Buses

Buses

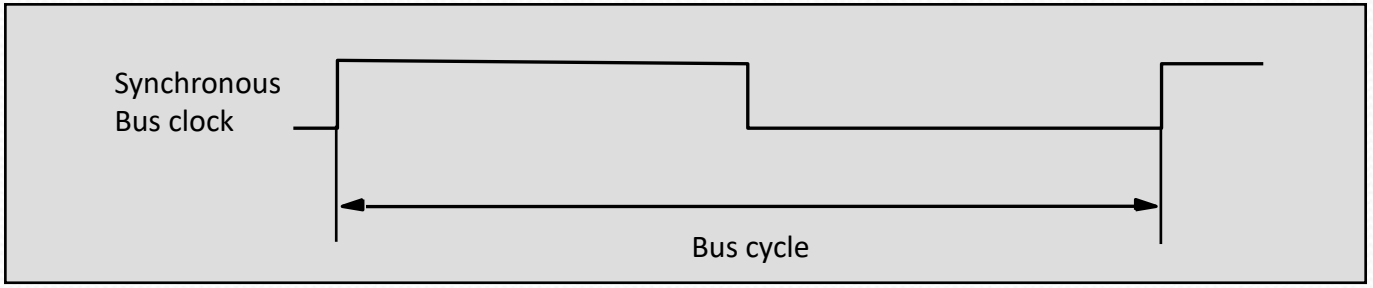
- BUS = Interconnection (connection wires) among Processor, main memory and any I/O devices.
- Processor, main memory, and I/O devices are **interconnected** by means of a bus.
- Bus provides a **Communication Path** for the **transfer of data.**
  - Bus also includes lines to support **Interrupts** and **Arbitration**.
- A **bus protocol** is the set of rules that govern the behavior of various devices connected to the bus, as to when to place information on the bus, when to assert control signals, etc.

Buses (contd..)

- Bus lines may be grouped into three types:
  - Data lines
  - Address lines
  - Control lines
- Control signals specify:
  - Whether it is a read or a write operation.
  - Required size of the data, when several operand sizes (byte, word, long word) are possible.
  - Timing information to indicate when the processor and I/O devices may place data or receive data from the bus.
- Schemes for Timing of data transfers over a bus can be classified into:
  - **Synchronous Bus,**
  - **Asynchronous Bus.**

