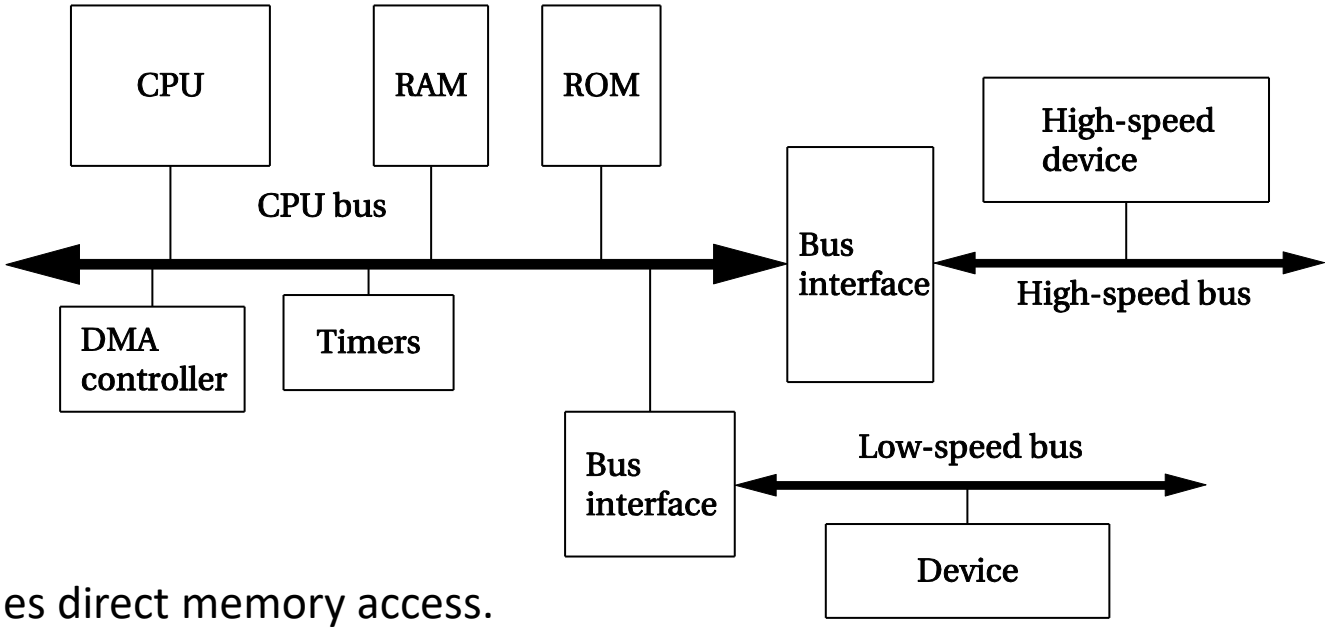
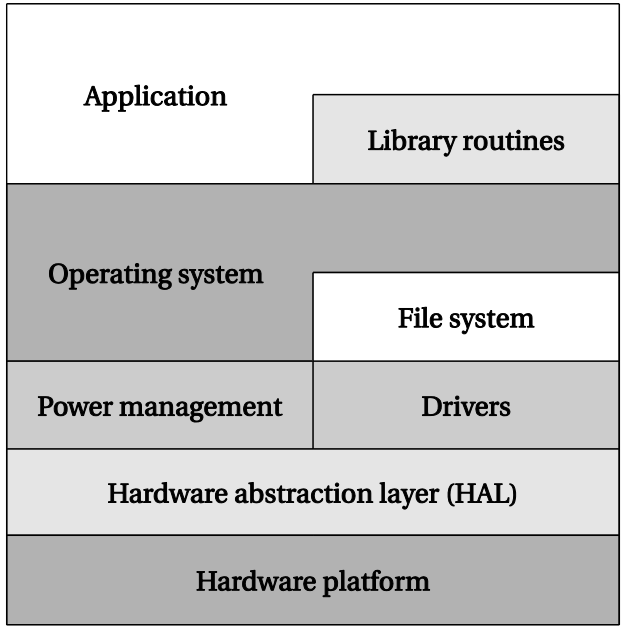
Computing Platform Architecture



DMA provides direct memory access.
Timers used by OS, devices.
Multiple busses connect CPU, memory to devices.

