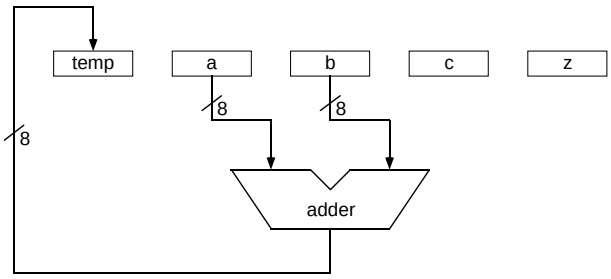
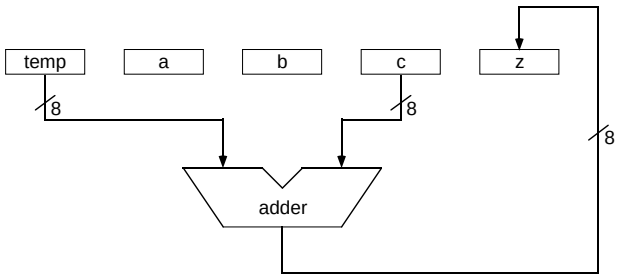
A Very Simple Computer Datapath (Review)

- temp = a + b (a to left input…)

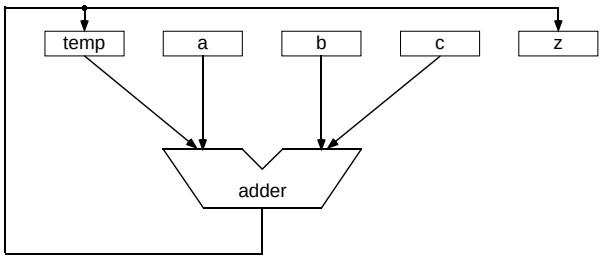


- z = temp + c (temp to left input…)

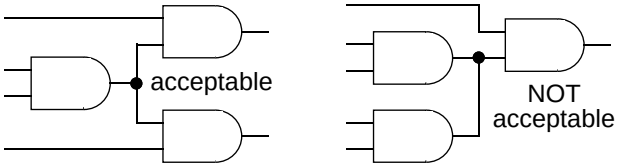


Connecting Inputs to Outputs

- What we'd like to do is something like the following: (warning — this doesn't work!)



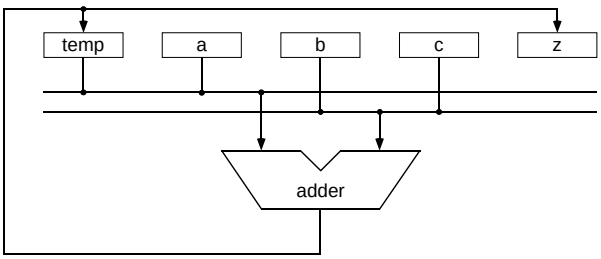
- Unfortunately, although one output can connect to multiple inputs



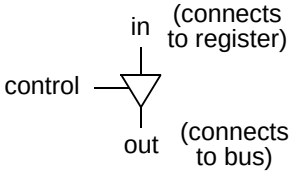
- Multiple outputs can ***not*** connect to one input!

Bus-Based Datapath

- A *bus* is a signal with multiple inputs (sources) and multiple outputs (sinks)



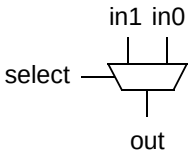
- We can use a new circuit element, called a *tri-state device*, to ensure that only one register drives each bus at a time



control	in	out
0	0	–
0	1	–
1	0	0
1	1	1

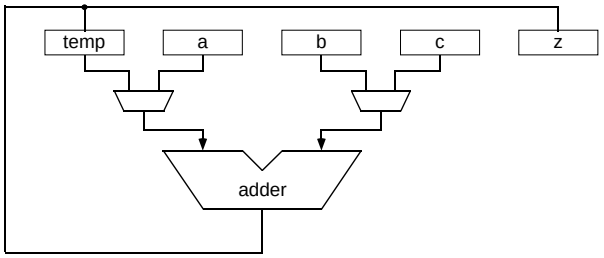
Multiplexer-Based Datapath

- A *2-input* multiplexer is a circuit element that can select between 2 inputs



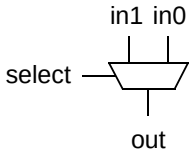
select	out
0	in0
1	in1

- Multiplexers can be used at each adder input to select the appropriate value



Building a Multiplexer

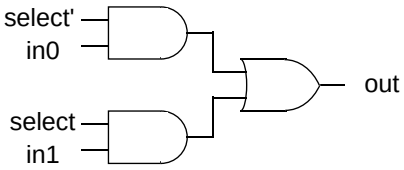
- A *2-input* multiplexer is a circuit element that can select between 2 inputs



select	out
0	in0
1	in1

- Multiplexers can be built from simple gates…

select	in1	in0	out
0	–	0	0
0	–	1	1
1	0	–	0
1	1	–	1

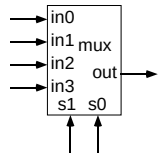


- The inverse of a multiplexer — the *demultiplexer* — can also be useful at times

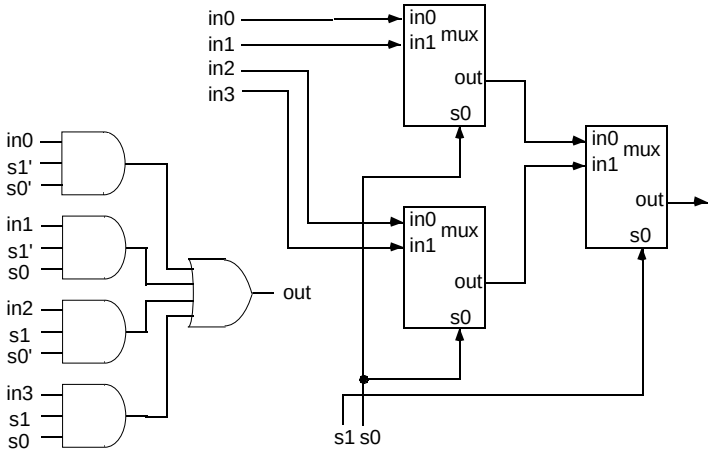
Building a Bigger Multiplexer

- A *4-input multiplexer* can select one of four inputs as shown below.

s1	s0	out
0	0	in0
0	1	in1
1	0	in2
1	1	in3

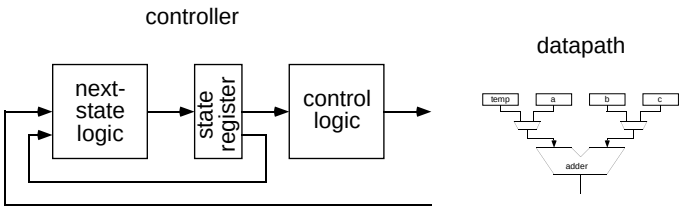


- The 4-input multiplexer can be built from simple gates, or from 2-input multiplexers



Controller

- A complicated digital circuit (such as a processor) typically consists of two main parts:
  - A *datapath* to perform the necessary functionality, consisting of registers, ALUs, multiplexers, buses, etc.
  - A *controller* to sequence the design and control the various registers and ALUs in the datapath



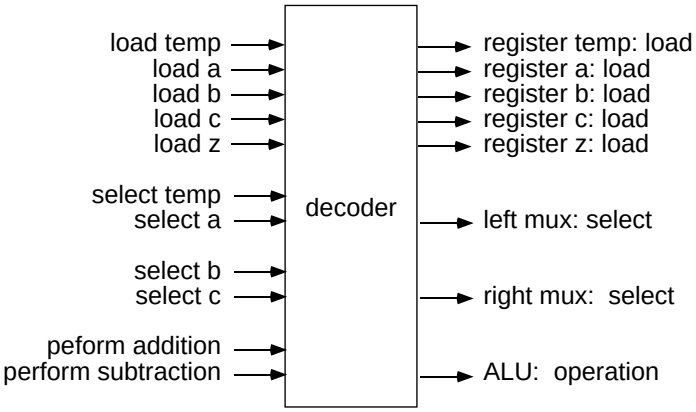
- The control logic has to *decode* the bits in the state register to decide what signals to activate to control the datapath

Decoding the State Register

- The state register might have bits to specify that:
  - Register *temp* should be loaded
    - Register *a* should be loaded
    - Same for other registers…
  - The multiplexer at the adder's left input should select the value from register *temp*
    - The multiplexer at the adder's left input should select the value from register *a*
    - Same for other multiplexer…
  - If the adder were a more general ALU, that it should perform an addition operation when executing our two example instructions

Decoder

- A *decoder* is a device to convert a set of inputs to a set of outputs according to a specific set of rules



- This decoder could be implemented with a 2-level *and-or* (or *nand*) circuit for each output

Homework #2 — Due 9/28/98 (Part 2)

3. The register given on slide 10 of Lecture 08 is simpler than the register given on page 58 of the text. Explain how the two differ.

4. Show how an 8-input multiplexer (with inputs labeled i7 (msb) to i0 (lsb) can be constructed from 4-input multiplexers. Assume the lsb of the select line is labeled s0.