Bus-Based Computer Systems

- Busses.
- Memory devices.
- I/O devices:
  - serial links
  - timers and counters
  - keyboards
  - displays
  - analog I/O

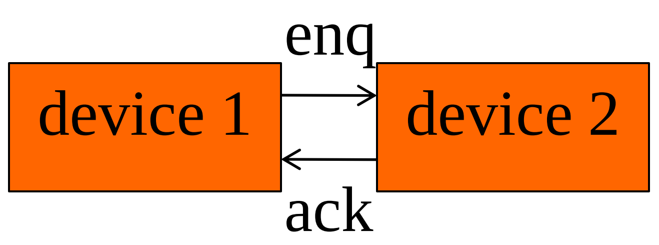
The CPU bus

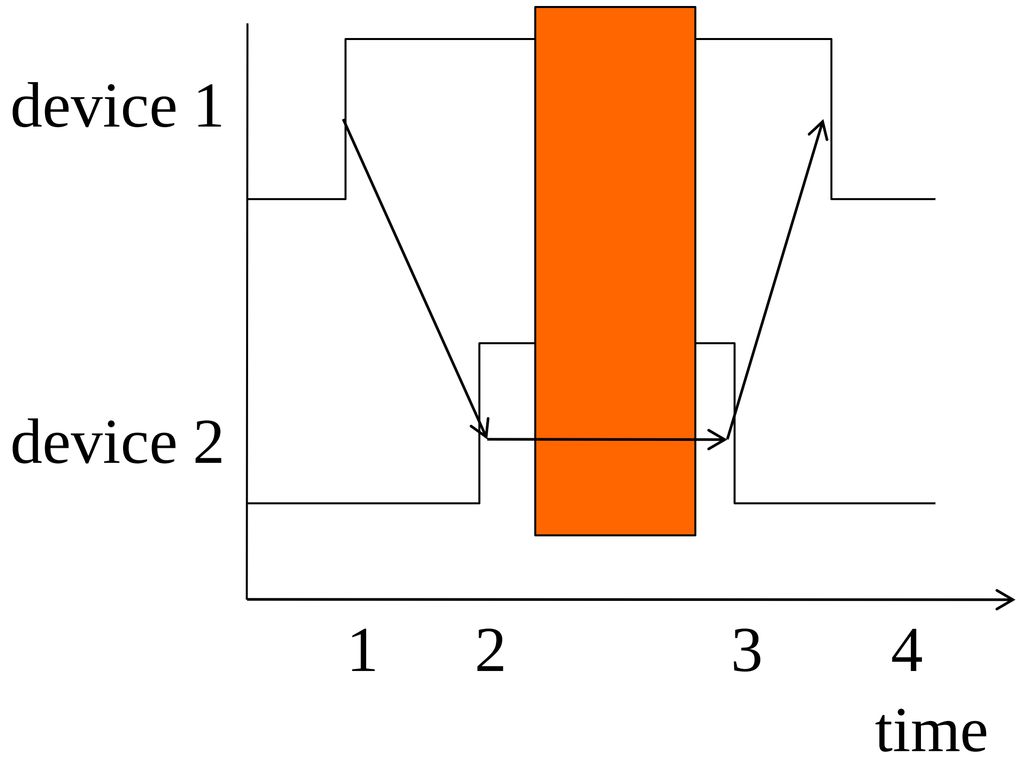
- Bus allows CPU, memory, devices to communicate.
    - Shared communication medium.
- A bus is:
    - A set of wires.
    - A communications protocol.

Bus protocols

- Bus protocol determines how devices communicate.
- Devices on the bus go through sequences of states.
  - Protocols are specified by state machines, one state machine per actor in the protocol.
- May contain asynchronous logic behavior.

Four-cycle handshake



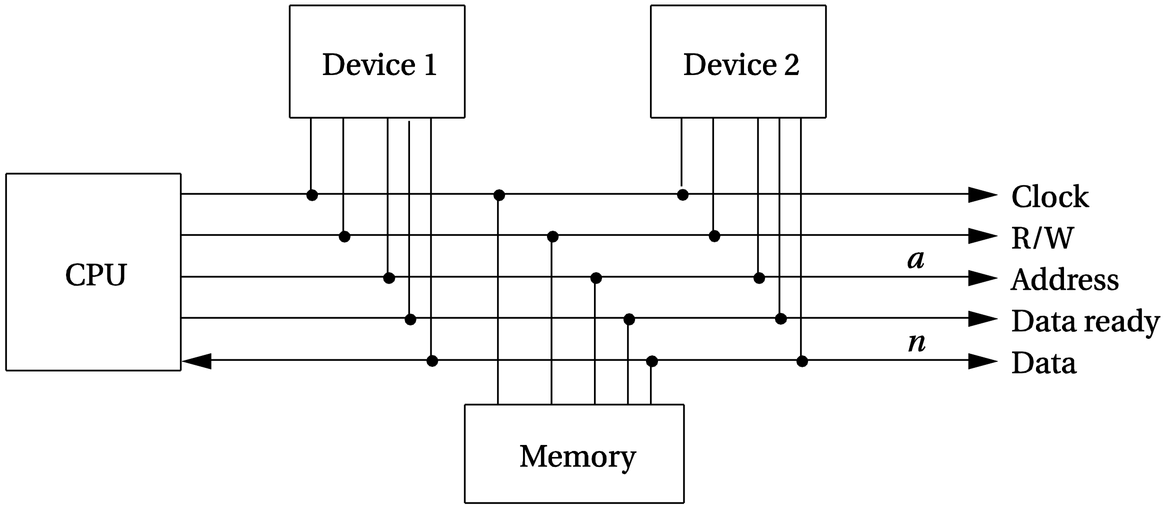


Four-cycle handshake, cont'd.

