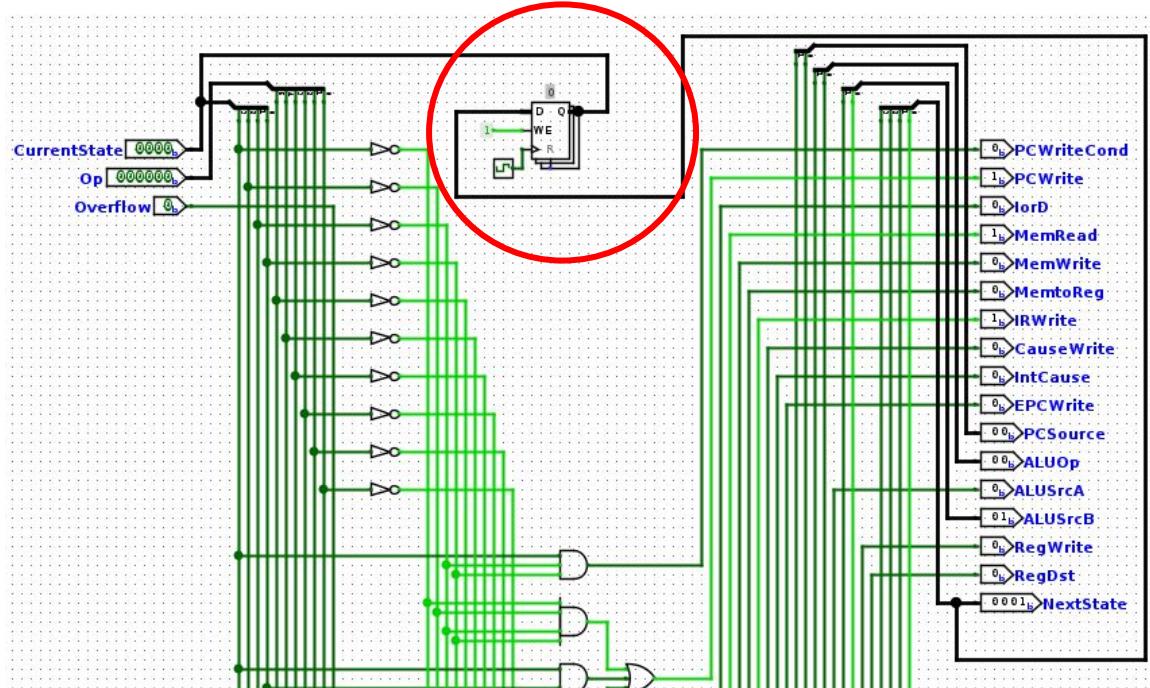
When we are in the second state (0001) we go to the next state based on the opcode.

# Control Unit (4)

After modifying the truth table we can click on “build circuit” and generate the ControlUnit.

The generated circuit is not a sequential circuit. We need to add a 4 bit register to store the state of the FSM. It will take NextState as input and CurrentState as output.

Delete then the “tunnels” CurrentState and NextState.



## Modifying the width of a signal (1)

You might need to modify the numbers of bit of a signal.

Go in the “Inputs & Outputs” tab,  
double click on the signal you want to  
modify and select the number of bits.

If you are adding bits remember to insert “0” where there are “-” in the new bits.



ALUSrcA[1..0]
-0
-0
-0
-0
-0
-0
-0
-0
-0
-0
-0
-0
-0
-0
-1
-1
-1
1
-1
-1
-1
0
-0
-0
-1
0
0
-1
-0
-0

## Modifying the width of a signal (2)

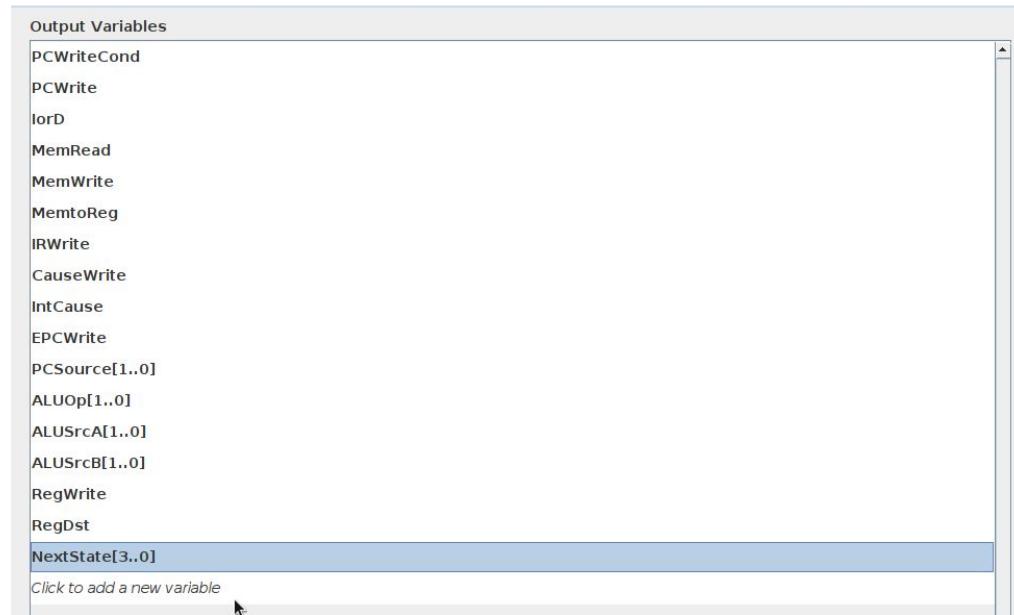
When you modify the width of the signals you also need to modify the components that use the signals.

Selection: Multiplexer	
VHDL	Verilog
Facing	East
Gate Size	Wide
Select Location	Top/Right
Select Bits	2
Data Bits	32
Disabled Output	Zero
Include Enable?	No

Selection: Tunnel "ALUSrcA"	
VHDL	Verilog
Facing	South
Data Bits	2
Label	ALUSrcA
Label Font	SansSerif Bold 16

# Add control signals (1)

To add a new control signals go in “Inputs & Outputs”, click “Add a new variable” , write the name of the signal and select the number of bits.



## Add control signals (2)

After building the new ControlUnit you can find the signal in the main.

Add a “Tunnel” to use the signal inside the datapath.



You can find everything on my github:

<https://github.com/fdila/MIPS-multicycle-datapath>

Open a issue if you find something wrong, and Pull Requests are also welcome!