Logisim

HDL language and Logisim

- Most real-world hardware design is done using a text-based hardware description language – VHDL, Verilog, etc.
  - Schematics can be compiled into a text decription
  - Can use a simulator to test the circuit
  - Other back-end tools optimize, perform layout and wire routing, floorplan, etc.
  - Final spec is either downloaded onto a programmable device, or etched into silicon.
- We will use Logisim for all hardware design
  - Interactive, graphical schematic editor
  - Educational use, user-friendly

Verilog Example: Decoder

```
3  // File Name   : decoder_using_case.v
4  // Function    : decoder using case
5  // Coder       : Deepak Kumar Tala
6  //-----------------------------------------------------
7  module decoder_using_case (
8  binary_in    , //  4 bit binary input
9  decoder_out , //  16-bit  out
10 enable          //  Enable for the decoder
11 );
12 input [3:0] binary_in  ;
13 input  enable ;
14 output [15:0] decoder_out ;
15
16 reg [15:0] decoder_out ;
17
18 always @ (enable or binary_in)
19 begin
20   decoder_out = 0;
21   if (enable) begin
22     case (binary_in)
23       4'h0 : decoder_out = 16'h0001;
24       4'h1 : decoder_out = 16'h0002;
25       4'h2 : decoder_out = 16'h0004;
26       4'h3 : decoder_out = 16'h0008;
27       4'h4 : decoder_out = 16'h0010;
28       4'h5 : decoder_out = 16'h0020;
29       4'h6 : decoder_out = 16'h0040;
30       4'h7 : decoder_out = 16'h0080;
31       4'h8 : decoder_out = 16'h0100;
32       4'h9 : decoder_out = 16'h0200;
33       4'hA : decoder_out = 16'h0400;
34       4'hB : decoder_out = 16'h0800;
35       4'hC : decoder_out = 16'h1000;
36       4'hD : decoder_out = 16'h2000;
37       4'hE : decoder_out = 16'h4000;
38       4'hF : decoder_out = 16'h8000;
39     endcase
40   end
```

Other CAD tools in circuit design

- Circuit level tool vendors
  - Cadence, Synopsys, etc.
  - other smaller players
- Board level tool vendors
  - Altium, Eagle and many more

