Bus Types

1. Dedicated – assigned to a single function (e.g. address bus) or a physical subset of components (e.g. I/O bus connects all I/O modules).

2. Multiplexed – a bus can be used for both addresses and data. In this case, an address valid control line is needed to determine whether the data is an address or data. Time multiplexing is using the same lines for multiple purposes.

Method of Arbitration

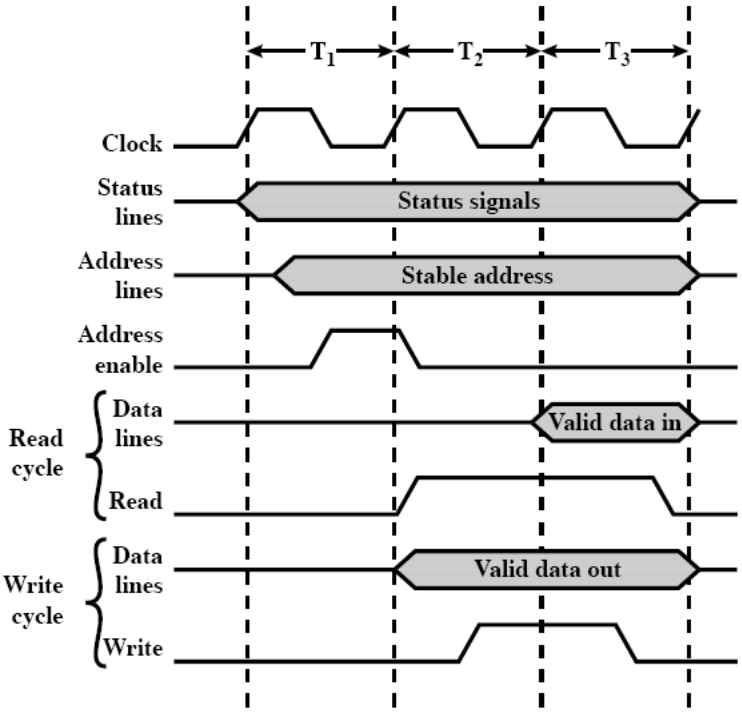
1. centralized – a single hardware device known as the bus controller (or arbiter) is responsible for allocating time on the bus.

2. distributed – with this form of communication there is no central controller. Instead each module has access control logic and the modules act together to share the bus.

In either case, the processor or I/O module is designated as a master for the purposes of transferring data and some other device is designated as a slave for the operation.

Timing

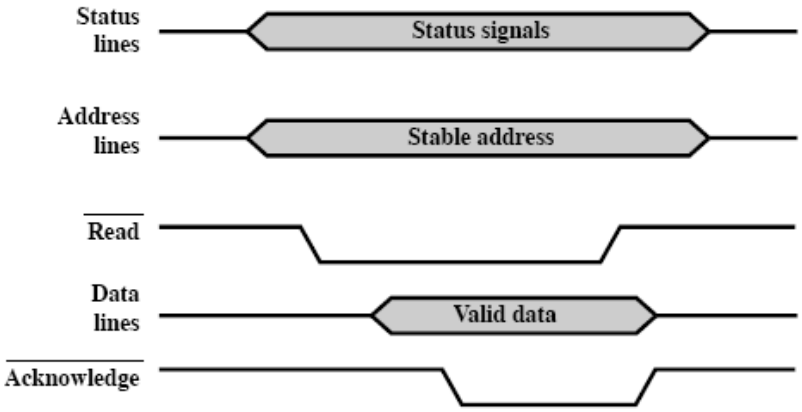
1. synchronous – the clock determines the occurrence of events on the bus as in Figure 3.19



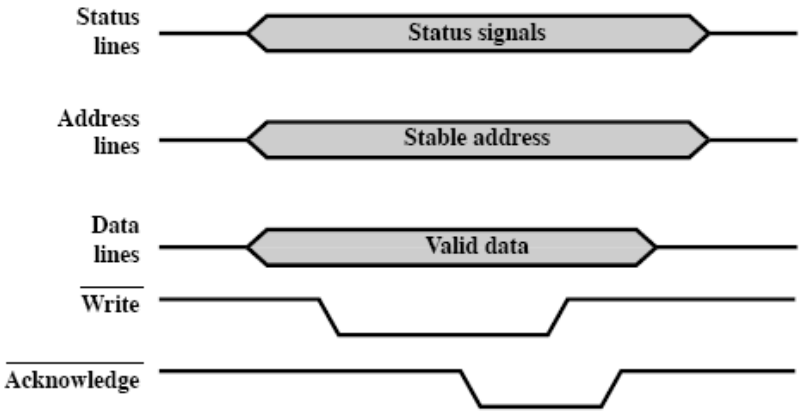
read – a) the processor places the memory address on the address bus at time T1, b) once the address is stable the processor issues an address enable signal on a read command at time T2, c) a memory module can recognize the address and place the data on the data bus at time T3.

