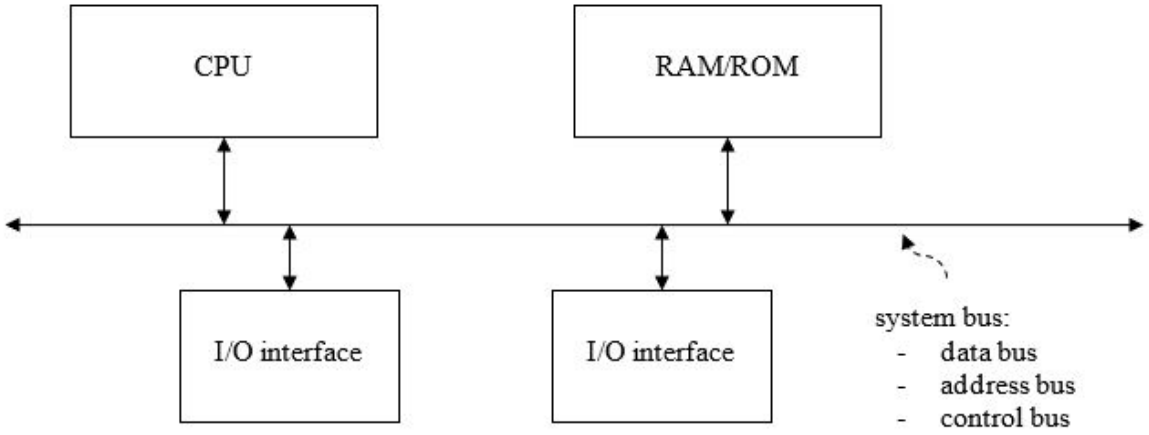
# I/O Hardware & Bus Basics

# Bus Structuring

- Note: in the following discussion, a single “system bus” is assumed.  For performance reasons, many computers have a hierarchy of busses
- Recall the basic organization:



# Bus Structuring

- For the **data bus**:
  - Can there be more than one writer?
  - Can there be more than one reader?
  - Clearly, yes to both
    - Only one writer at a time

# Bus Structuring

- For the **address bus**?
- In a simple system (e.g. no DMA, only one CPU/core)
  - NO to both
    - The CPU drives the address bus and controls all bus transfers

# Bus Structuring

- For the **control bus lines**?
  - The answers depend on the particular line.
  - Bus transfers must follow a protocol
    - Must be controlled by a master.
  - Bus Master - component which drives address bus and bus control signals such as R/W
  - R/W – single writer – the bus master (typically the CPU), but multiple readers

# Bus Structuring

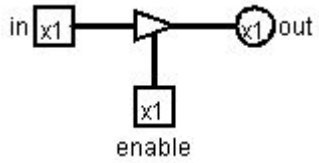
- **Control bus lines** continued …
  - IPL lines – multiple writers (non-CPU entities) and a single reader – the CPU
  - Again, in more complex systems, the CPU isn't always the bus master. DMA will be discussed later

# Three State Logic

- **Recall:** with multiple writers to a bus line, we must avoid connecting gate outputs together
- We have already seen one technique for allowing multiple (non-simultaneous) bus writers – what is it?
  - A multiplexer. (seen in the Simple Computer)
  - Typically not used for the external system bus.  A more flexible approach is to use three state logic.

# Three State Logic

- Idea: disconnect gates from the bus when they are not needed, using a kind of switch.




- Three state buffer:

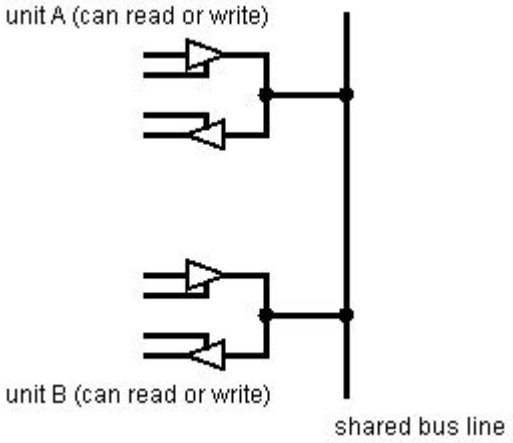
if enable = 1: out is connected to in (switch is closed)
out = in out can be either 0 or 1 (2 states)
if enable = 0: out is disconnected from in (switch is open)
out = ? (“high impedance” or floating) (1 state)