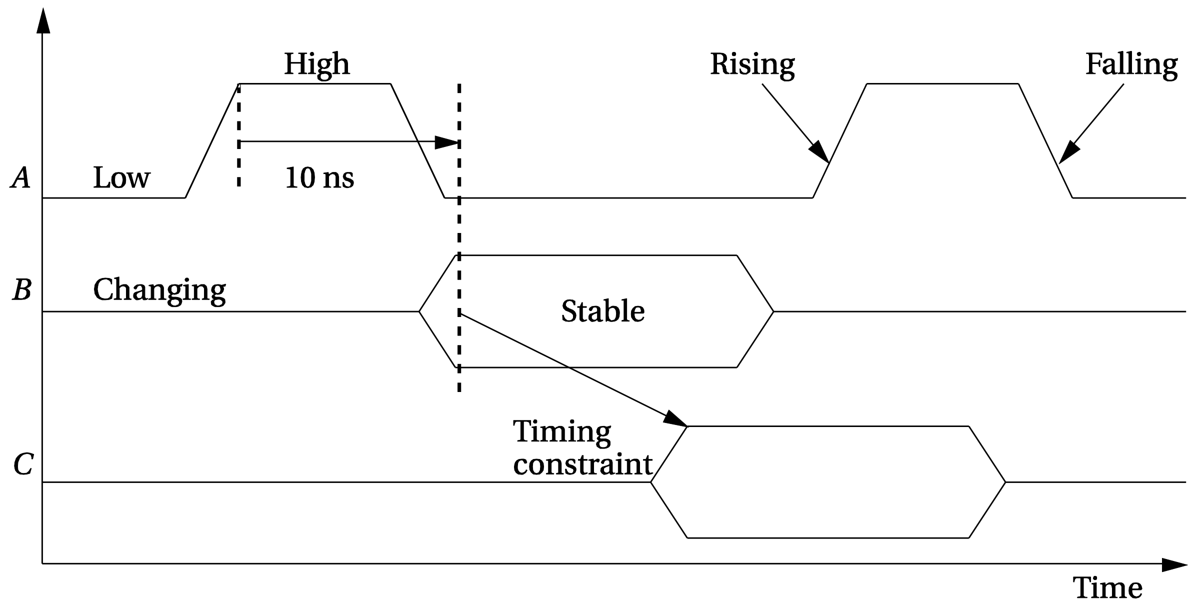
1. Device 1 raises enq.
2. Device 2 responds with ack.
3. Device 2 lowers ack once it has finished.
4. Device 1 lowers enq.