Platform Software Components

- Hardware and software are inseparable each needs the other to perform its function.
- These components range across many layers of abstraction.
- Layer diagrams are often used to describe the relationships between different software components in a system.
- The hardware abstraction layer (HAL) provides a basic level of abstraction from the hardware.
- Device drivers often use the HAL to simplify their structure.
- The application makes use of all these layers, either directly or indirectly.



Software layer diagram for an embedded system

CPU Busses

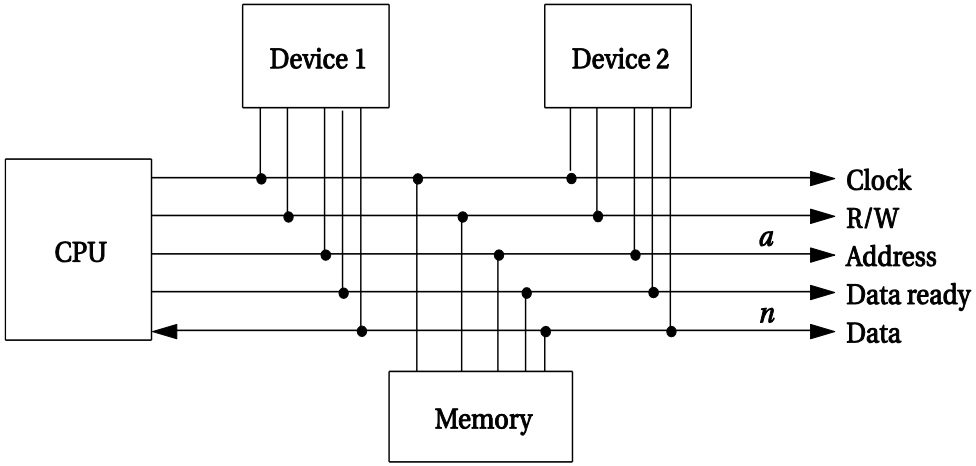
- The bus is the mechanism by which the CPU communicates with memory and devices
- A bus is, at a minimum, a collection of wires but it also defines a protocol by which the CPU, memory, and devices communicate.
- Bus allows CPU, memory, devices to communicate.
  - Shared communication medium.
- A bus is:
  - A set of wires.
  - A communications protocol.

Bus Protocols

- Bus protocol determines how devices communicate.
- A bus is a common connection between components in a system.
- The signals that make up the bus provide the necessary communication: the data itself, addresses, a clock, and some control signals.
- In a typical bus system, the CPU serves as the bus master and initiates all transfers
- 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.

Microprocessor Busses

- Clock provides synchronization.
- R/W is true when reading (R/W' is false when writing).
- 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.



Four-cycle Handshake

- The basic building block of most bus protocols is the four-cycle handshake.
- The handshake ensures that when two devices want to communicate, one is ready to transmit and the other is ready to receive.
- The handshake uses a pair of wires dedicated to the handshake:
  - enq (meaning enquiry)
  - ack (meaning acknowledge).

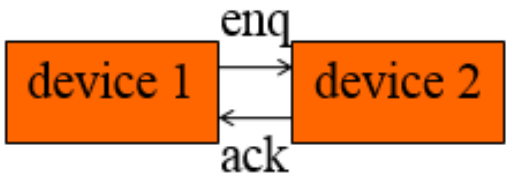
Extra wires are used for the data transmitted during the handshake.