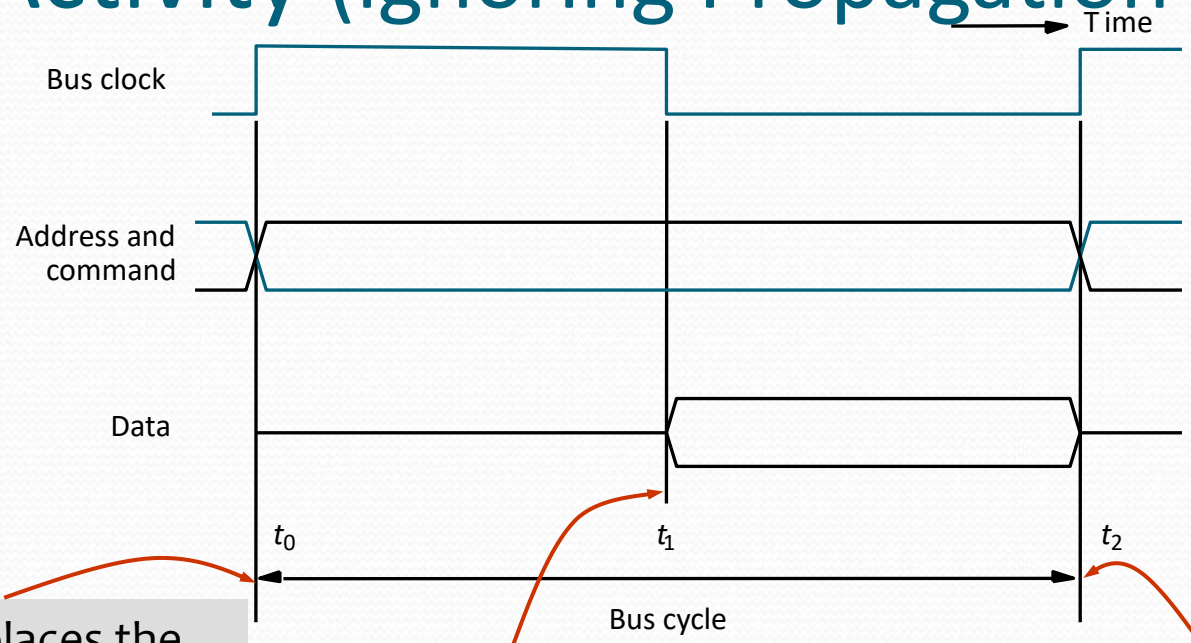
Synchronous & Asynchronous bus

- **Synchronous Bus** => A shared CLOCK line carries Clock Pulses (CP) that reach Every I/O device connected with the BUS. (e.g., 800MHz CP = 800 Millions Clock Pulses generated by a chip that reaches every I/O device connected to the bus)
- All I/O devices are synchronized with the clock pulses => All I/O operations (read/write) take place at the **Rising Edge** or **Falling Edges** of the shared clock Pulses.



- **Asynchronous Bus** => No Clock pulses. So no control line to carry any CP to the devices.
- Allows More flexibility of design. I/O devices can operate in widely varying speed, But they now have more responsibility, because they themselves have to produce the **Handshake Control Signals** to interact with each other for any I/O activity.

Synchronous bus: Timing Diagram of an Input Activity (ignoring Propagation Delay)



Master places the device address and command on the bus, and indicates that it is a Read operation.

Addressed **slave** places data on the data lines

Master "**strobe**s" the data on the data lines into its input buffer, for a Read operation.

- *In case of a Write operation, the **master** places the data on the bus along with the address and commands at time t_0.*
- *The **slave** strobes the data into its input buffer at time t_2.*

Synchronous bus (contd..)

- Once the master places the device address and command on the bus, it takes time for this information to propagate to the devices:
  - This time depends on the physical and electrical characteristics of the bus.
- Also, all the devices have to be given enough time to decode the address and control signals, so that the addressed slave can place data on the bus.
- **Width of the pulse $t_1 - t_0$ depends on:**
  - Maximum **Propagation Delay** between any two devices connected to the bus. Also, some time required by all the devices to decode the address and control signals, so that the addressed slave can respond at time t_1. (So, $t_1 - t_0$ **should be sufficiently large**)

Synchronous bus (contd..)

- At the end of the clock cycle, at time t_2, the master strobes the data on the data lines into its input buffer if it's a Read operation.
  - **"Strobe" means to capture the values of the data and store them into a buffer.**
- When data are to be loaded into a storage buffer register, the data should be available for a period longer than the setup time of the device.
- Width of the pulse t_2 - t_1 should be longer than:
  - Maximum propagation time of the bus plus
  - Set up time of the input buffer register of the master.

Synchronous bus (contd..)

- Data transfer has to be completed within one clock cycle.
  - Clock period t2 - to must be such that the longest propagation delay on the bus and the slowest device interface must be accommodated.
  - Forces all devices to operate at the speed of the **slowest** device!
- Processor just assumes that the data are available at t2 in case of a Read operation, or are read by the device in case of a Write operation.
  - What if the device is actually failed, and never really responded?

Synchronous bus (contd..)

- Most buses have control signals to represent a response from the slave.
- **Control signals** serve two purposes:
  - Inform the master that the slave has recognized the address, and is ready to participate in a data transfer operation.
  - Enable to adjust the duration of the data transfer operation based on the speed of the participating slaves.
- High-frequency bus clock is used:
  - Data transfer spans several clock cycles instead of just one clock cycle as in the earlier case.