Microprocessor busses

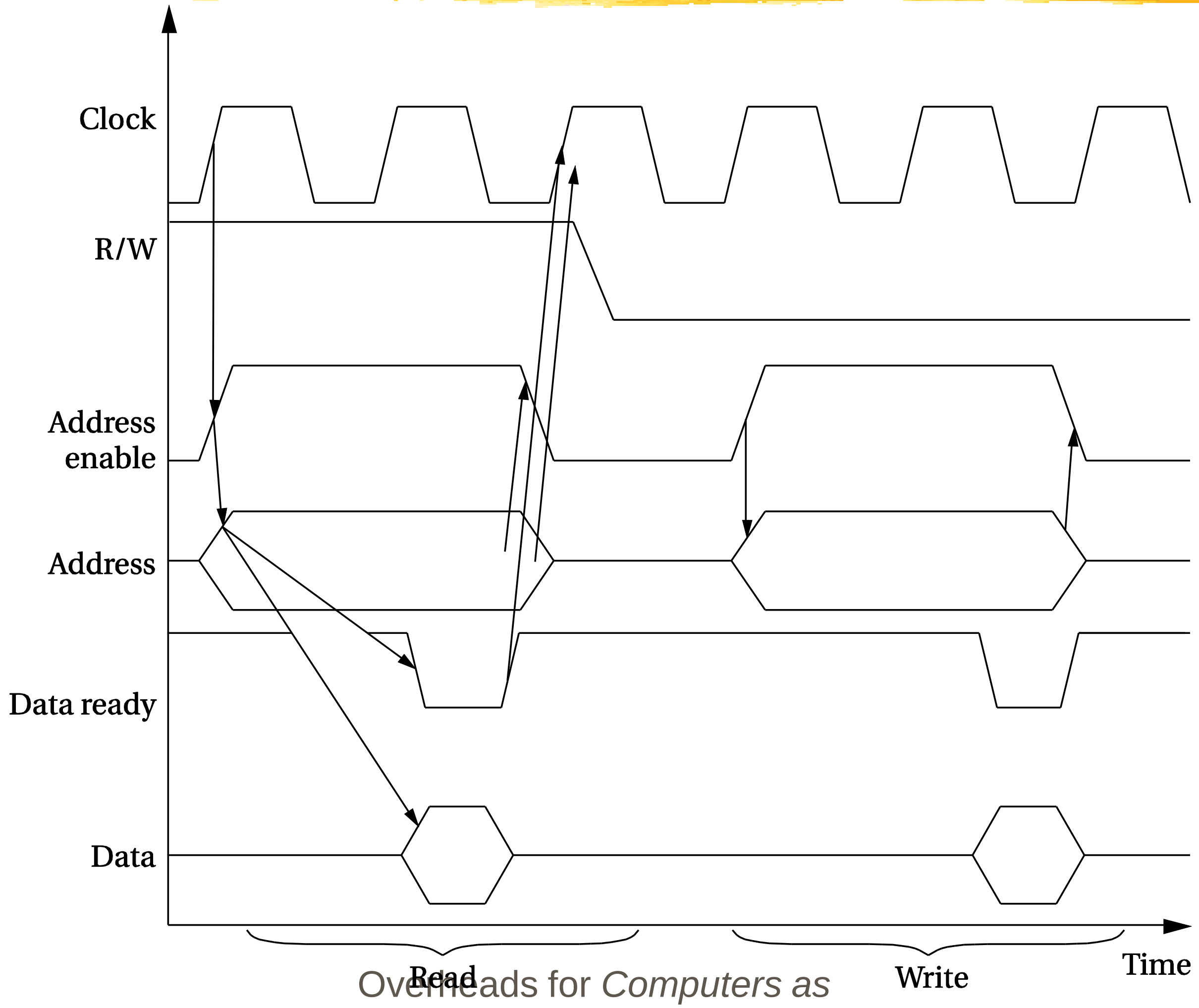
- Clock provides synchronization.
- R/W is true when reading (R/W' is false when reading).
- Address is a-bit bundle of address lines.
- Data is n-bit bundle of data lines.
- Data ready signals when n-bit data is ready.