# Three State Logic

- Idea: each bus line writer (and reader) can be connected to the (data) bus lines through a three state buffer
  - Max. one writer should be connected at a time
  - If no writers are connected, the bus line is said to be “floating”
    - It is neither 0 nor 1, relative to ground
- Aside: this is why unused gate inputs should never be left unconnected. They should be wired to either 0 or 1
  - Remember IC would sometimes be ‘on’ even when no on signal present

# Three State Logic

- Bus control signals from the bus master drive the enable signals.
- Why might readers be disconnected when not reading?
    - This avoids reading an incorrect value due propagation delay



# Three State Logic

- Why might readers be disconnected when not reading?
- This avoids reading an incorrect value due propagation delay.
- Not all connections to the bus are three-state.
  - Three state buffers are involved for the data bus
  - This is not the case for external control signals (R/W)
  - Depends for the address bus on whether there is a single bus master or not
- For the system bus, why is three state logic more flexible than using a multiplexer?

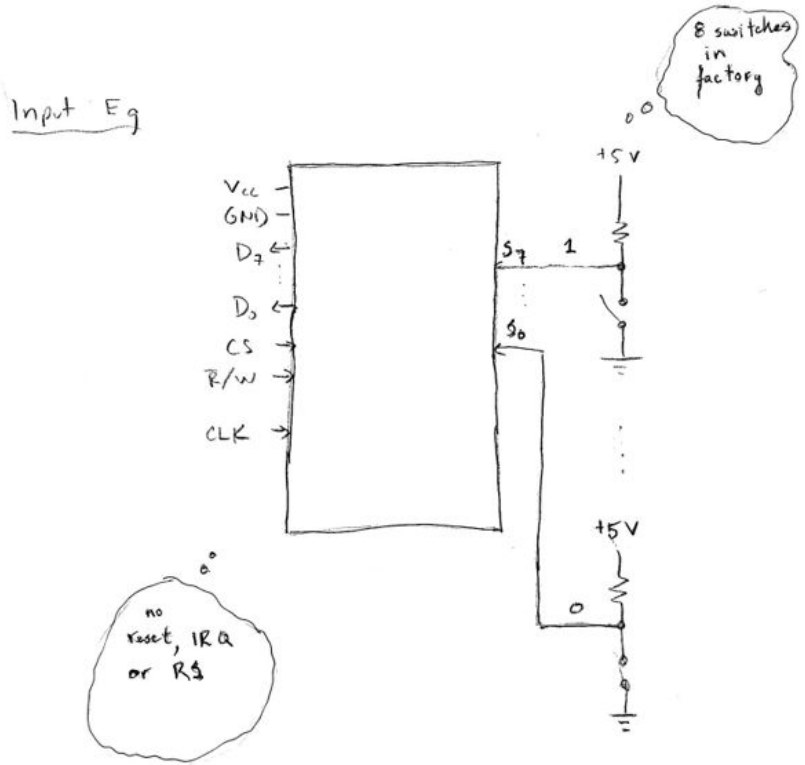
# Three State Logic

- For the system bus, why is three state logic more flexible than using a multiplexer?
  - A multiplexer would need to be specifically created for each motherboard
    - How would you handle adding interfaces (i.e. expansion boards)?
  - A multiplexer requires a select signal (lines) would need to either determine which CS line(s) to use for this OR would need to add select lines from CPU
  - As well, CS/RS signals are already generated for individual interfaces based on addresses