HDL language and Logisim

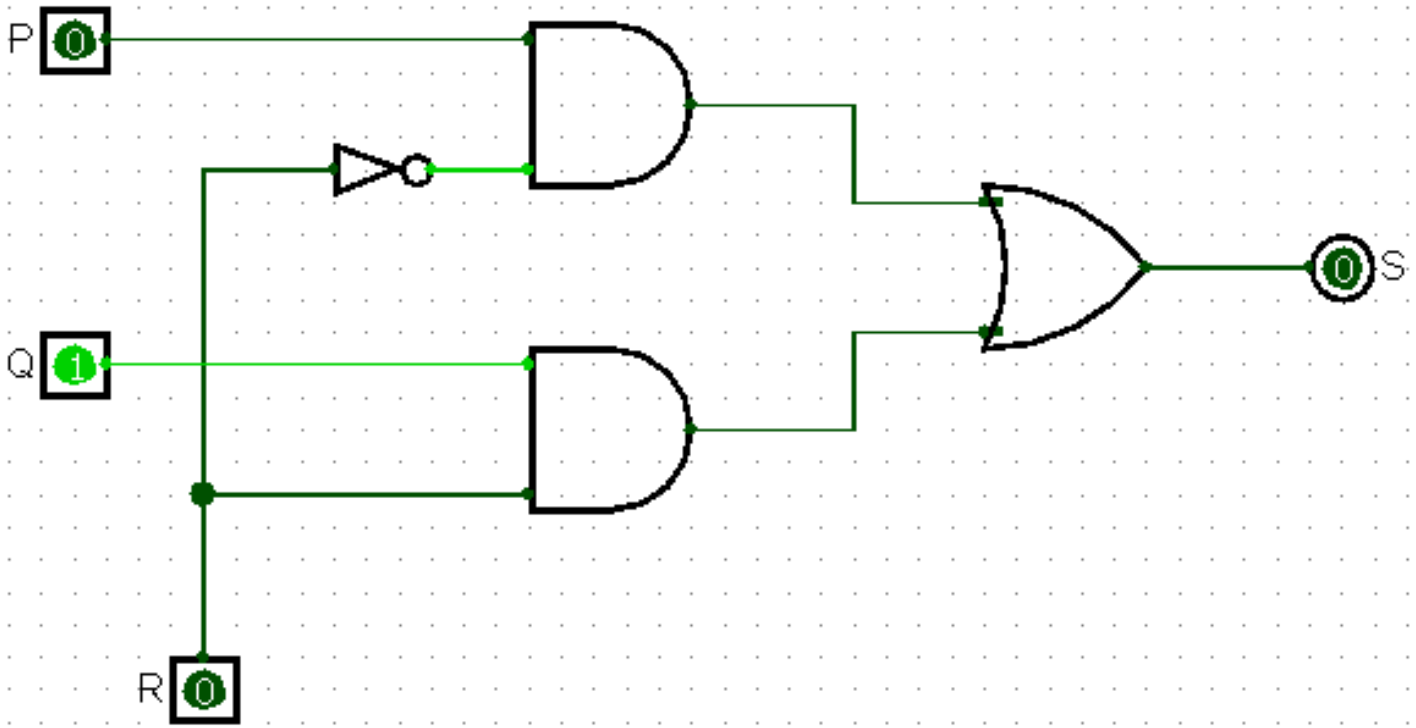
- Most real-world hardware design is done using a text-based hardware description language – VHDL, Verilog, etc.
  - Schematics can be compiled into a text decription
  - Can use a simulator to test the circuit
  - Other back-end tools optimize, perform layout and wire routing, floorplan, etc.
  - Final spec is either downloaded onto a programmable device, or etched into silicon.
- We will use Logisim for all hardware design
  - Interactive, graphical schematic editor
  - Educational use, user-friendly

