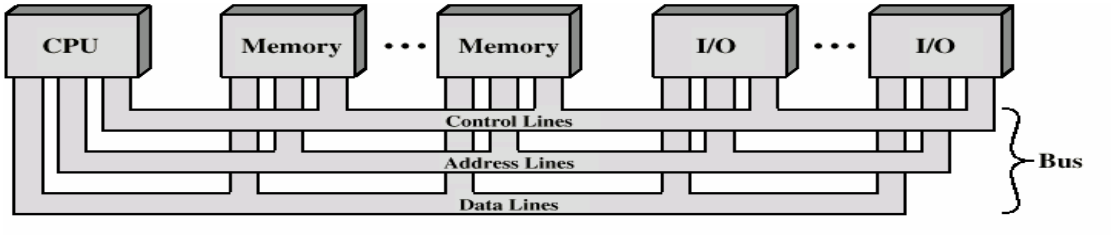
Bus Interconnection Schemes

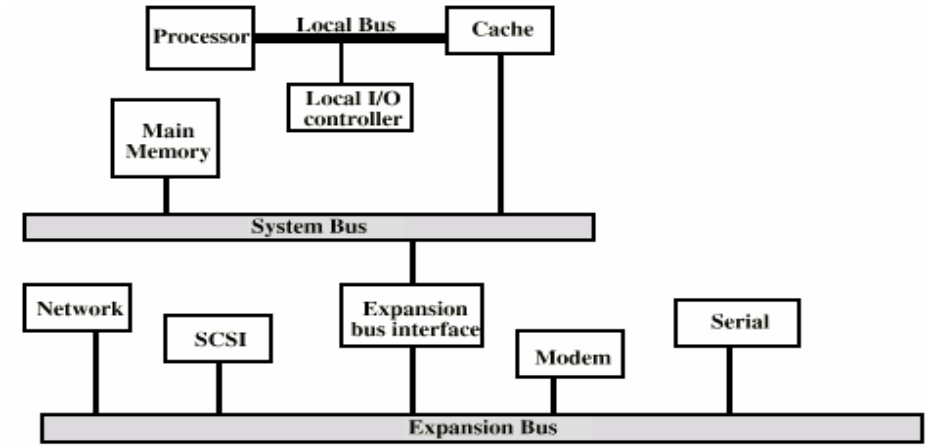
Single Bus



Single Bus Problems

- Lots of devices on one bus leads to:
  - Propagation delays
    - Long data paths mean that co-ordination of bus use can adversely affect performance
    - Bus may become bottleneck if aggregate data transfer approaches bus capacity
- Most systems use multiple buses to overcome these problems

Multiple-Bus



Traditional bus architecture

SCSI : small computer system interface to support local disk drives, CD-ROMs, and other peripherals
Serial: serial port to support a printer or scanner

It is possible to connect I/O controllers directly onto the system bus. A more efficient solution is to make use of one or more expansion buses for this purpose

- Allows system to support wide variety of I/O devices
- Insulates memory-to-process traffic from I/O traffic

Acknowledgement:
"Computer Organization and Architecture", by William Stallings, 6th edition

Bus Arbitration

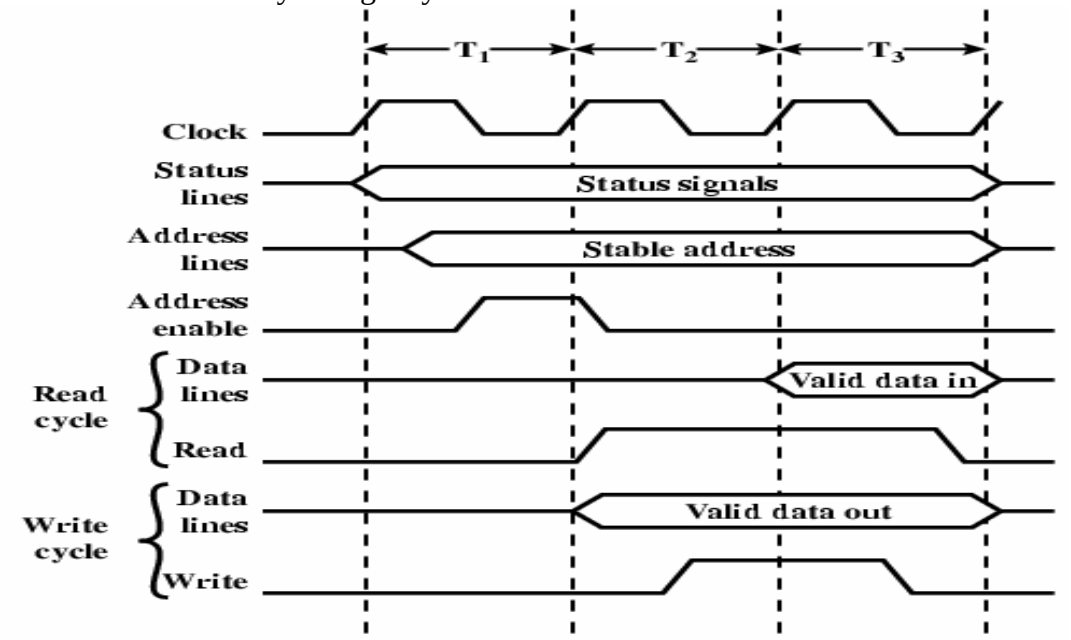
- More than one module may control the bus
  - e.g. CPU or DMA controller
- Only one module may control bus at one time
- Arbitration may be centralised or distributed
- Centralised:
  - Single hardware device controlling bus access
    - Bus Controller
    - Arbiter
  - May be part of CPU or separate
- Distributed:
  - Each module may claim the bus
  - Control logic on all modules

Timing

- Defines co-ordination of events on bus

Synchronous Bus Operation

- Events determined by clock signals
- Control Bus includes clock line
- A single 1-0 is a bus cycle
- All devices can read clock line
- Usually sync on leading edge
- Usually a single cycle for an event

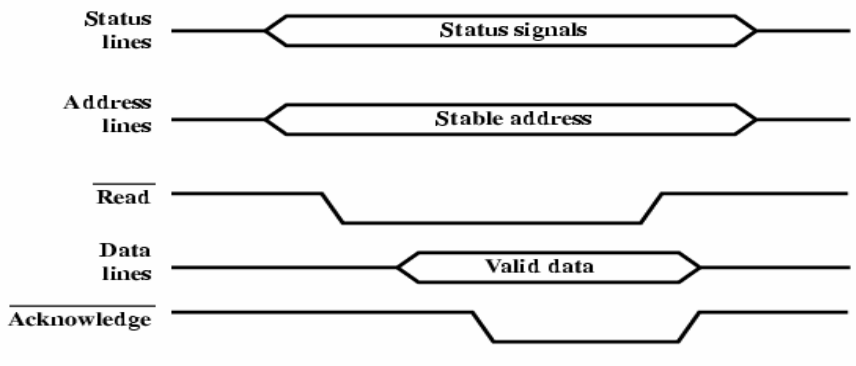


Acknowledgement:
"Computer Organization and Architecture", by William Stallings, 6th edition

Asynchronous Bus Operation

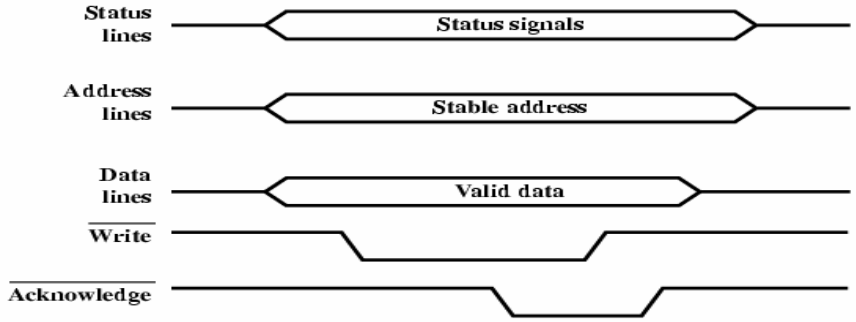
- Data transfer control on the bus is based on the use of a *handshake* between the master and the slave

System Bus Read Cycle



- Processor places address and status signals on the bus
- Issues Read command after these signals stabilize indicating presence of valid address and control signals
- Appropriate memory decodes the address and responds by placing data on the data line
- Once data lines have stabilized, memory module asserts the Acknowledge line to signal the processor that data are available
- Once data is read by the master, it deasserts the Read signal
- The memory module drops the data and acknowledge lines
- The master removes the address information

System Bus Write Cycle



- Master places the data on the data line at the same time as status and address lines
- Memory module responds to the write command by copying data
- Memory module then assert the acknowledge line
- The master drops the write signal and memory module drops the acknowledge signal

Acknowledgement:
"Computer Organization and Architecture", by William Stallings, 6th edition