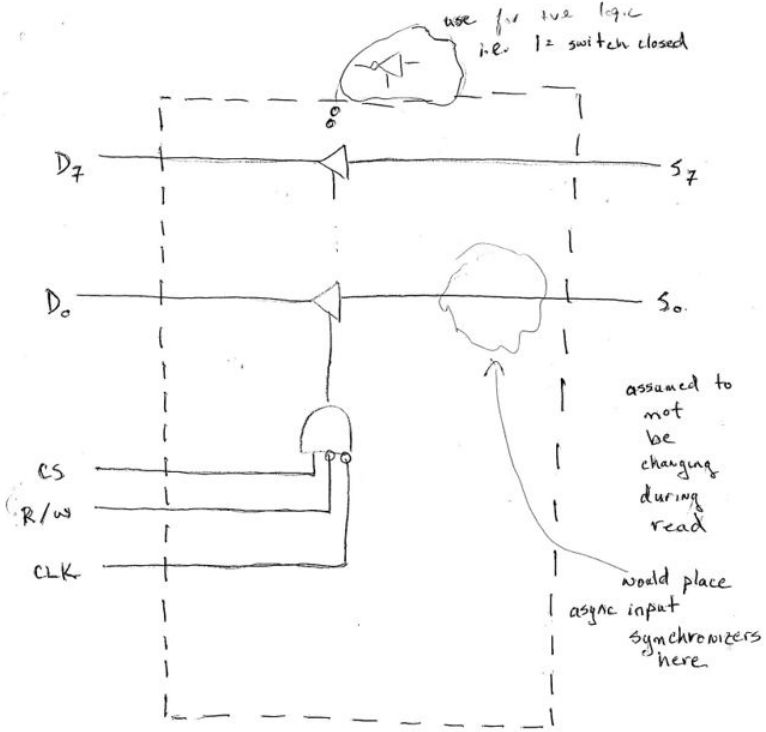
# Simple Input and Output Interfaces

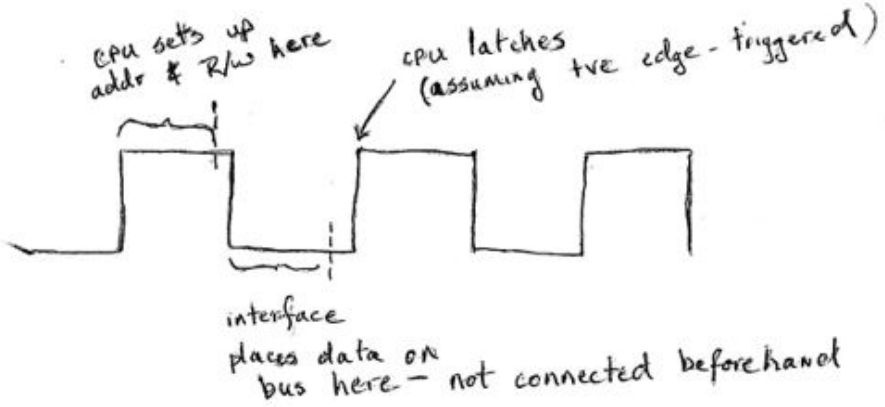
- Note:
  - Interfaces are parallel, but with no handshaking
  - Neither includes status or control registers; no RS signal
  - Neither supports interrupt-driven I/O
  - Both assume a synchronous bus
  - Generation of CS signal for interfaces is discussed