To be covered…

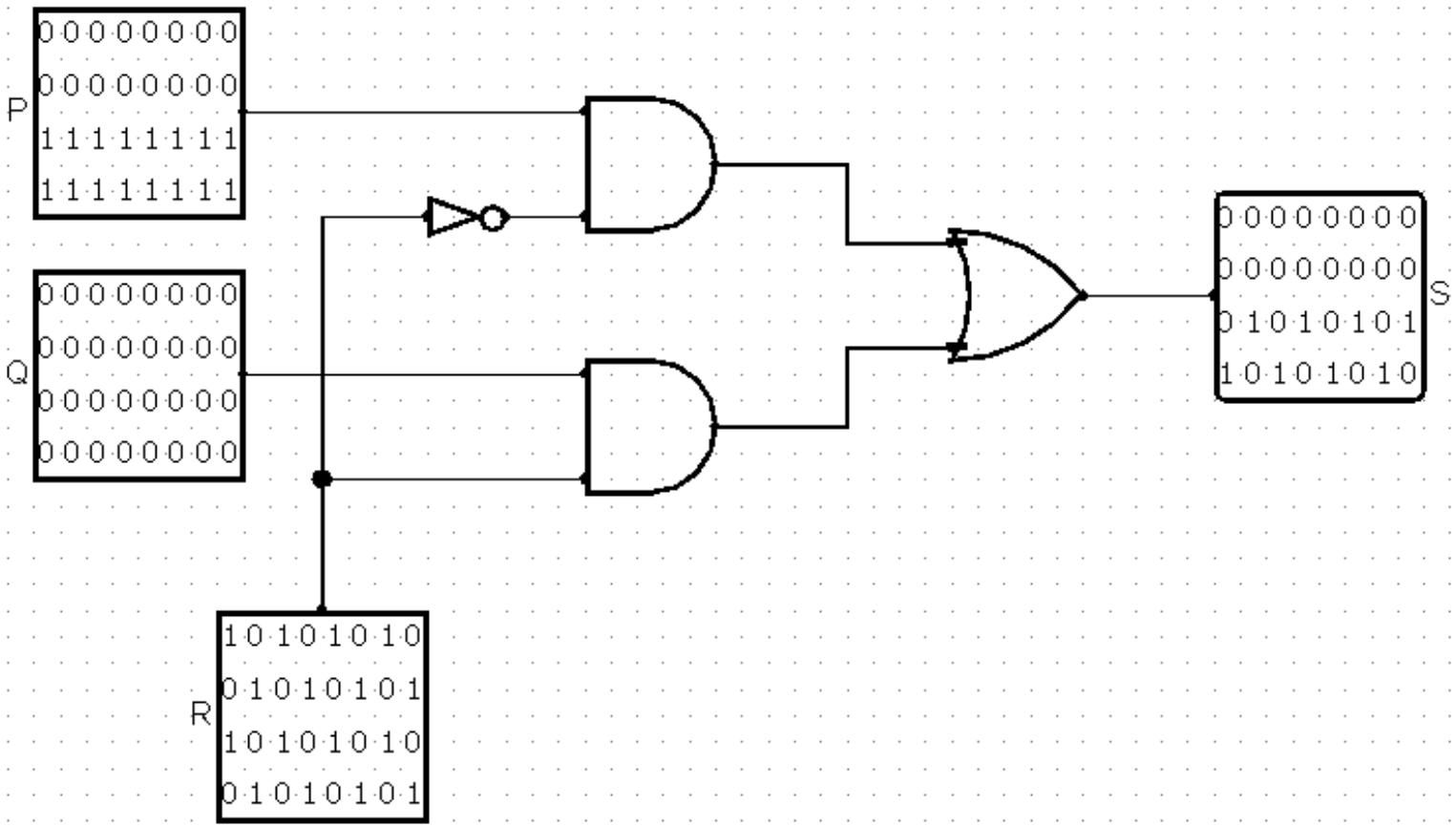
- Pins and subcircuits
- Probes for debugging
- Bundles/splitters
- Logging
- Test vectors
- S-R latch, D latch, D flip-flop
- Examples