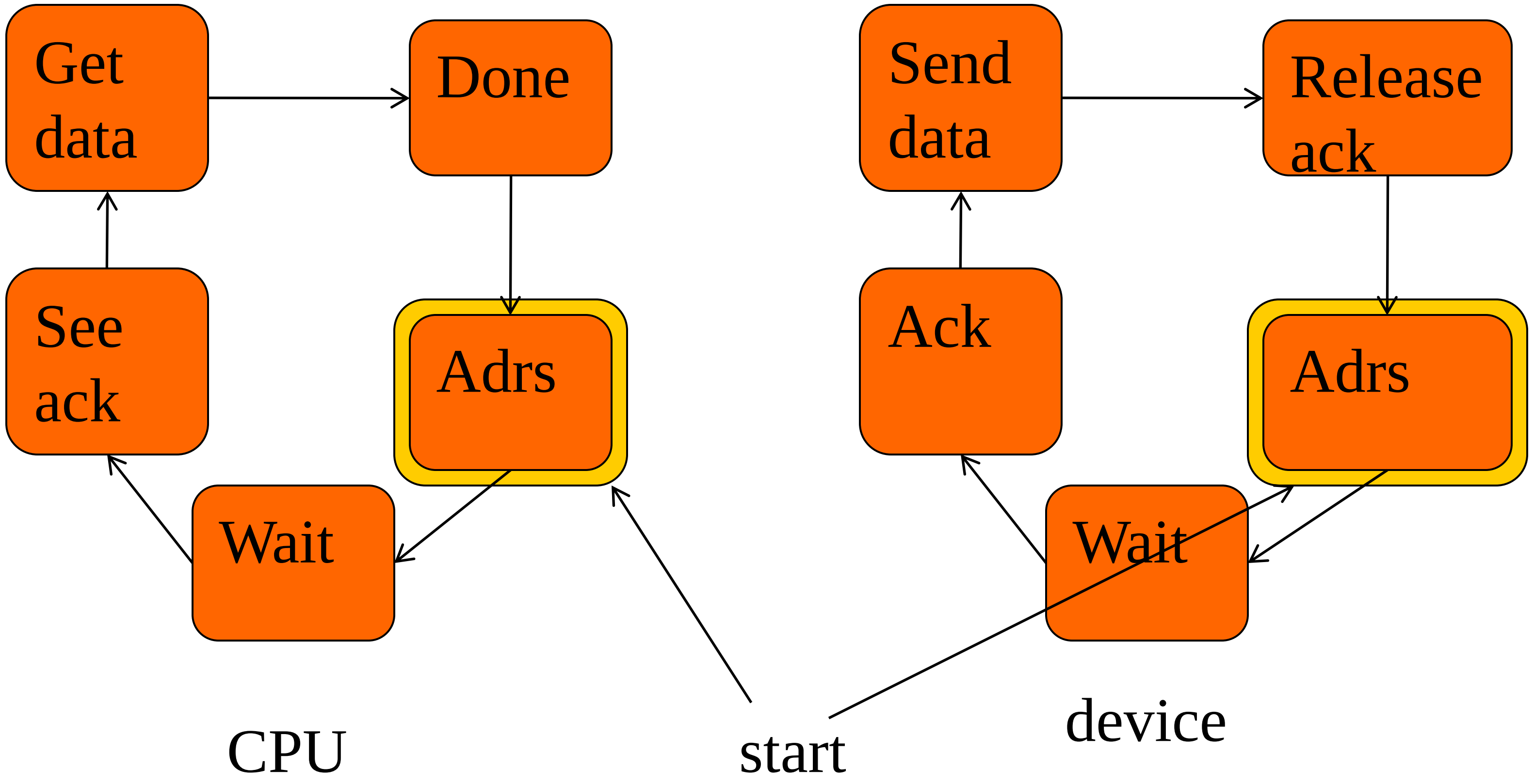
Timing diagrams



Bus read


Clock
R/W
Address enable
Address
Data ready
Data
Read
Write
Time

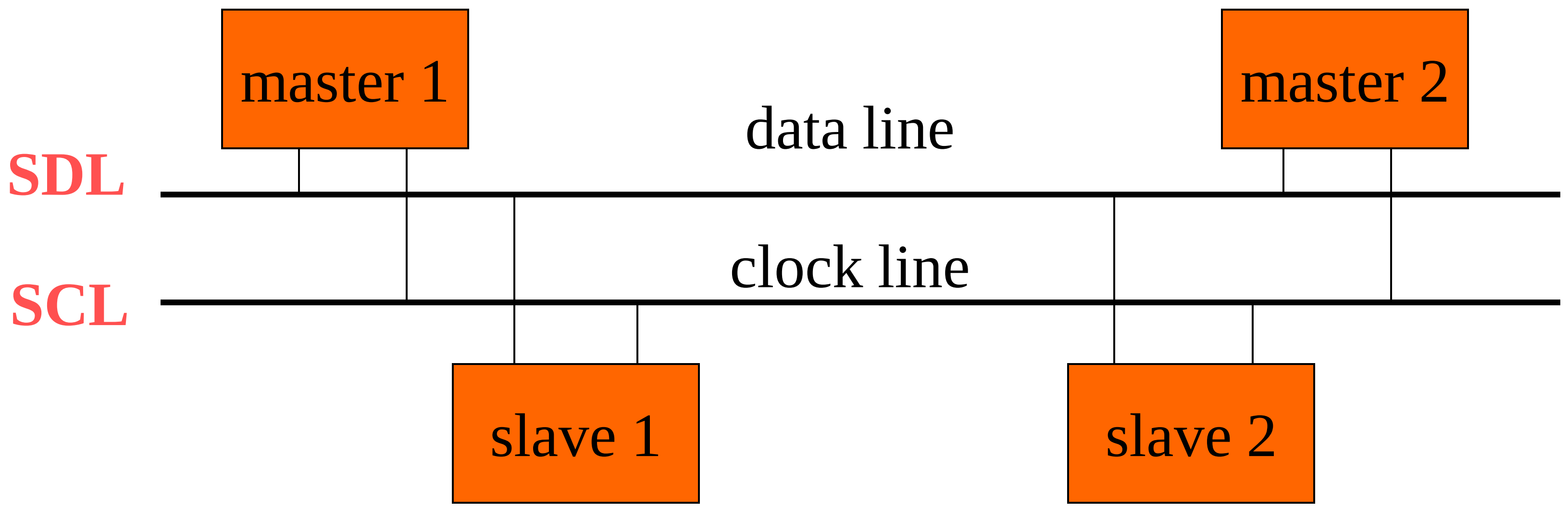
State diagrams for bus read



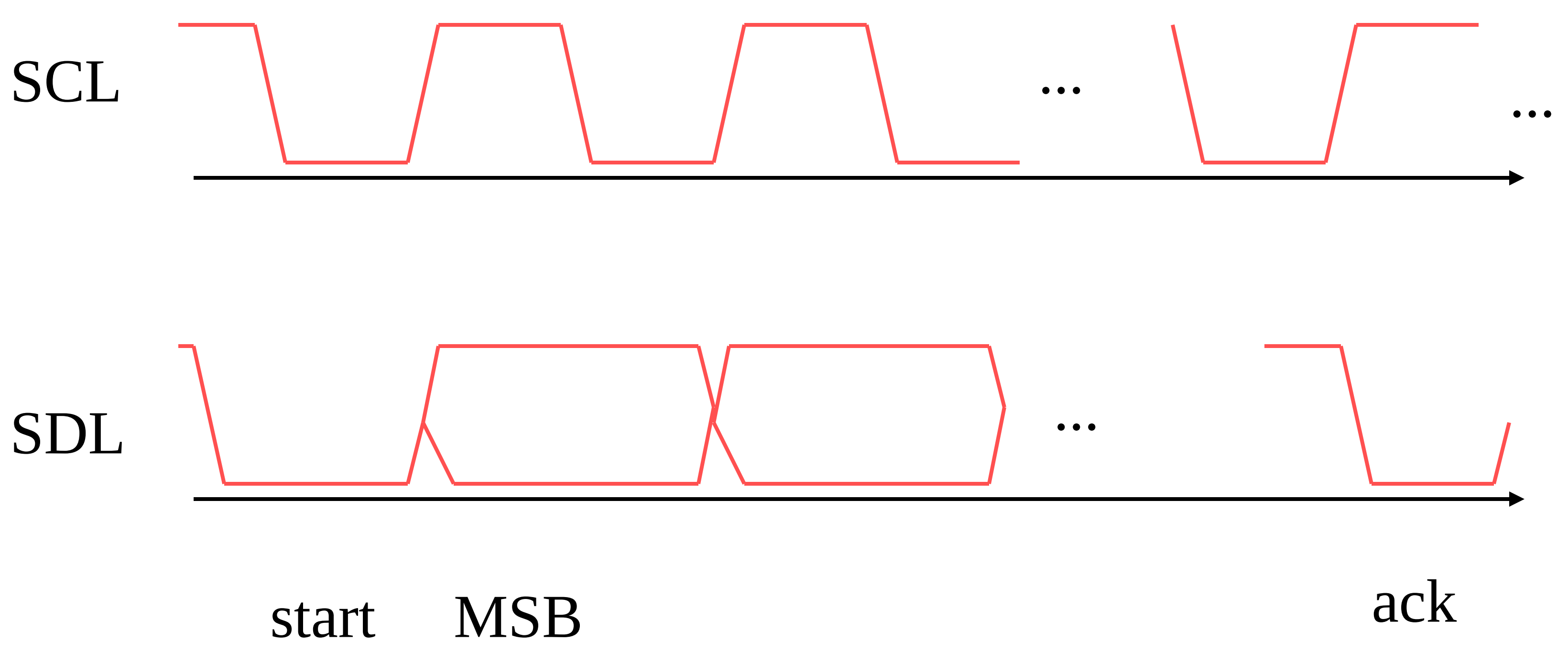
I^{2}C bus

- Designed for low-cost, medium data rate applications.