# Simple Input Example

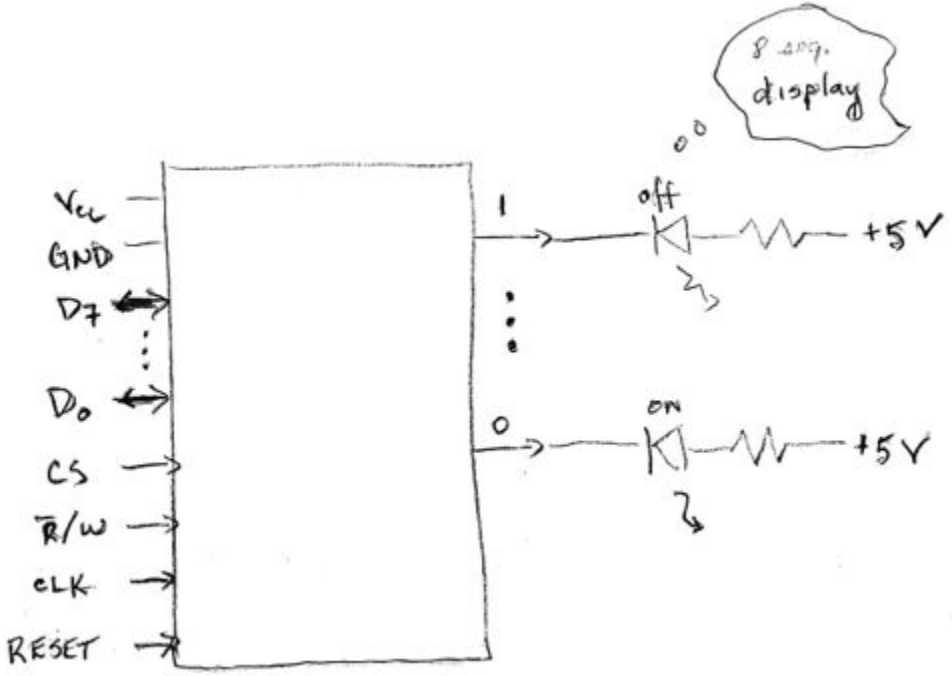


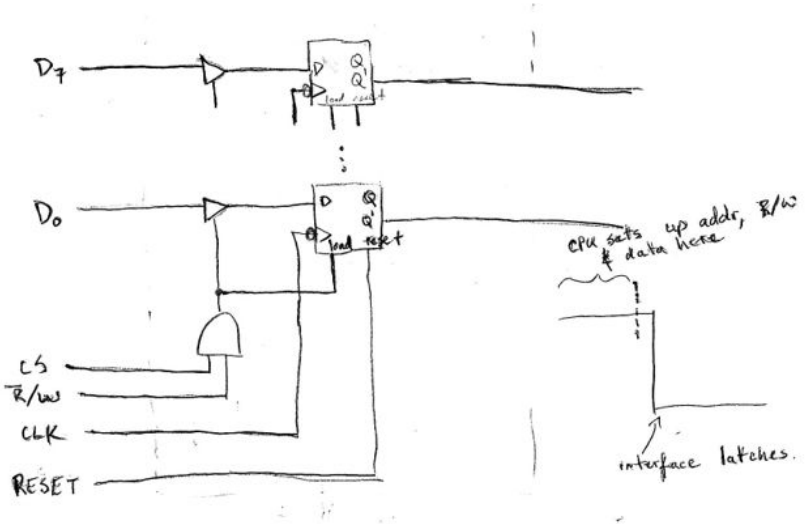
# Simple Input Example



# Simple Output Example



# General Purpose Parallel Interface



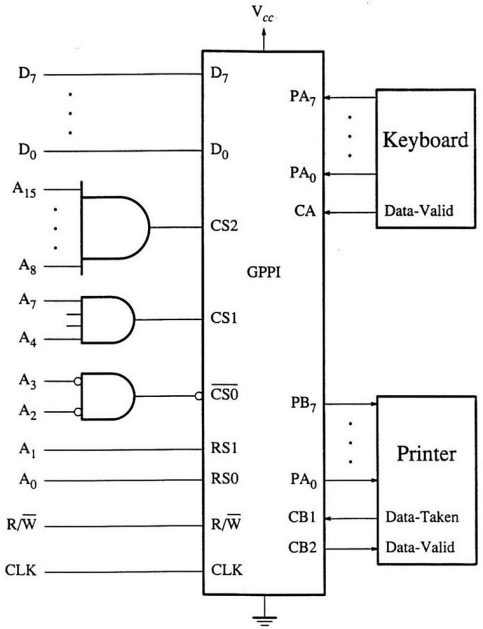
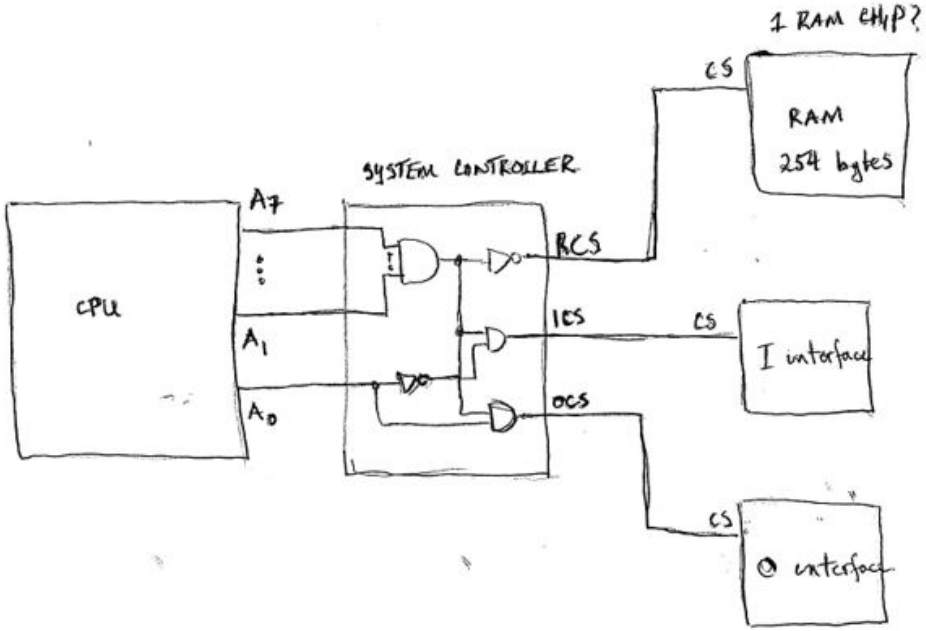



**Figure 6.27** A keyboard and a printer connected to a microprocessor bus via a general-purpose parallel interface chip.