Four-cycle Handshake

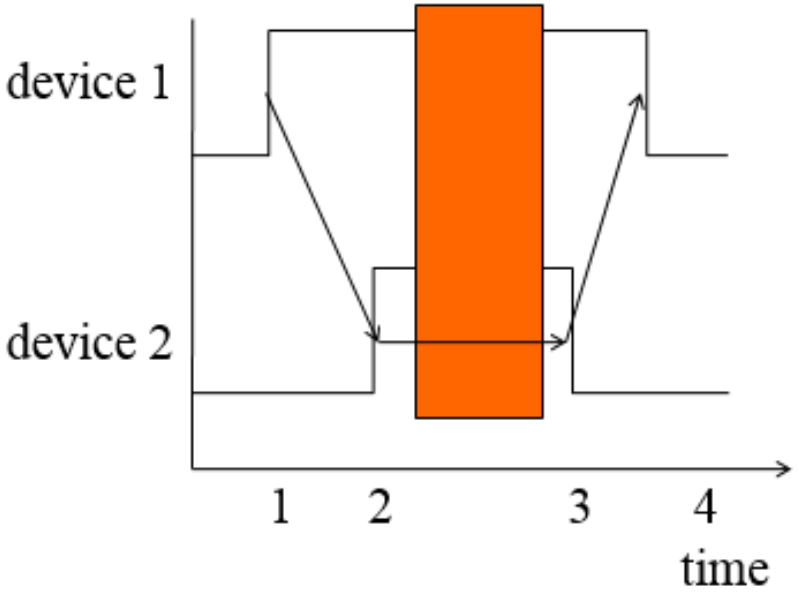
1. Device 1 raises its output to signal an enquiry, which tells device 2 that it should get ready to listen for data. (Device 1 raises enq)
2. When device 2 is ready to receive, it raises its output to signal an acknowledgment At this point, devices 1 and 2 can transmit or receive. (Device 2 responds with ack)
3. Once the data transfer is complete, device 2 lowers its output, signaling that " has received the data. (Device 2 lowers ack once it has finished)
4. After seeing that ack has been released, device 1 lowers its output. (Device 1 lowers enq)

At the end of the handshake, both handshaking signals are low, just as they were at the start of the handshake.

Four-cycle handshake

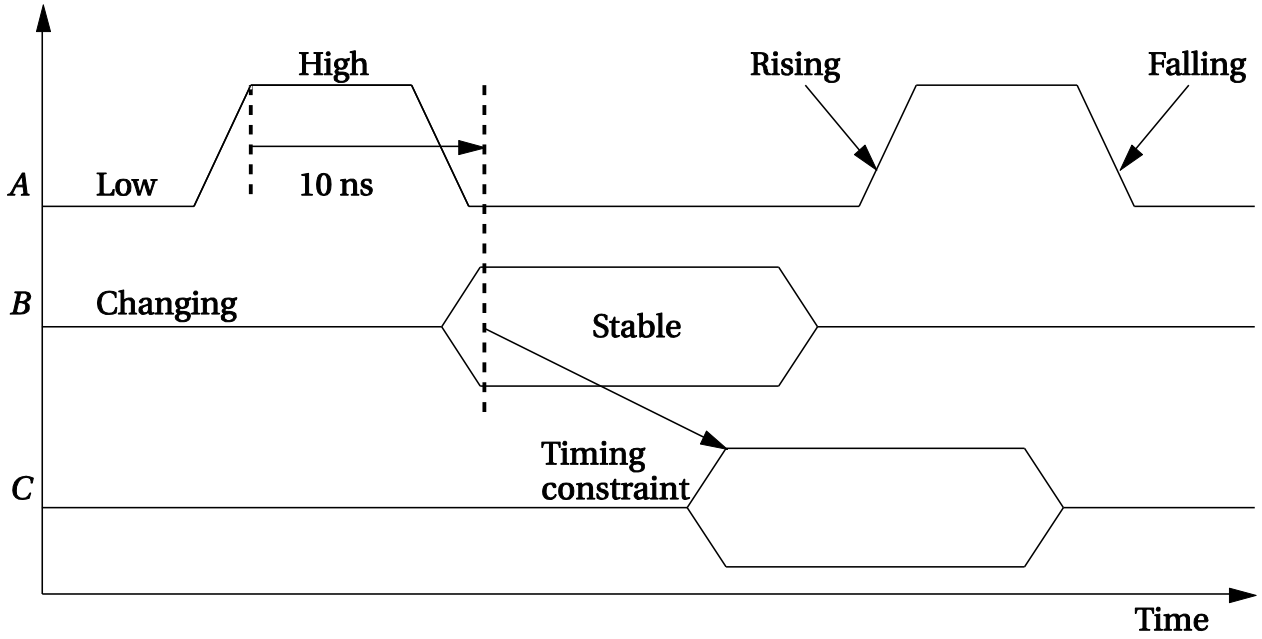


Structure

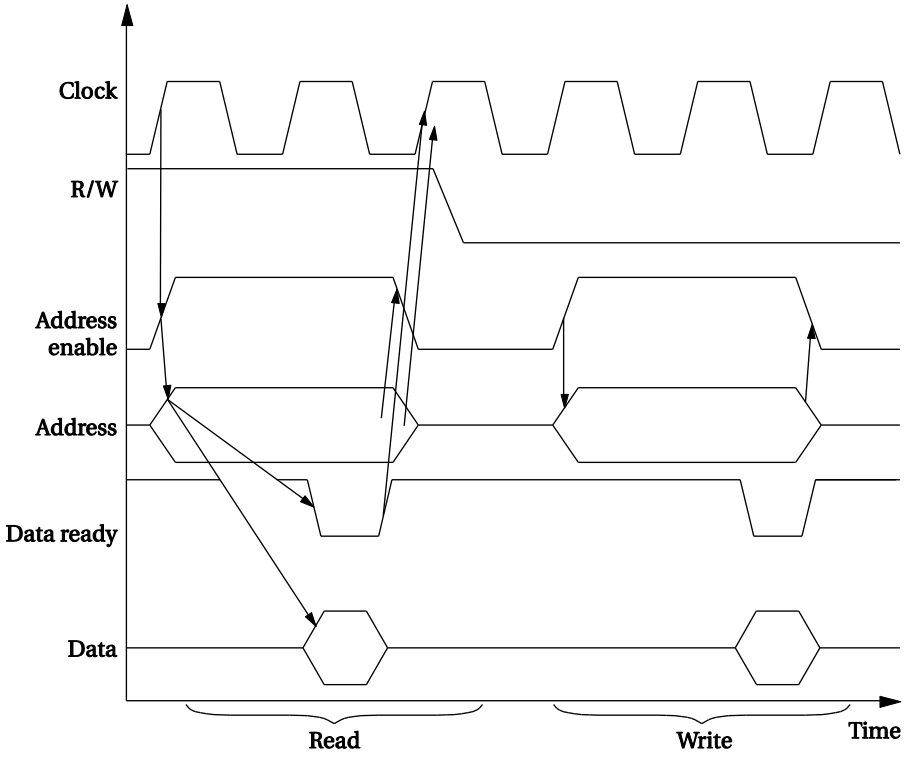


Behavior

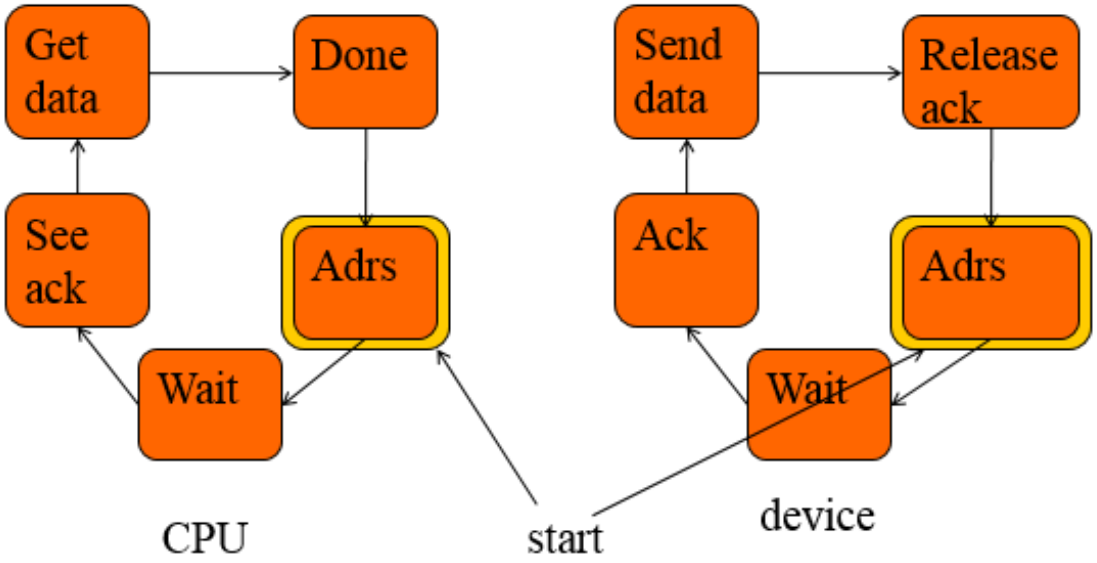
Timing Diagrams



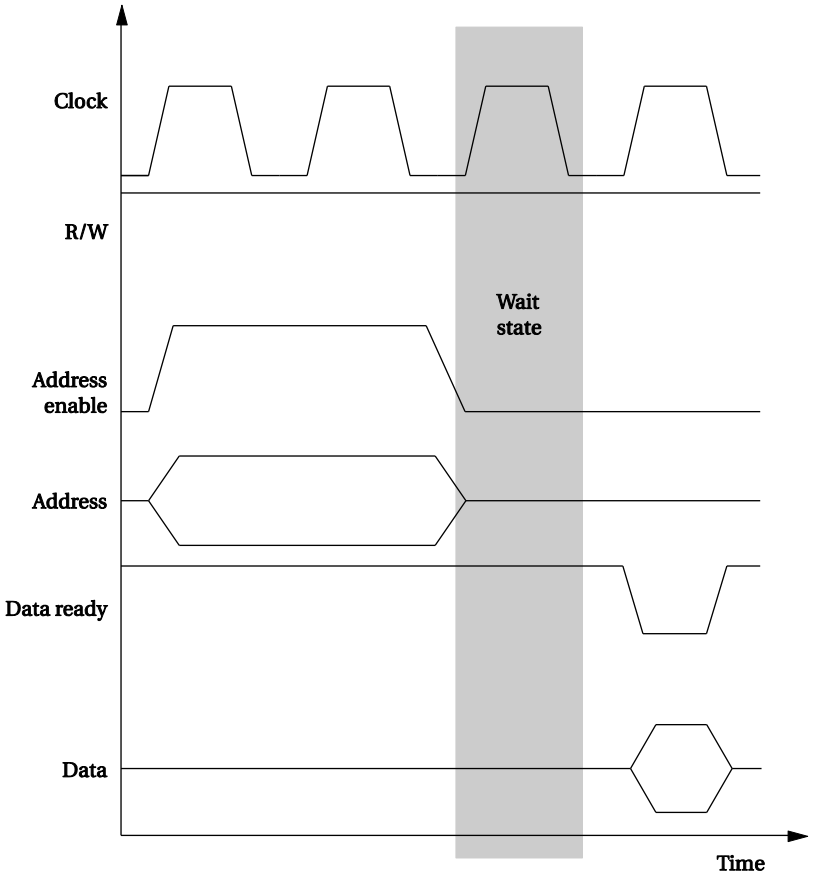
Bus Read/Write Operation



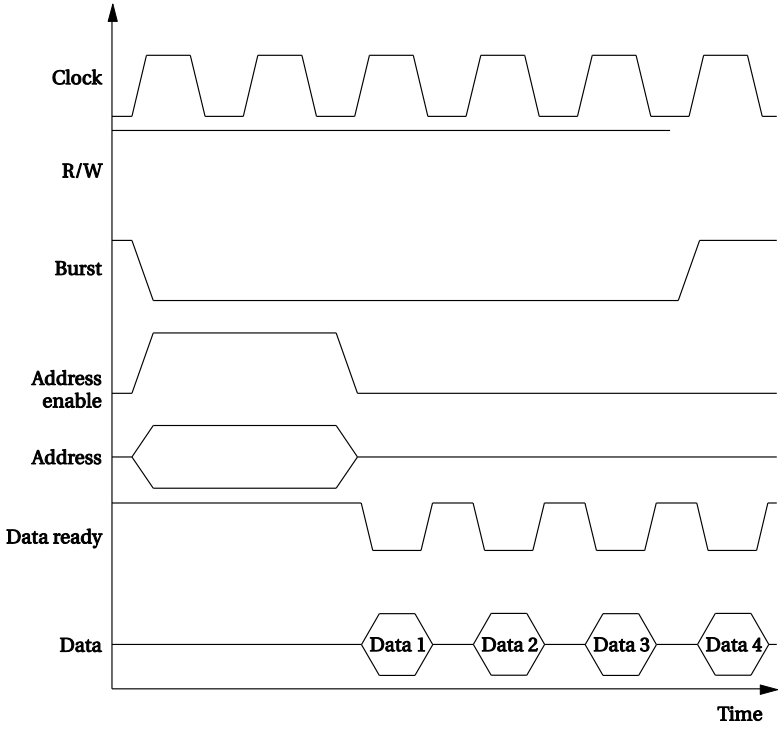
State Diagram for Bus Read



Wait Cycles for Busses



Bus Burst Operations



Bus multiplexing