Synchronous & Asynchronous Data Transfer

Maninder Kaur

professormaninder@gmail.com

Synchronous Data Transfer

- In a digital system, the internal operations are synchronized by means of clock pulses supplied by a common pulse generator.
- In a computer, CPU and an I/O interface are designed independently of each other.
- If the registers in the interface share a common clock with the CPU registers, the data transfer between two units are said to be synchronous.

Asynchronous Data Transfer

- In a computer system, CPU and an I/O interface are designed independently of each other.
- When internal timing in each unit is independent from the other and when registers in interface and registers of CPU uses its own private clock.
- In that case the two units are said to be asynchronous to each other. CPU and I/O device must coordinate for data transfers.

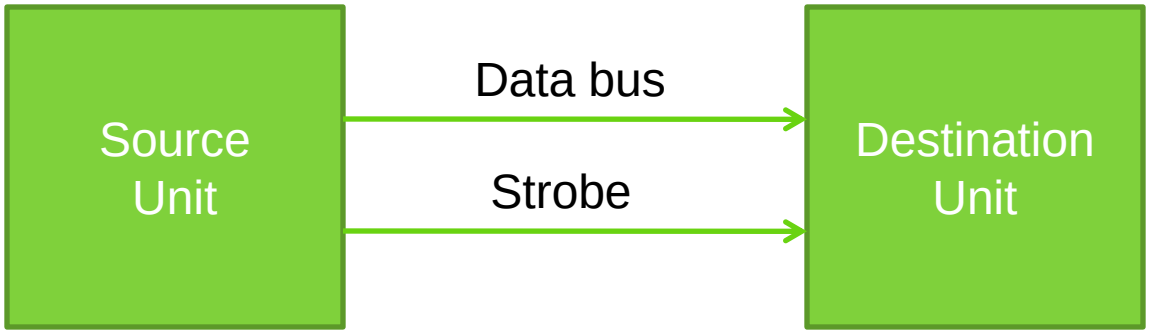
Methods Used In Asynchronous Data Transfer

- Strobe Control: This is one way of transfer i.e. by means of strobe pulse supplied by one of the units to indicate to the other unit when the transfer has to occur.

- Handshaking: This method is used to accompany each data item being transferred with a control signal that indicates the presence of data in the bus. The unit receiving the data item responds with another control signal to acknowledge receipt of the data.

Strobe Control

- Strobe control method of data transfer uses a single control signal for each transfer. The strobe may be activated by either the source unit or the destination unit.
  - Source Initiated Strobe
  - Destination Initiated Strobe

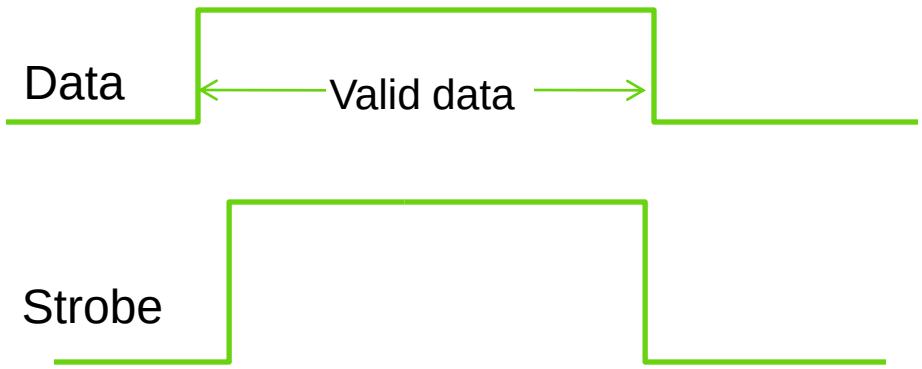


Source Initiated Strobe

- The *data bus* carries the binary information from source unit to the destination unit as shown below.

- The *strobe* is a single line that informs the destination unit when a valid data word is available in the bus.

