Asynchronous Data Transfer in Computer Organization

The internal operations in an individual unit of a digital system are synchronized using clock pulse. It means clock pulse is given to all r a unit. And all data transfer among internal registers occurs simultaneously during the occurrence of the clock pulse. Now, suppose any digital system are designed independently, such as CPU and I/O interface.

If the registers in the I/O interface share a common clock with CPU registers, then transfer between the two units is said to be synch most cases, the internal timing in each unit is independent of each other, so each uses its private clock for its internal registers. In this units are said to be asynchronous to each other, and if data transfer occurs between them, this data transfer is called **Asynchronous Da**

But, the Asynchronous Data Transfer between two independent units requires that control signals be transmitted between the comm so that the time can be indicated at which they send data. These two methods can achieve this asynchronous way of data transfer:

- **Strobe control:** A strobe pulse is supplied by one unit to indicate to the other unit when the transfer has to occur.
- **Handshaking:** This method is commonly used to accompany each data item being transferred with a control signal that indica bus. The unit receiving the data item responds with another signal to acknowledge receipt of the data.

The strobe pulse and handshaking method of asynchronous data transfer is not restricted to I/O transfer. They are used extensively occasions requiring the transfer of data between two independent units. So, here we consider the transmitting unit as a source and rec destination.

For example, the CPU is the source during output or write transfer and the destination unit during input or read transfer.

Therefore, the control sequence during an asynchronous transfer depends on whether the transfer is initiated by the source or by the de

So, while discussing each data transfer method asynchronously, you can see the control sequence in both terms when it is initiated b destination. In this way, each data transfer method can be further divided into parts, source initiated and destination initiated.

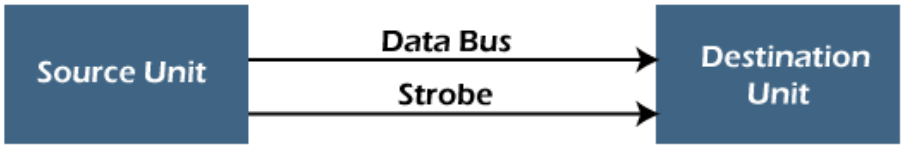
Asynchronous Data Transfer Methods

The asynchronous data transfer between two independent units requires that control signals be transmitted between the communi indicate when they send the data. Thus, the two methods can achieve the asynchronous way of data transfer.

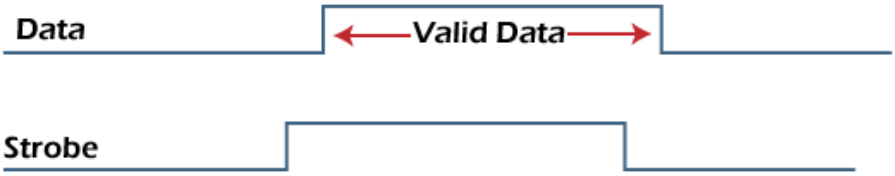
1. Strobe Control Method

The Strobe Control method of asynchronous data transfer employs a single control line to time each transfer. This control line is al strobe, and it may be achieved either by source or destination, depending on which initiate the transfer.

a. **Source initiated strobe:** In the below block diagram, you can see that strobe is initiated by source, and as shown in the timin source unit first places the data on the data