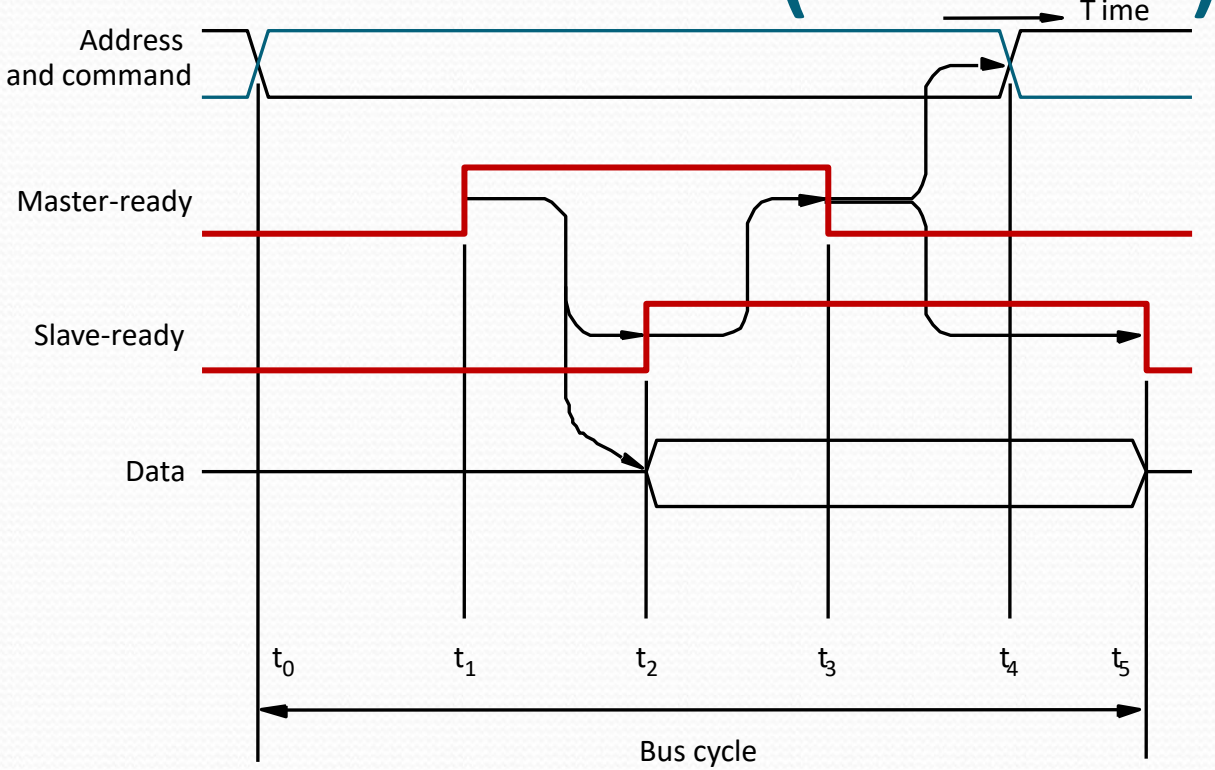
Asynchronous bus

- Data transfers on the bus is controlled by a handshake between the master and the slave.
- Common clock in the synchronous bus case is replaced by two timing Control Lines/Signals:
  - **Master-ready,**
  - **Slave-ready.**
- **Master-ready** signal is asserted by the master to indicate to the slave that it is ready for data transfer.
- **Slave-ready** signal is asserted by the slave in response to the master-ready from the master, and it indicates to the master that the slave is ready for a data transfer.

Asynchronous bus (contd..)

- Data transfer using the **Handshake protocol**:
  - Master places the address and command information on the bus
  - Asserts the Master-ready signal to indicate to the slaves that address and command info has been placed on the bus.
  - All devices on the bus decode the address.
  - Address slave performs the required operation, and informs the processor it has done so by asserting the Slave-ready signal.
  - Master removes all the signals from the bus, once Slave-ready is asserted.
  - If the operation is a Read operation, Master also strobes the data into its input buffer.

Asynchronous bus (contd..)



t_0 - Master places the address and command information on the bus.

t_1 - Master asserts the Master-ready signal. Master-ready signal is asserted at t_1 instead of t_0

t_2 - Addressed slave places the data on the bus and asserts the Slave-ready signal.

t_3 - Slave-ready signal arrives at the master.

t_4 - Master Strobes ("catches") the data, and removes the address & command information

t_5 - Slave receives the transition of the Master-ready signal from 1 to 0. It removes the data and the Slave-ready signal from the bus.

Asynchronous vs. Synchronous bus

- Advantages of asynchronous bus:
  - Eliminates the need for synchronization between the sender and the receiver.
  - Can accommodate varying delays automatically, supporting I/O devices with varying speed using the Slave-ready signal. ***(More Flexibility, but Slow!,)*** ***Slow because* two-round trip delays.**
- Disadvantages of asynchronous bus:
  - Data transfers using a synchronous bus involves **only one round trip delay**, and hence a synchronous bus can achieve faster rates. ***(Faster, but Little Flexibility: can't efficiently support very slow devices)***