Timing diagram



Source Initiated Strobe

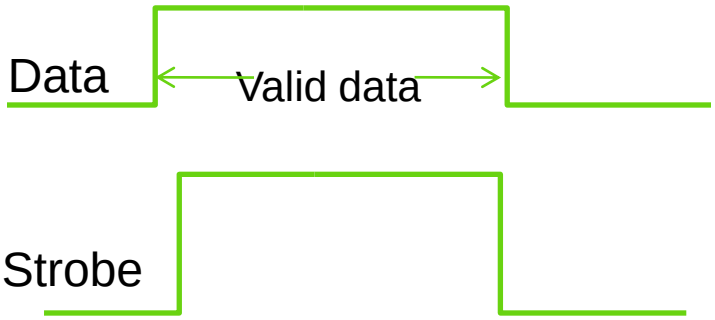
- The source unit first places the data on the bus.
- After a brief delay to ensure that the data settle to a steady value, the source activities the strobe pulse.
- The information of the data bus and the strobe signal remain in the active state for a sufficient time period to allow the destination unit to receive the data.
- The source removes the data from the bus for a brief period of time after it disables its strobe pulse.

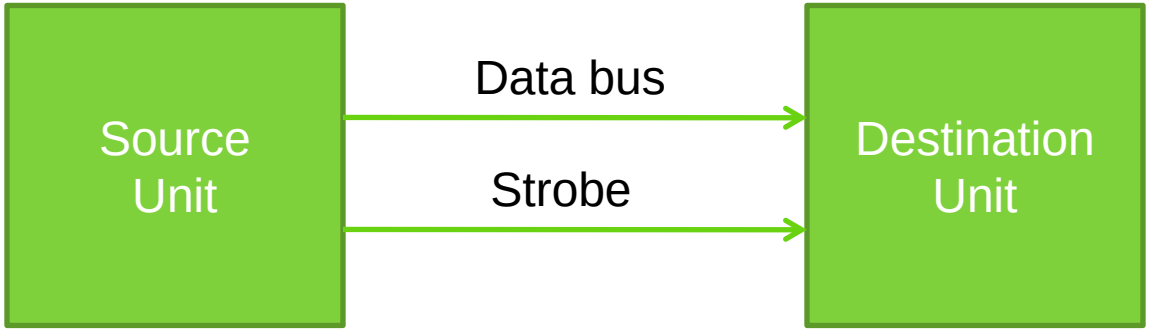
Destination Initiated Strobe

- First, the destination unit activates the strobe pulse, informing the source to provide the data.
- The source unit responds by placing the requested binary information on the unit to accept it.
- The data must be valid and remain in the bus long enough for the destination unit to accept it.
- The falling edge of the strobe pulse can be used again to trigger a destination register.
- The destination unit then disables the strobe. The source removes the data from the bus after a predetermined time interval.

Destination Initiated Strobe

Timing diagram





Block diagram

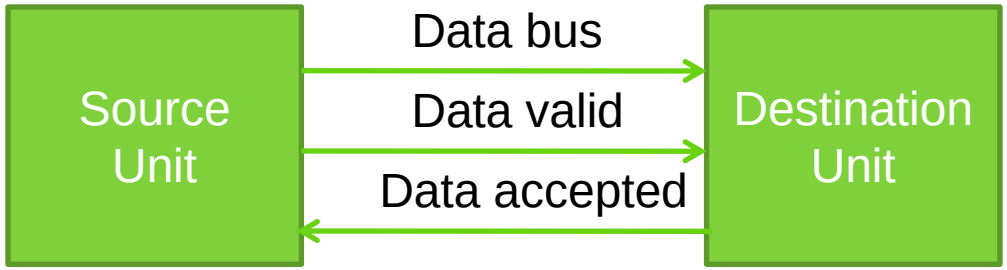
HANDSHAKING

- In case of source initiated data transfer under strobe control method, the source unit has no way of knowing whether destination unit has received the data or not.
- Similarly, destination initiated transfer has no method of knowing whether the source unit has placed the data on the data bus.
- Handshaking mechanism solves this problem by introducing a second control signal that provides a reply to the unit that initiate the transfer.
- There are two control lines in handshaking technique:
  - Source to destination unit
  - Destination to source unit

Source Initiated Transfer

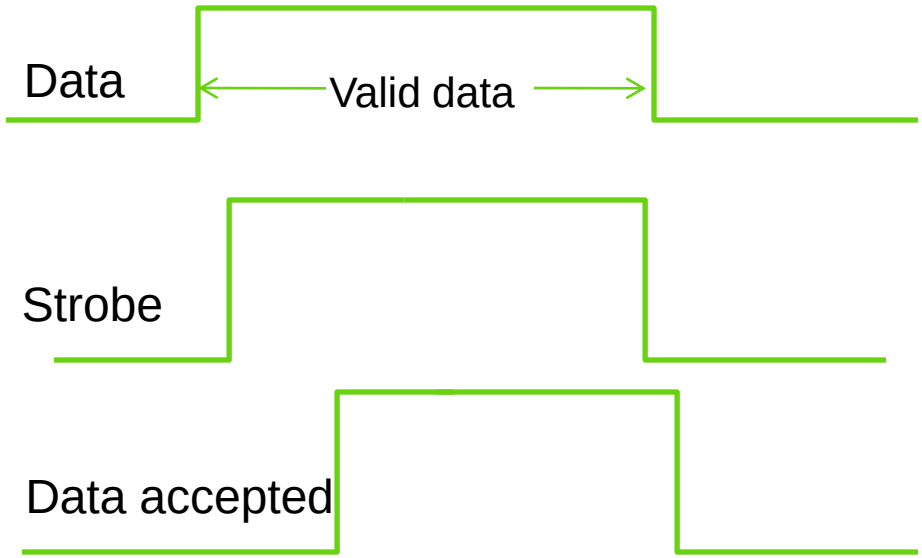
- Handshaking signals are used to synchronize the bus activities.
- The two handshaking lines are *data valid*, which is generated by the source unit, and *data accepted*, generated by the destination unit.
- The timing diagram shows exchange of signals between two units.

SOURCE INITIATED TRANSFER



Block diagram

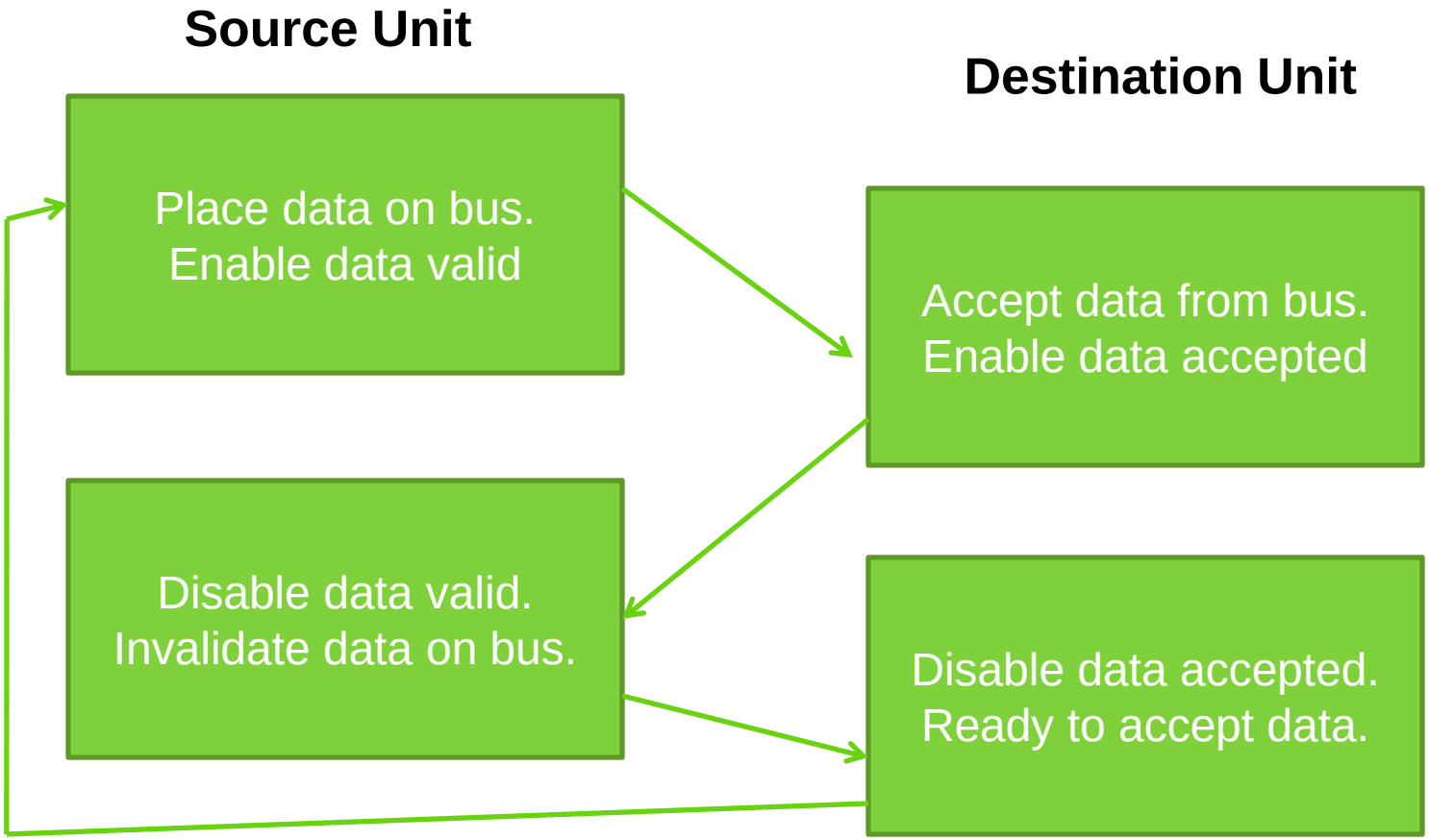
Timing diagram



Source Initiated Transfer Using Handshaking

- **The sequence of events:**
  - The source unit initiates the transfer by placing the data on the bus and enabling its data valid signal.
  - The data accepted signals is activated by the destination unit after it accepts the data from the bus.
  - The source unit then disables its data valid signal, which invalidates the data on the bus.
  - The destination unit the disables its data accepted signal and the system goes into its initial state.

Source Initiated Transfer Using Handshaking

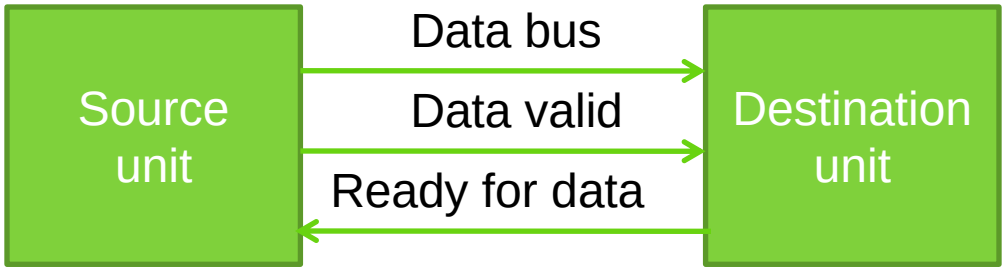


Sequence of events

Destination Initiated Transfer Using Handshaking

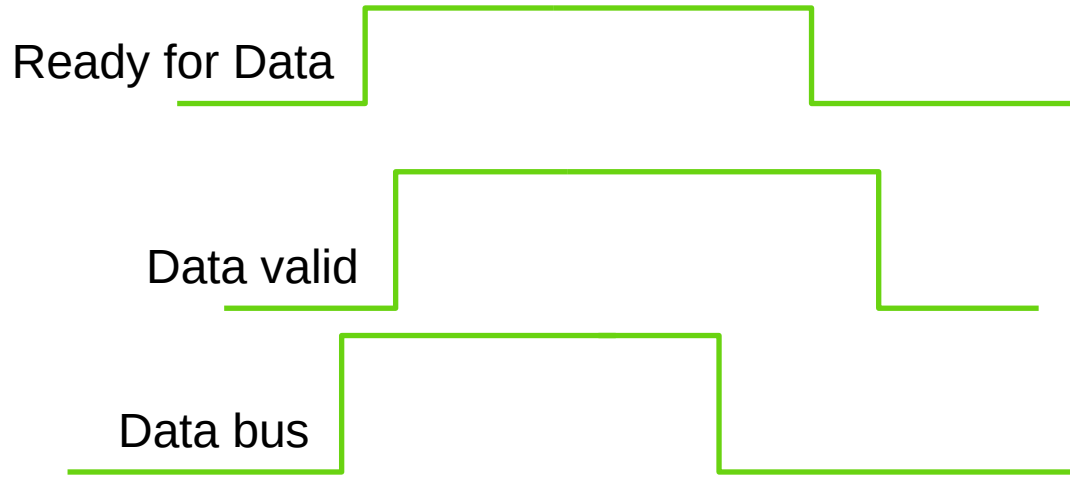
- In this case the name of the signal generated by the destination unit is *ready for data*.
- The source unit does not place the data on the bus until it receives the *ready for data* signal from the destination unit.
- The handshaking procedure follows the same pattern as in source initiated case. The sequence of events in both the cases is almost same except the *ready for signal* has been converted from *data accepted* in case of source initiated.

Destination Initiated Transfer

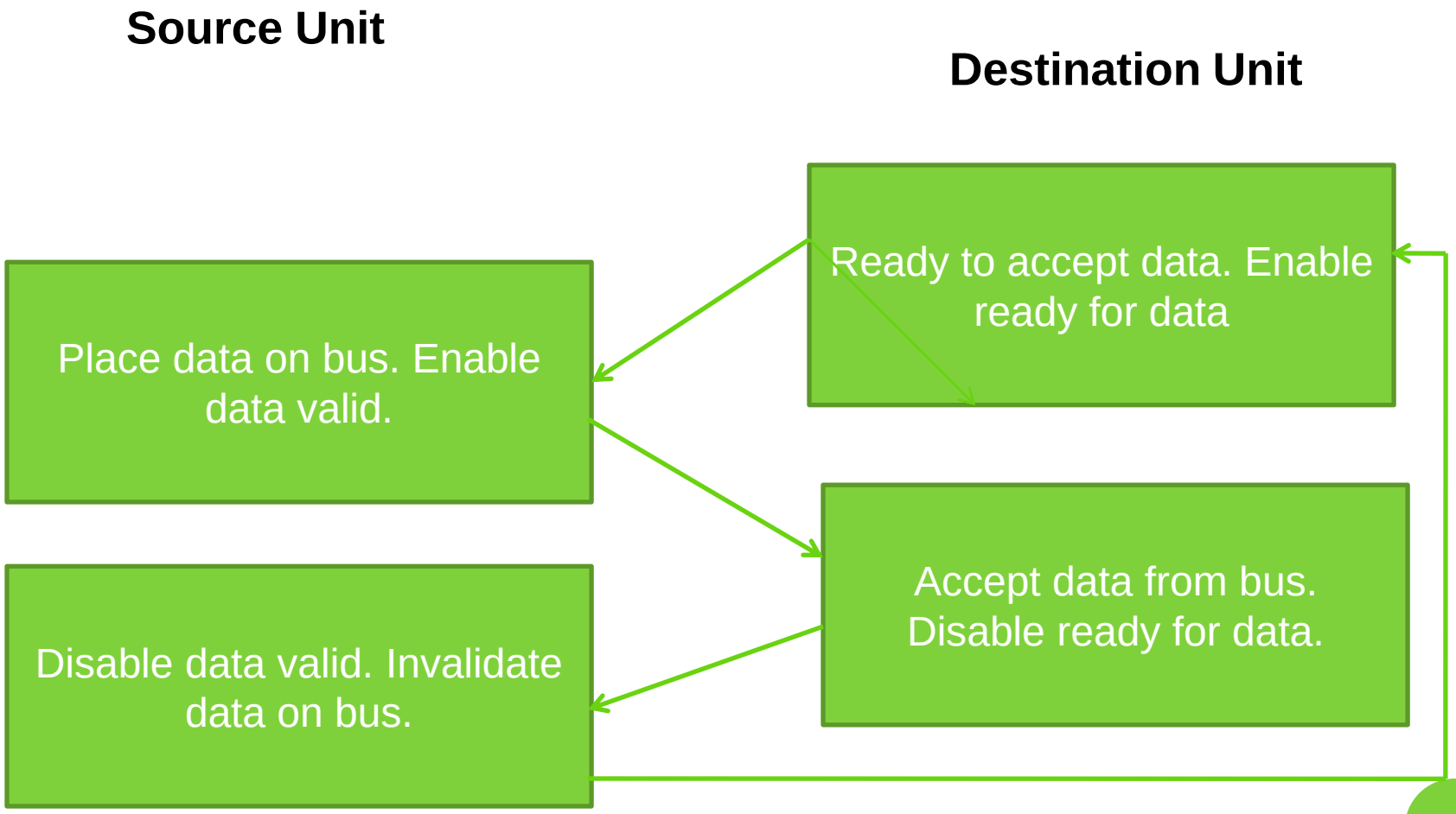


Block diagram

Timing diagram



DESTINATION INITIATED TRANSFER USING HANDSHAKING



THANKS A LOT