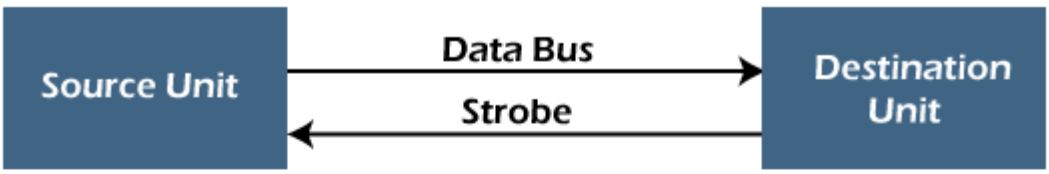
(a) Block Diagram



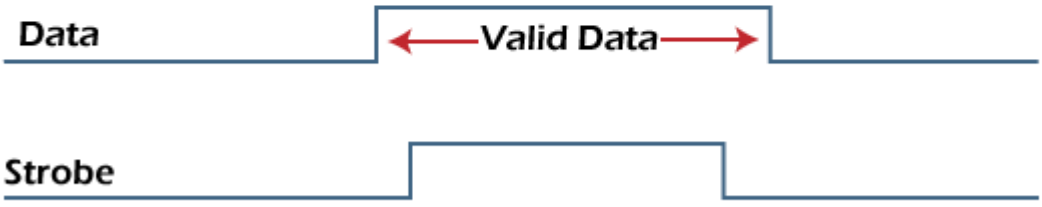
(b) Timing Diagram

After a brief delay to ensure that the data resolve to a stable value, the source activates a strobe pulse. The information on the data l control signal remains in the active state for a sufficient time to allow the destination unit to receiv The destination unit uses a falling edge of strobe control to transfer the contents of a data bus to one of its internal registers. The sourc data from the data bus after it disables its strobe pulse. Thus, new valid data will be available only after the strobe is e In this case, the strobe may be a memory-write control signal from the CPU to a memory unit. The CPU places the word on the data b the memory unit, which is the destination.

b. **Destination initiated strobe:** In the below block diagram, you see that the strobe initiated by destination, and in the timing destination unit first activates the strobe pulse, informing the source to provide



(a) Block Diagram



(b) Timing Diagram

The source unit responds by placing the requested binary information on the data bus. The data must be valid and remain o enough for the destination unit to accept The falling edge of the strobe pulse can use again to trigger a destination register. The destination unit then disables the strol source removes the data from the data bus after a determined time

In this case, the strobe may be a memory read control from the CPU to a memory unit. The CPU initiates the read operation memory, which is a source unit, to place the selected word into the data bus.

2. Handshaking Method

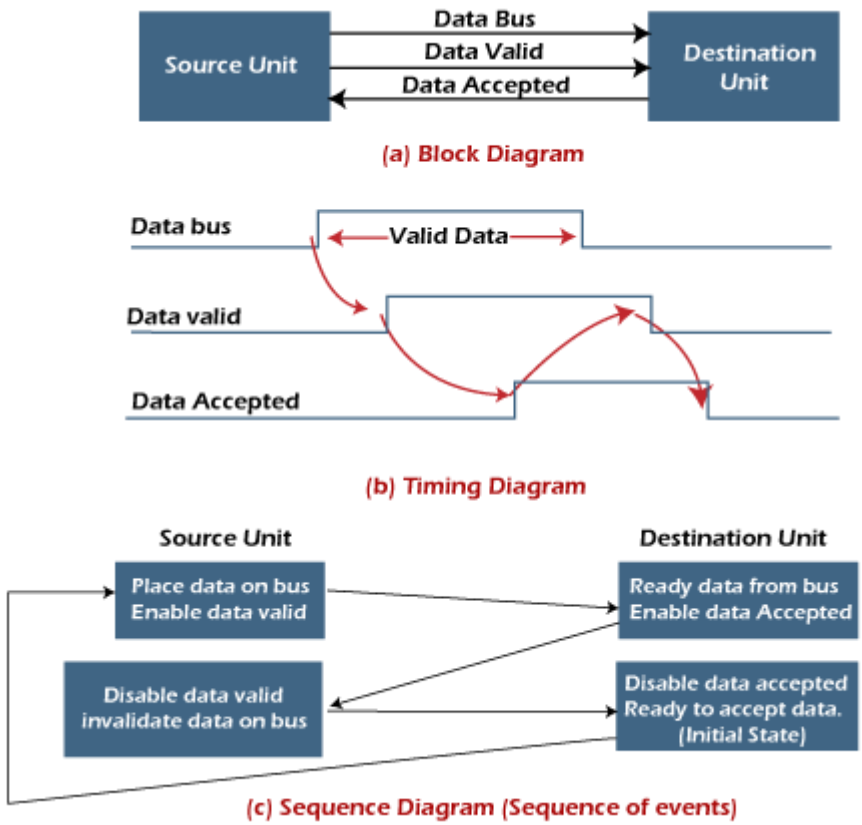
The strobe method has the disadvantage that the source unit that initiates the transfer has no way of knowing whether the destinatio the data that was placed in the bus. Similarly, a destination unit that initiates the transfer has no way of knowing whether the source u data on the bus.