# Choosing The Interface

- How are the addresses decoded?
  - Assuming and 8-bit address bus
  - Input interface at $FE
  - Output interface at $FF



# Input/Output Hardware

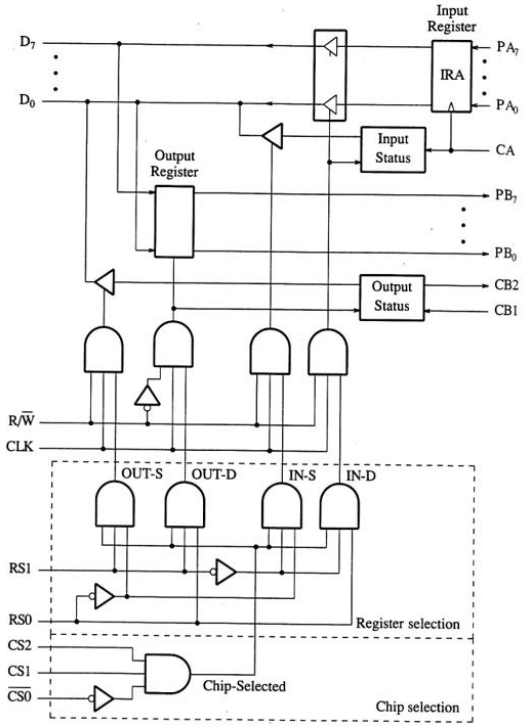
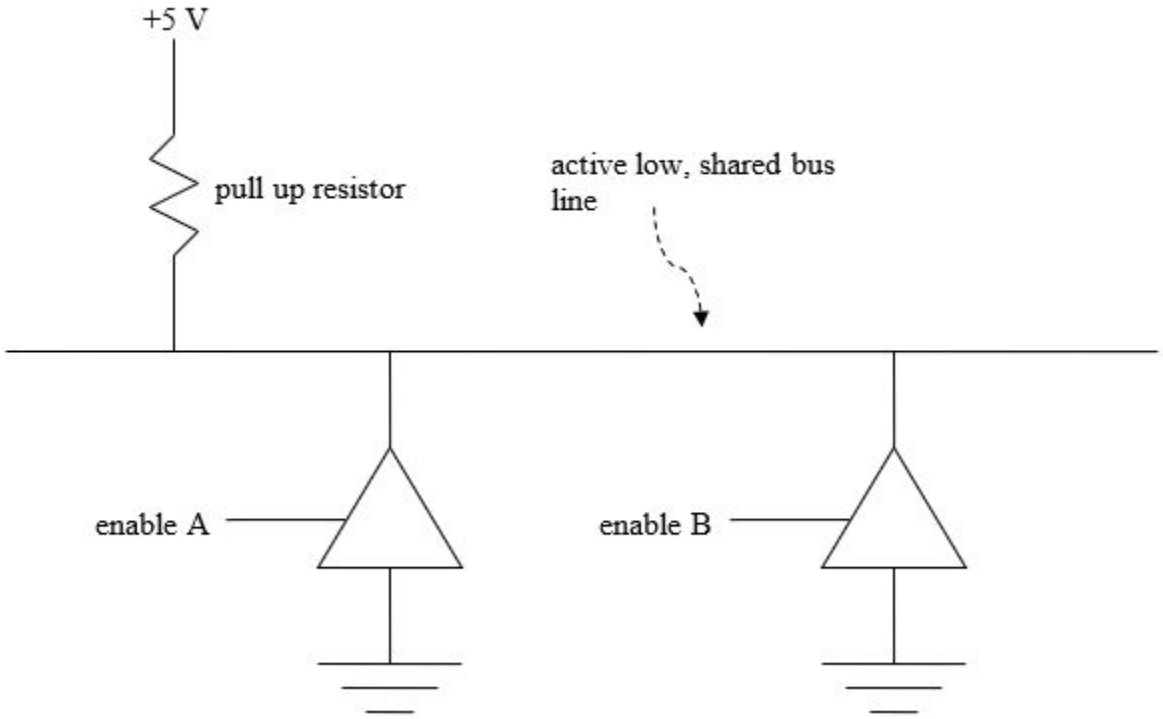



**Figure 6.25** Internal organization of a general-purpose input/output interface.

# Open Collector/Open Drain Laines

- Often, it is useful for ≥ 1 units to be able to assert a signal on a common bus line
- An open collector (or “open drain”) line is used – the precise term depends on the semiconductor logic family used
- The precise electrical details are beyond the scope of this course, but the basic idea is:
    - If “enable A” and “enable B” are both low, the bus line is high. Otherwise, the bus line is pulled low
    - The line is not three state – it never floats

# Open Collector/Open Drain Laines



# Open Collector/Open Drain Laines

- This is used to implement signals such as IRQ: