Lecture 35

Asynchronous Data Transfer

Synchronous and Asynchronous Operations

Synchronous - All devices derive the timing information from common clock line
Asynchronous - No common clock

Asynchronous Data Transfer

Asynchronous data transfer between two independent units requires that *control signals* be transmitted between the communicating units *to indicate the time at which data is being transmitted*

Two Asynchronous Data Transfer Methods

Strobe pulse
- A strobe pulse is supplied by one unit to indicate the other unit when the transfer has to occur
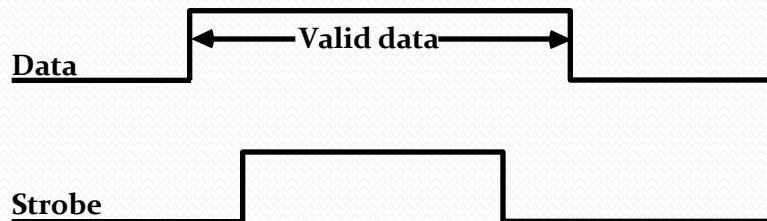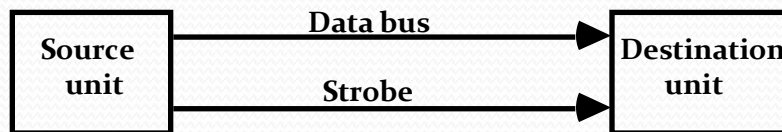
Handshaking
- A control signal is accompanied with each data being transmitted to indicate the presence of data
- The receiving unit responds with another control signal to acknowledge receipt of the data

Strobe Control

* Employs a single control line to time each transfer
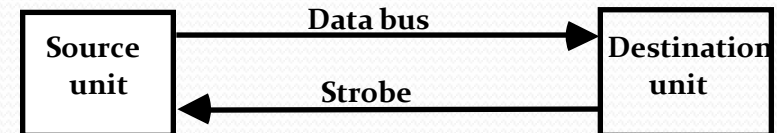*** The strobe may be activated by either the source or the destination unit**

Source-Initiated Strobe for Data Transfer
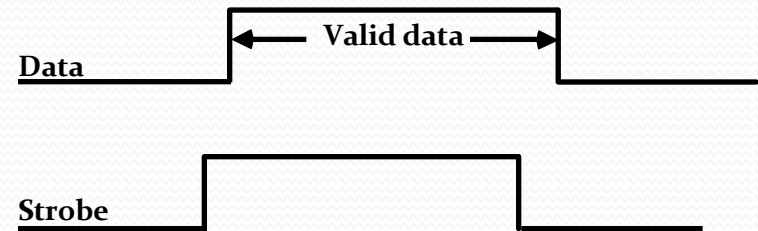
Block Diagram

Destination-Initiated Strobe for Data Transfer

Block Diagram

Timing Diagram

Which of the following methods of data transfer belongs to Asynchronous data transfer?

a) Strobe pulse
b) Handshaking
c) Both a and b
d) None of the above

Handshaking

Strobe Methods

<u>Source-Initiated</u>

The source unit that initiates the transfer has no way of knowing whether the destination unit has actually received data

<u>Destination-Initiated</u>

The destination unit that initiates the transfer no way of knowing whether the source has actually placed the data on the bus

To solve this problem, the *HANDSHAKE* method introduces a second control signal to provide a *Reply* to the unit that initiates the transfer
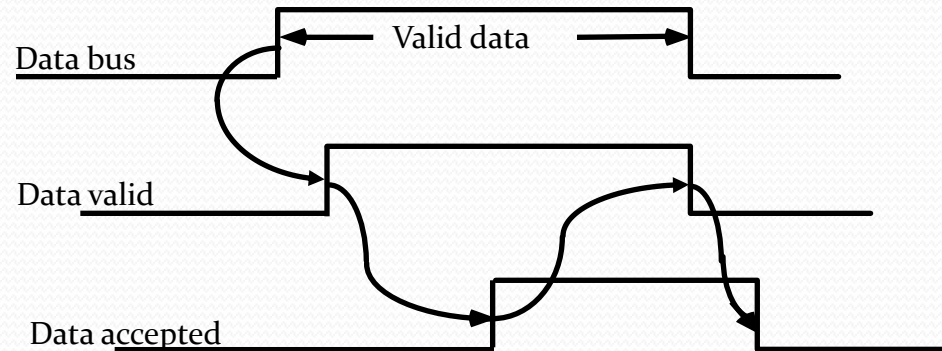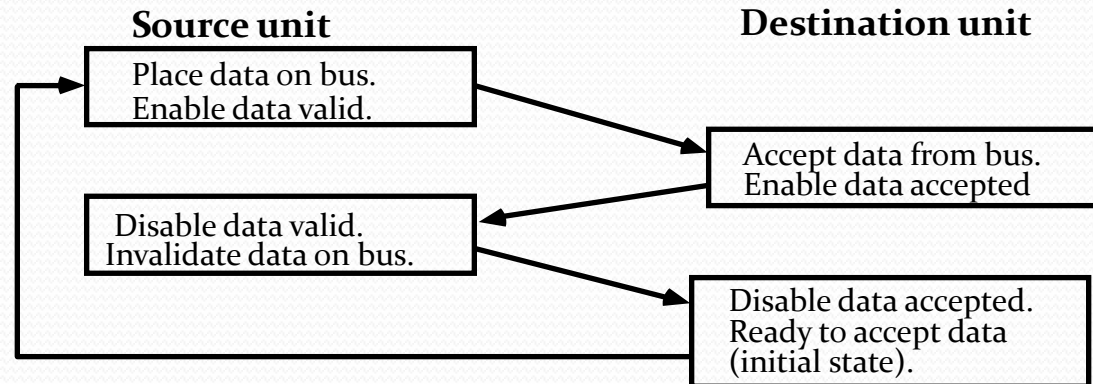
Lecture 35

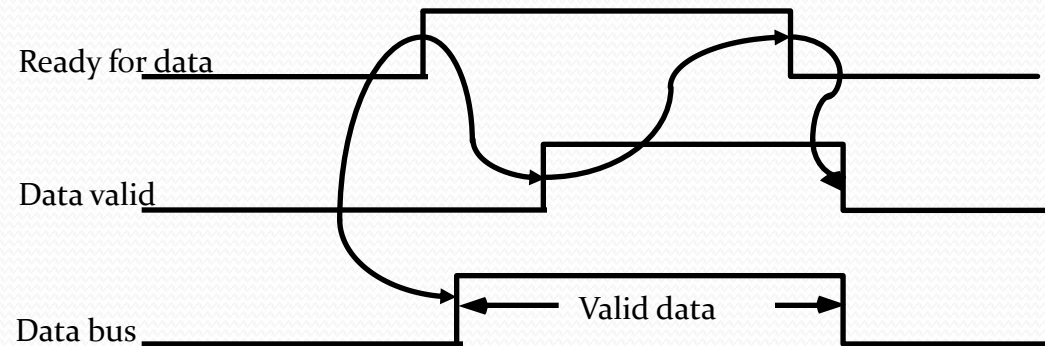# Source Initiated Transfer using Handshaking

Block Diagram

Source unit → Data valid → Destination unit
Source unit ← Data accepted ← Destination unit

Timing Diagram

Data bus ⟶ ←— Valid data —→

Data valid

Data accepted

Sequence of Events

Source unit   **Destination unit**

Place data on bus.
Enable data valid.

Accept data from bus.
Enable data accepted

Disable data valid.
Invalidate data on bus.

Disable data accepted.
Ready to accept data
(initial state).

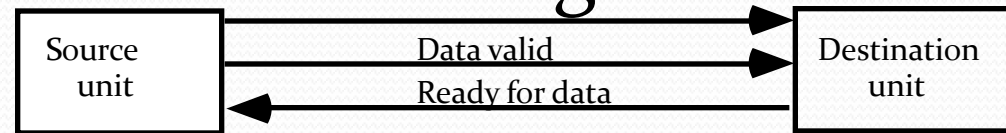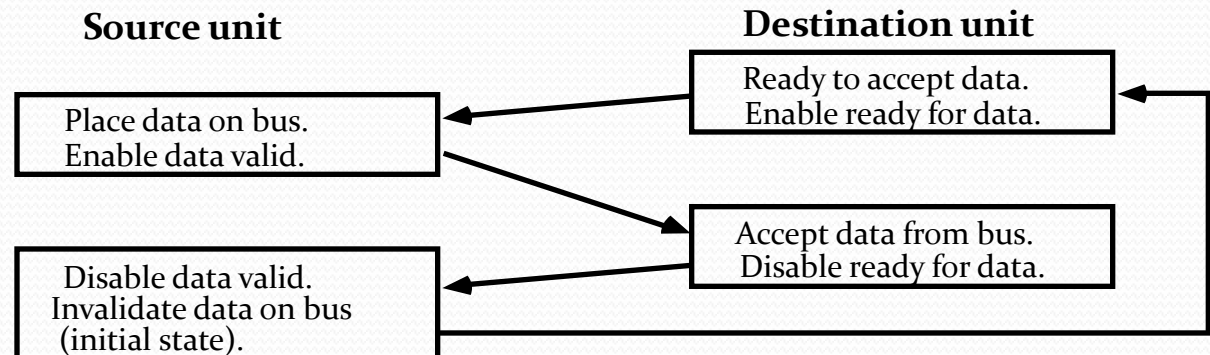*** Allows arbitrary delays from one state to the next**
* Permits each unit to respond at its own data transfer rate
* The rate of transfer is determined by the slower unit

Destination Initiated Transfer using Handshaking

Lecture 35

Block Diagram

Source unit	→
	Data valid →
	← Ready for data

→ Destination unit

Ready for data

Data valid

Data bus   Valid data

Sequence of Events

Source unit   **Destination unit**

Ready to accept data.
Enable ready for data.

Place data on bus.
Enable data valid.

Accept data from bus.
Disable ready for data.

Disable data valid.
Invalidate data on bus
(initial state).

* **Handshaking provides a high degree of flexibility and reliability because the successful completion of a data transfer relies on active participation by both units**
* **If one unit is faulty, data transfer will not be completed   -> Can be detected by means of a *timeout* mechanism**

Strobe pulse based data transfer method includes Acknowledgement. Is this statement true or false?

TRUE
FALSE

Lecture 35

Asynchronous Serial Transfer

Four Different Types of Transfer

Asynchronous serial transfer
Synchronous serial transfer
Asynchronous parallel transfer
Synchronous parallel transfer

Asynchronous Serial Transfer

- Employs special bits which are inserted at both ends of the character code
- Each character consists of three parts; Start bit;   Data bits;   Stop bits.
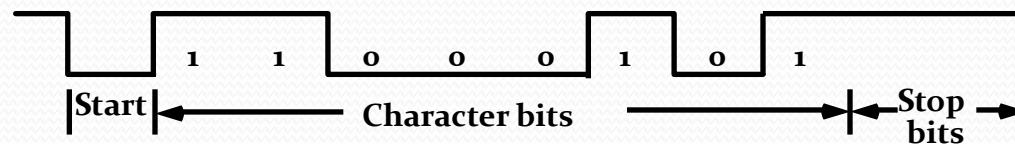

A character can be detected by the receiver from the knowledge of 4 rules;

- When data are not being sent, the line is kept in the 1-state (idle state)
- The initiation of a character transmission is detected by a *Start Bit* , which is always a 0
- The character bits always follow the *Start Bit*
- After the last character , a *Stop Bit* is detected when the line returns to the 1-state for at least 1 bit time

The receiver knows in advance the transfer rate of the bits and the number of information bits to expect

Lecture 36
Universal Asynchronous Receiver Transmitter

A typical asynchronous communication interface available as an IC
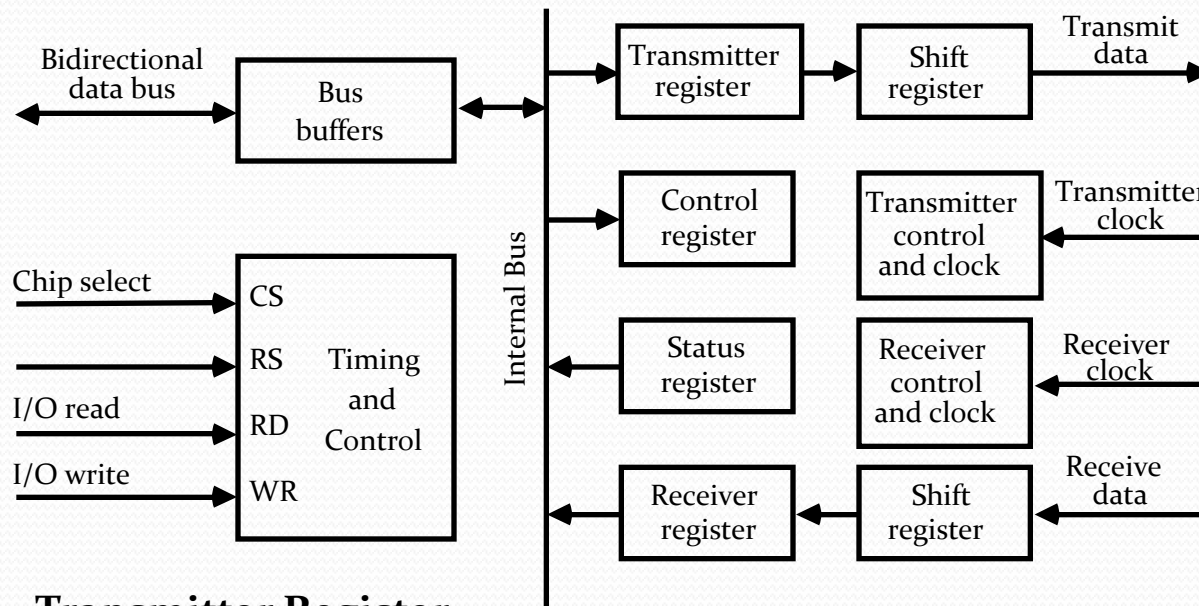
CS	RS	Oper.	Register selected
0	x	x	None
1	0	WR	Transmitter register
1	1	WR	Control register
1	0	RD	Receiver register
1	1	RD	Status register

Transmitter Register
   - Accepts a data byte(from CPU) through the data bus
   - Transferred to a shift register for serial transmission

Receiver
   - Receives serial information into another shift register
   - Complete data byte is sent to the receiver register

Status Register Bits
   - Used for I/O flags and for recording errors

Control Register Bits
   - Define baud rate, no. of bits in each character, whether to generate and check parity, and no. of stop bits