Example Circuit: 1 bit 2:1 Mux

- S = P if R == 0
- S = Q if R == 1

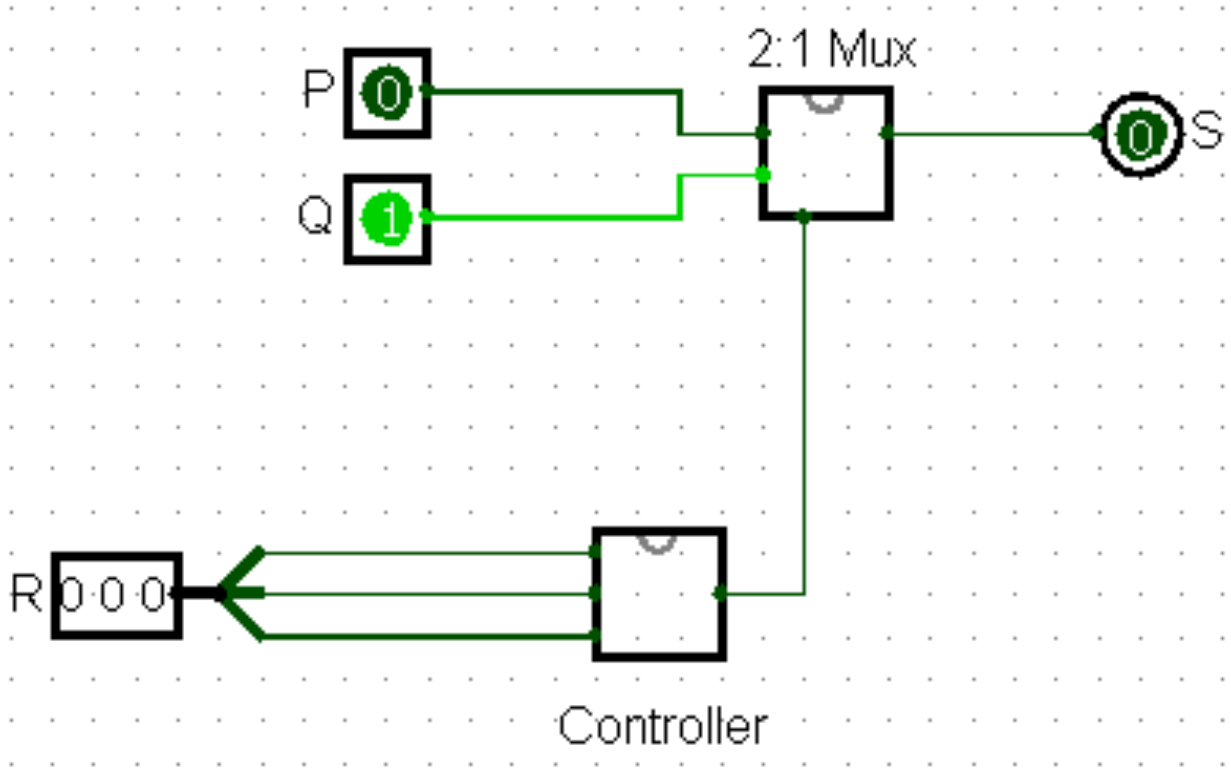


Example Circuit: 32-bit 2:1 Mux



Subcircuits: 2:1 Mux and Controller

- S = Q if R == 010
- S = P otherwise



Logging and Test Vectors

Log File

```
# Logisim: Log main of multiplexer1
# Mon Jan 25 11:18:42 EST 2010
P         Q         R         S
0         0         0         0
0         0         1         0
0         1         0         0
0         1         1         1
1         0         0         1
1         0         1         0
1         1         0         1
1         1         1         1
```

Test Vector Truth Table

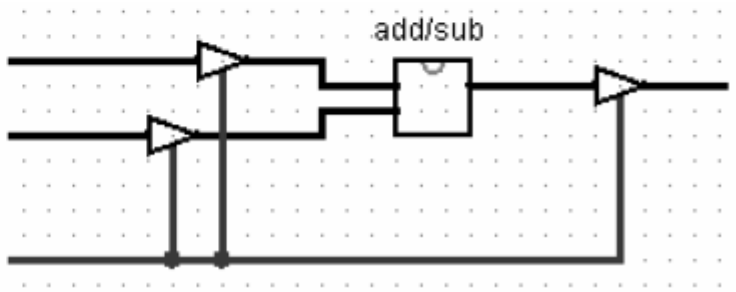
Passed: 8 Failed: 0

status	P	Q	R	S
pass	0	0	0	0
pass	0	0	1	0
pass	0	1	0	0
pass	0	1	1	1
pass	1	0	0	1
pass	1	0	1	0
pass	1	1	0	1
pass	1	1	1	1

Logisim Don’ts

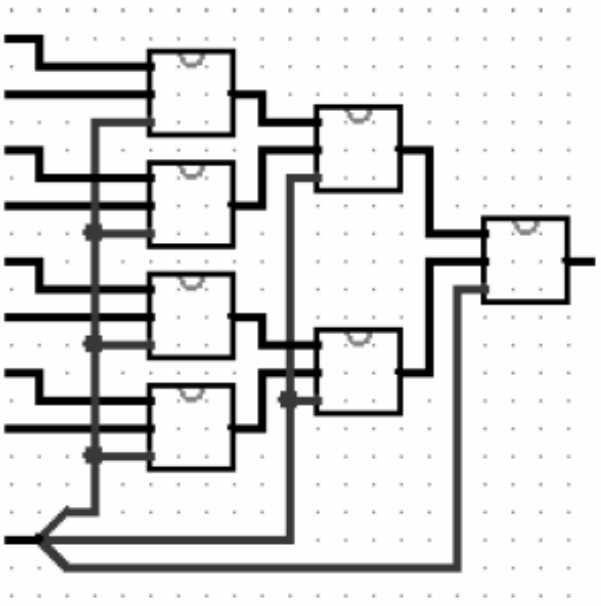
- Leave wires floating
  - Works in logisim
  - Breaks in real life