- Characteristics:
  - serial;
  - multiple-master;
  - fixed-priority arbitration.
- Several microcontrollers come with built-in I^{2}C controllers.

I^{2}C physical layer

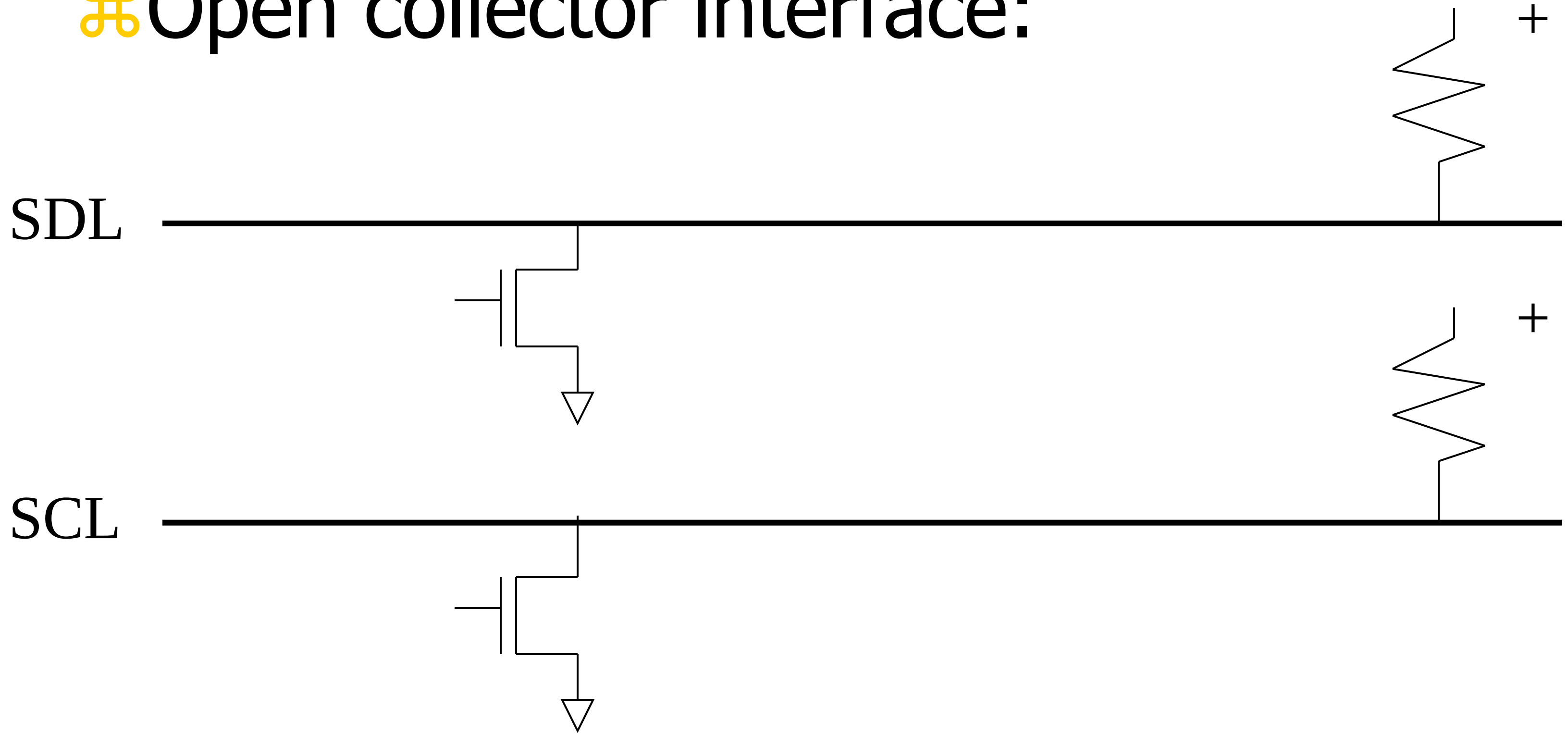


I^{2}C data format



I²C electrical interface

- Open collector interface:



I²C signaling

- Sender pulls down bus for 0.
- Sender listens to bus---if it tried to send a 1 and heard a 0, someone else is simultaneously transmitting.
- Transmissions occur in 8-bit bytes.

I^{2}C data link layer

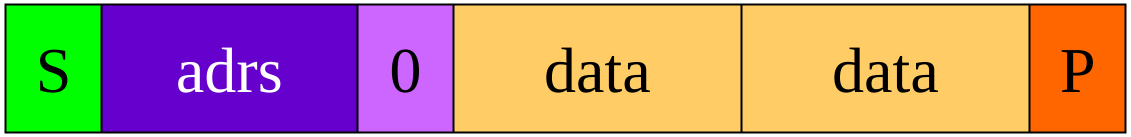
- Every device has an address (7 bits in standard, 10 bits in extension).
  - Bit 8 of address signals read or write.
- General call address allows broadcast.

I^{2}C bus arbitration

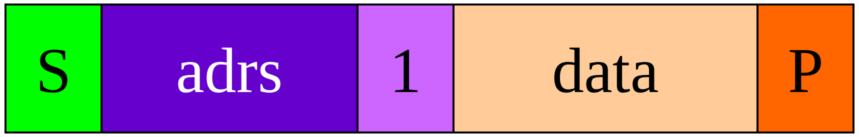
- Sender listens while sending address.
- When sender hears a conflict, if its address is higher, it stops signaling.
- Low-priority senders relinquish control early enough in clock cycle to allow bit to be transmitted reliably.

I^{2}C transmissions

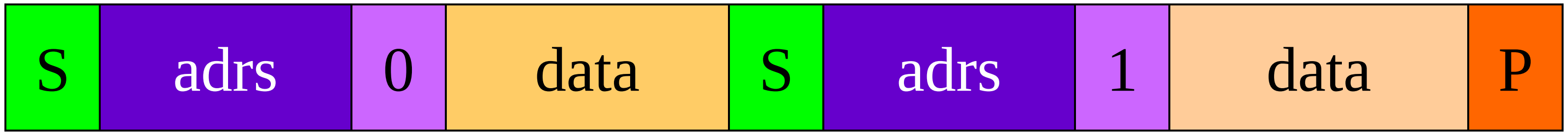
multi-byte write



read from slave



write, then read



CAN bus

- The **CAN bus** [Bos07] was designed for automotive electronics and was first used in production cars in 1991. CAN is very widely used in cars as well as in other applications.
- The CAN bus uses bit-serial transmission. CAN runs at rates of 1 MB/s over a twisted pair connection of 40 m.
- An optical link can also be used. The bus protocol supports multiple masters on the bus. Many of the details of the CAN and I2C buses are similar, but there are also significant differences.

Continued..

- As shown in Figure 8.22,each node in the CAN bus has its own electrical drivers and receivers that connect the node to the bus in wired-AND fashion.
- In CAN terminology, a logical 1 on the bus is called ***recessive*** and a logical 0 is ***dominant***.
- The driving circuits on the bus cause the bus to be pulled down to 0 if any node on the bus pulls the bus down (making 0 dominant over 1).
- When all nodes are transmitting 1s, the bus is said to be in the recessive state;when a node transmits a0, the bus is in the dominant state. Data are sent on the network in packets known
- as ***data frames***.

text book.pdf - Adobe Reader

Document Tools Window Help

422 (447 of 533) 75% Find

rks

Options

are similar, but there are also significant differences.

As shown in Figure 8.22, each node in the CAN bus has its own electrical drivers and receivers that connect the node to the bus in wired-AND fashion. In CAN terminology, a logical 1 on the bus is called ***recessive*** and a logical 0 is ***dominant***. The driving circuits on the bus cause the bus to be pulled down to 0 if any node on the bus pulls the bus down (making 0 dominant over 1). When all nodes are transmitting 1s, the bus is said to be in the recessive state; when a node transmits a 0, the bus is in the dominant state. Data are sent on the network in packets known as ***data frames***.

