# Lecture 29, Nov 22, 2022

## Input/Output

- Memory mapped I/O – we can manipulate I/O devices though loads and stores to specific memory addresses
  - Devices sit at the memory locations for certain addresses and respond to those addresses
  - Consequently real memory ignores those addresses
  - These are known as *address spaces*
- The address bus goes into an address decoder, which outputs enable signals to different memory or I/O devices depending on which device's memory map the address is in
  - This enable signal controls which device the data on the bus is written to
  - The address decoder also controls a mux of all the device outputs to select which device's output goes on the data input bus to the CPU
- Example: I/O device 1 is on memory address `0x20001000`; write the value 7 to this device and read its output
  - When we read data from the device, this may or may not be the same data we sent; it could also be e.g. an ack or some processed form of data

```
li s1, 0x20001000 # Load the device address
addi s0, zero, 7
sw s0, 0(s1)      # Write the value to the device
lw s0, 0(s1)      # Read back a value from the device
```

- Often we might need a delay loop to intentionally slow down the CPU to match the speed of the I/O device
  - e.g. using a delay when updating a counter connected to a hex display to make the numbers readable