write – a) the processor places the memory address on the address bus at time T1, b) the processor places the data on the data bus at time T2 and issues a write signal after the data has stabilized, c) the data is written at time T3.

2. asynchronous – timing does not involve a clock, instead one event on a bus follows and depends on some previous event as in Figure 3.20.



(a) System bus read cycle



(b) System bus write cycle

read – a) the processor places the address on the address bus and issues status signals, b) after signals have stabilized a read signal is issued, c) the data is placed on the data bus by the memory module and an acknowledge is sent from the memory module.

write – a) the processor places the address on the address bus, issues status signals, and places data on the data bus, b) after signals have stabilized a write signal is issued, c) the data is written to the memory module and an acknowledge is sent.

Note: Synchronous timing is easier to implement but less flexible. Asynchronous timing allows devices of all speeds to communicate more effectively.

Bus Width

Data bus – the greater the bus, the more information that can be transferred simultaneously (e.g. 32-bit vs 64-bit)

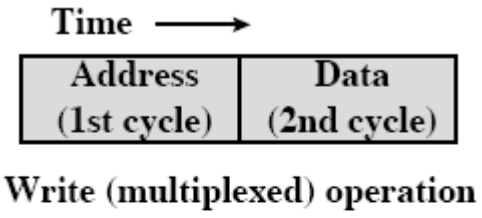
address bus – the greater the bus, the more locations that can be accessed.

Q1: A 32-bit buss allows the access of how many locations?

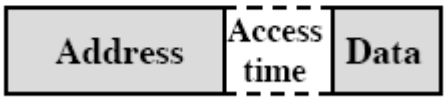
Data Transfer

It is the case that all buses support: write (master to slave) and read (slave to master)

The read and write operations can be multiplexed as follows from Figure 3.21:



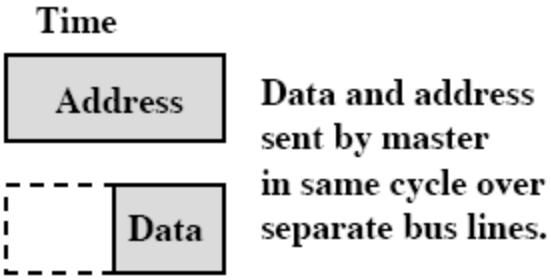
Write (multiplexed) operation



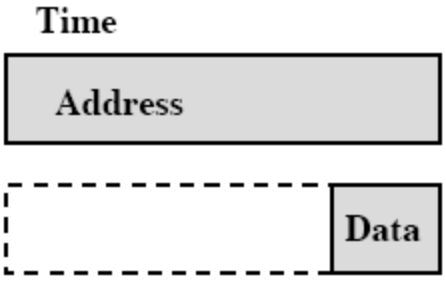
Read (multiplexed) operation

Q2: Explain each of the above operations and what multiplexed means.

The read & write operations can be non-multiplexed as follows from Figure 3.21:



Write (non-multiplexed) operation

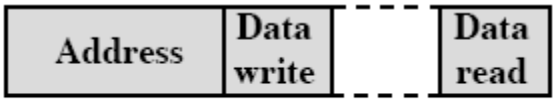


Read (non-multiplexed) operation

Some buses allow a read-modify-write which is a read followed by an immediate write using the same address as follows from Figure 3.21:

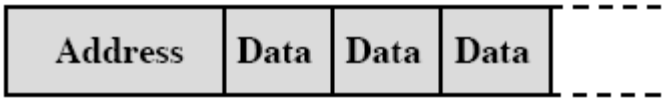
Read-modify-write operation

Some buses allow a read-after-write operation using the same address. The main reason for doing this is to check the value that was written as in Figure 3.21:



Read-after-write operation

Finally, a block transfer looks like the following from Figure 3.21



Block data transfer