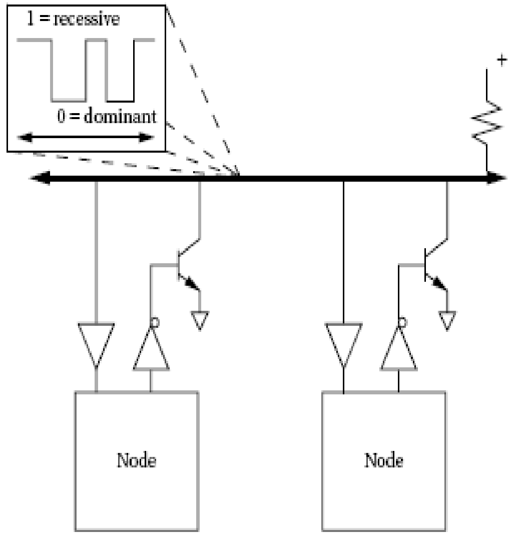


FIGURE 8.22

Physical and electrical organization of a CAN bus.

Continued..

- The format of a CAN data frame is shown in Figure 8.23.
- A data frame starts with a 1 and ends with a string of seven zeroes. (There are at least three bit fields between data frames.)
- The first field in the packet contains the packet's destination address and is known as the arbitration field.
- The destination identifier is 11 bits long.
- The trailing remote transmission request (RTR) bit is set to 0 if the data frame is used to request data from the device specified by the identifier. When
- ,

- When RTR1, the packet is used to write data to the destination identifier.
- The control field provides an identifier extension and a 4-bit length for the data field with a 1 in between The data field is from 0 to 64 bytes, depending on the value given inthe control field.
- A cyclic redundancy check (CRC) is sent after the data field forerror detection.
- The acknowledge field is used to let the identifier signal whetherthe frame was correctly received:The sender puts a recessive bit (1) in the ACK slot of the acknowledge field; if the receiver detected an error, it forces the value to a dominant (0) value.
- If the sender sees a 0 on the bus in theACK slot, it knows that it must retransmit. The ACK slot is followed by a single bit delimiter

between data frames.) The first field in the packet contains the packet's destination address and is known as the arbitration field. The destination identifier is 11 bits long. The trailing remote transmission request (RTR) bit is set to 0 if the data frame is used to request data from the device specified by the identifier. When RTR = 1, the packet is used to write data to the destination identifier. The control field provides an identifier extension and a 4-bit length for the data field with a 1 in between. The data field is from 0 to 64 bytes, depending on the value given in the control field. A cyclic redundancy check (CRC) is sent after the data field for error detection. The acknowledge field is used to let the identifier signal whether the frame was correctly received: The sender puts a recessive bit (1) in the ACK slot of the acknowledge field; if the receiver detected an error, it forces the value to a dominant (0) value. If the sender sees a 0 on the bus in the ACK slot, it knows that it must retransmit. The ACK slot is followed by a single bit delimiter followed by the end-of-frame field.

Control of the CAN bus is arbitrated using a technique known as Carrier Sense Multiple Access with Arbitration on Message Priority (CSMA/AMP). (As seen in

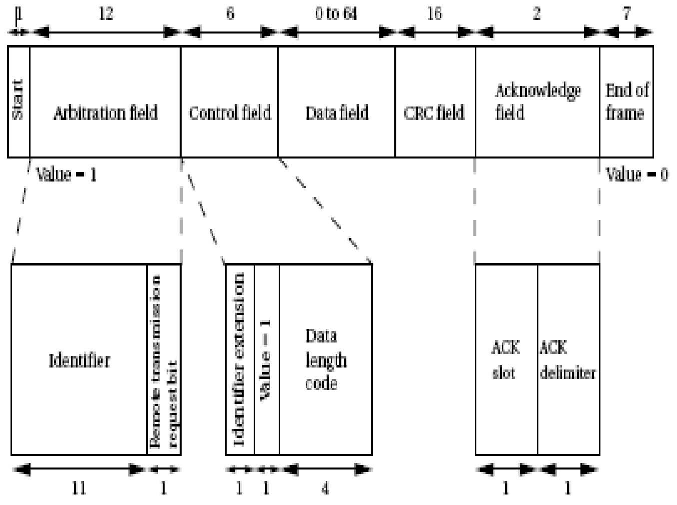


FIGURE 8.23

The CAN data frame format.

My Documents
wayne wolf text book...
ppt on can bus - Goo...
canbus
ch4-1-bus protocols

ELEVATOR CONTROLLER

- We design a multiple elevator system to increase the challenge. The configuration
- of a bank of elevators is shown in Figure 8.25.The *elevator car* is the unit that runs
- up and down the *hoistway* (also known as the shaft) carrying passengers;we will
- use *N* to represent the number of hoistways. Each car runs in a hoistway and can
- stop at any of *F* floors. (For convenience we will number the floors 1 through *F*,
- although some of the elevator doors may in fact be in the basement.) Every elevator
- car has a *car control panel* that allows the passengers to select floors to stop at.
- Each floor has a single *floor control panel* that calls for an elevator. Each floor also
- has a set of displays to show the current state of the elevator systems.

Continued…

- A set of displays to show the current state of the elevator systems.
- The user interface consists of the *elevator control panels*, floor control panels,and *displays.*
- The car control panels have *F* buttons to request the floors plus an emergency stop button.
- Each floor control panel has an up button and a downbutton that request an elevator going in the chosen direction.
- There is one displayper *hoistway* on each floor. Each *display* has an up light and a down light; if the elevator is idle,neither light is on.The displays for a *hoistway* always show the same state on all floors

Continued…

- The displays for a *hoistway* always show the same state on all floors.
- The elevator control system consists of two types of components. First, a single master controller governs the overall behavior of all elevators, and second, on each elevator a car controller runs everything that must be done within the car.
- The car controller must of course sense button presses on the car control panel, but itmust also sense the current position of the elevator.
- As shown in Figure 8.26, the car controller reads two sets of indicators on the wall of the elevator hoistway to sense position.

Continued...

- The *coarse indicators* run the entire length of the hoistway and a *sensor* determines when the elevator passes each one.
- *Fine indicators* are located only around the stopping point for each floor.
- There are 2S+1 fine indicators on each floor, one at the exact stopping point and S on each side of it.
- The sensor also reads fine indicators; it puts out separate signals for the coarse and fine indicators.
- The elevator system can stop at the proper position by counting coarse and fine indicators

428 (453 of 533) 75% Find

Each floor has a single *floor control panel* that calls for an elevator. Each floor also has a set of displays to show the current state of the elevator systems.

The user interface consists of the *elevator control panels*, floor control panels, and *displays*. The car control panels have *F* buttons to request the floors plus an emergency stop button. Each floor control panel has an up button and a down button that request an elevator going in the chosen direction. There is one display

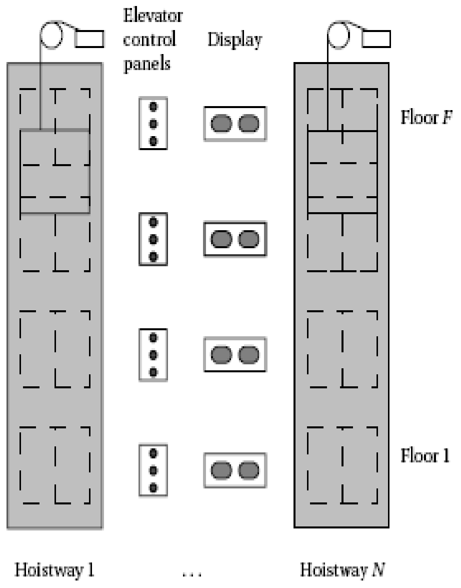


FIGURE 8.25

A bank of elevators.

per *hoistway* on each floor. Each *display* has an up light and a down light; if the elevator is idle, neither light is on. The displays for a *hoistway* always show the same state on all floors.

The elevator control system consists of two types of components. First, a single

My Documents | wayne wolf text book... | ppt on can bus - Goo... | ch4-1-bus protocols

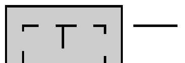
The elevator's movement is controlled by two motor control inputs: one for up and one for down. When both are disabled, the elevator does not move. The system should not enable both up and down on a single hoistway simultaneously.

The master controller has several tasks—it must read inputs from the floor control panels, send signals to the lights on the floor displays, read floor requests from the car controllers, and take inputs from the car sensors. Most importantly, it must tell the elevators when to move and when to stop. It must also schedule the elevators to efficiently answer passenger requests.

The basic requirements for the elevator system follow.

Name	Elevator system
Inputs	*F* floor control inputs, *N* position sensors, *N* car control panels, one master control panel
Outputs	*F* displays, *N* motor controllers
Functions	Responds to floor, car, and master control panels; operates cars safely
Performance	Control of elevators is time critical
Manufacturing cost	Cost of electronics is small compared to mechanical systems
Power	Not important
Physical size and weight	Cabling is the major concern

In this design, we are much more aware of the surrounding mechanical elements than we have been in previous examples. The electronics are clearly a small part of



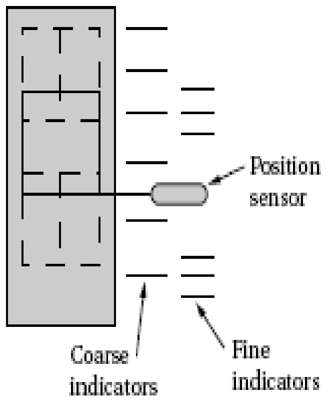


FIGURE 8.26

Sensing elevator position.

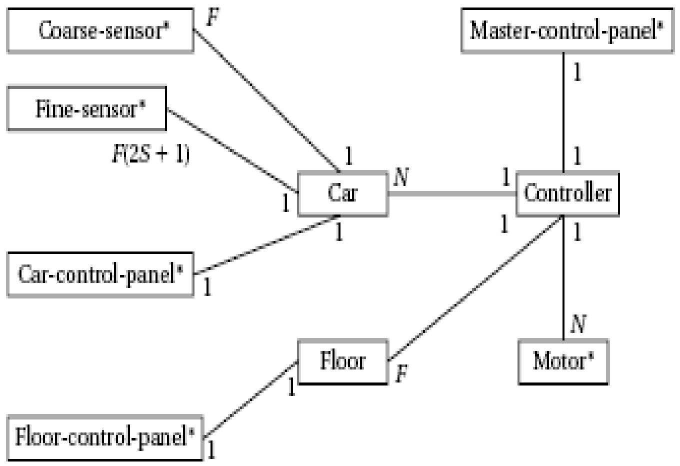


FIGURE 8.27

Basic class diagram for the elevator system.