- Use a multiplexor instead of controlled buffer



Logisim Don’ts (cont.)

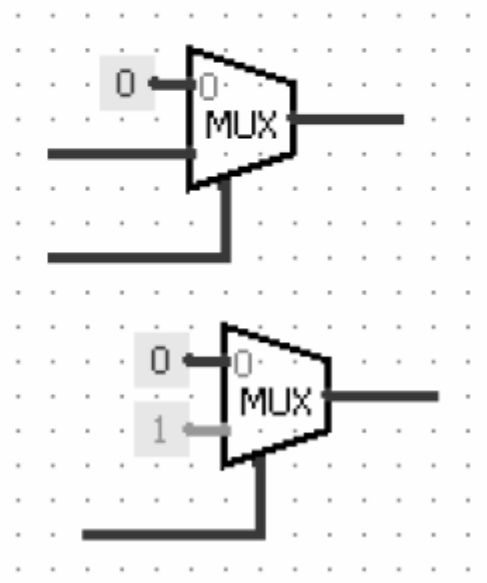
- Don’t reinvent the wheel
  - Waste time
  - Confusing to grade
- Almost every component is customizable
  - Number of inputs
  - Input bit width



Building an 8-way Mux the hard way

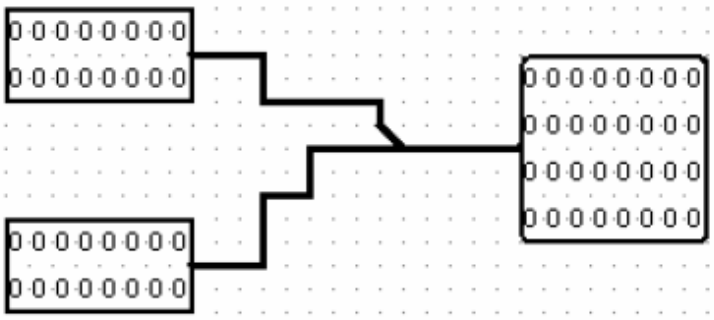
Logisim Don’ts (cont.)

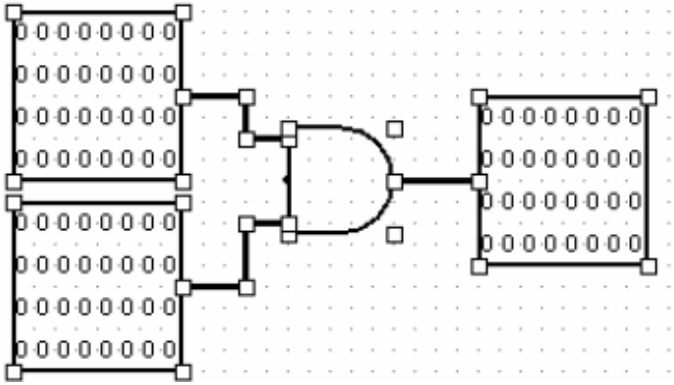
- Avoid Constant input
  - Constants are almost never necessary
  - Exception is supplying value to extra input
- Try to optimize away before using
  - Truth table



Logisim Don’ts (cont.)

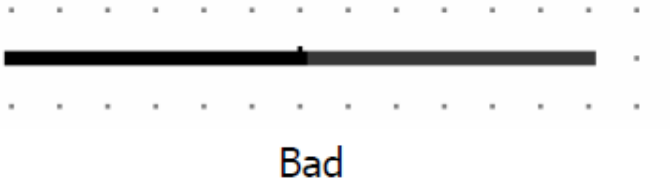
- Don’t make trivial subcircuits
  - Sub-circuit needs to perform some meaningful logic functions
  - You will never have a C function just to add two numbers up, do you?
- Problems
  - Wasting time specifying inputs and outputs of small circuits
  - Big hierarchy hard to understand





Logisim Don’ts (cont.)

- Don’t use invisible splitters
  - All you really need is just a wire
  - It is really hard to see them when we grade

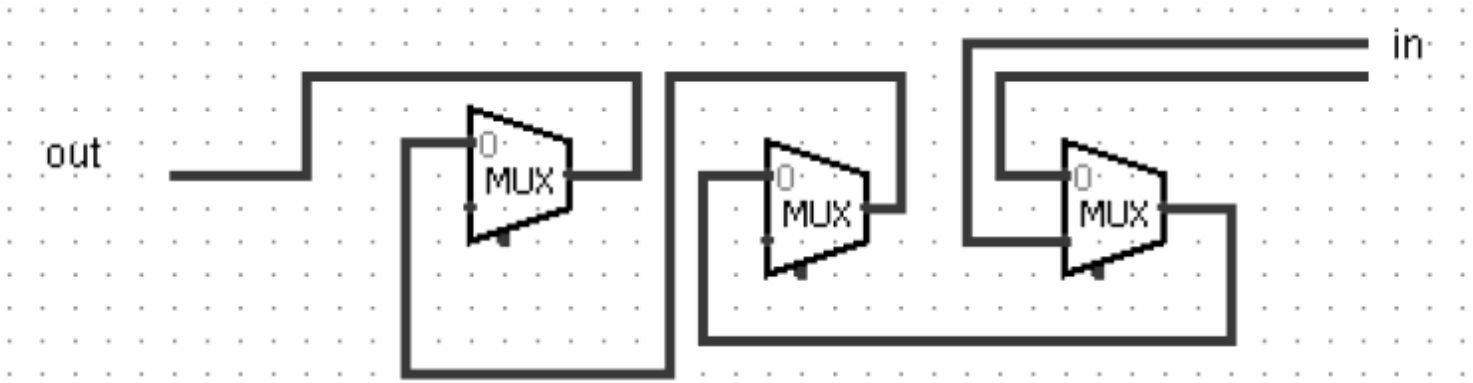


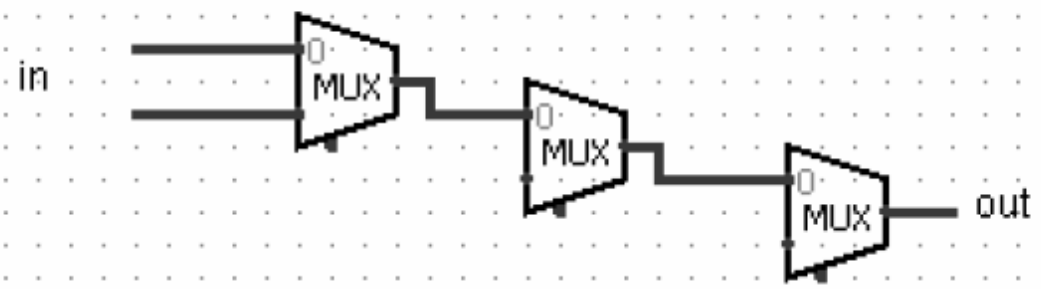
Bad

Good

Logisim Don’ts (cont.)

- Don’t work from Right to Left





Some more information

- MIPS assignment:
  - 32-bit ALU
  - 32-bit pipelined processor
- Looking for help?
  - Course webpage http://www.cs.cornell.edu/courses/cs3410/2011sp/
  - Newsgroup: cornell.class.cs3410
  - Staff email list: cs3410-staff-l@cs.cornell.edu