So this problem is solved by the handshaking method. The handshaking method introduces a second control signal line that replay initiates the transfer.

In this method, one control line is in the same direction as the data flow in the bus from the source to the destination. The source inform the destination unit whether there are valid data in the bus.

The other control line is in the other direction from the destination to the source. This is because the destination unit uses it to info whether it can accept data. And in it also, the sequence of control depends on the unit that initiates the transfer. So it means the seque depends on whether the transfer is initiated by source and destination.

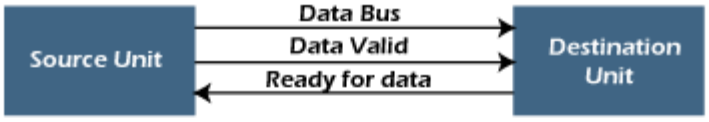
- **Source initiated handshaking:** In the below block diagram, you can see that two handshaking lines are "**data valid**", which is the source unit, and "**data accepted**", generated by the destinati



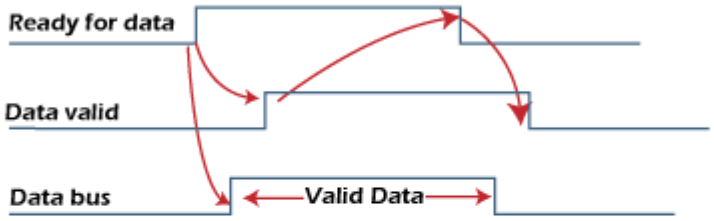
The timing diagram shows the timing relationship of the exchange of signals between the two units. The source initiates a tran data on the bus and enabling its data valid signal. The destination unit then activates the data accepted signal after it accepts the

The source unit then disables its valid data signal, which invalidates the data on
After this, the destination unit disables its data accepted signal, and the system goes into its initial state. The source unit doe
next data item until after the destination unit shows readiness to accept new data by disabling the data ac
This sequence of events described in its sequence diagram, which shows the above sequence in which the system is presen
time.

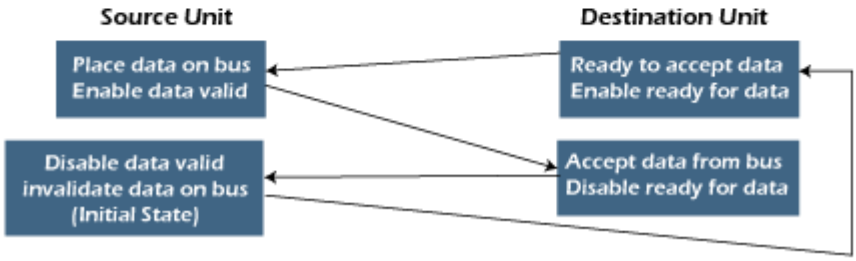
- **Destination initiated handshaking:** In the below block diagram, you see that the two handshaking lines are "**data valid**", ge
source unit, and "**ready for data**" generated by the destinati
Note that the name of signal data accepted generated by the destination unit has been changed to ready for data to reflect its



(a) Block Diagram



(b) Timing Diagram



(c) Sequence Diagram (Sequence of events)

The destination transfer is initiated, so the source unit does not place data on the data bus until it receives a ready data s
destination unit. After that, the handshaking process is the same as that of the sour
The sequence of events is shown in its sequence diagram, and the timing relationship between signals is shown in its ti
Therefore, the sequence of events in both cases would be identical.

Advantages of Asynchronous Data Transfer

Asynchronous Data Transfer in computer organization has the following advantages, such as:

- It is more flexible, and devices can exchange information at their own pace. In addition, individual data characters can compl
so that even if one packet is corrupted, its predecessors and successors will not be affected.
- It does not require complex processes by the receiving device. Furthermore, it means that inconsistency in data transfer does
big crisis since the device can keep up with the data stream. It also makes asynchronous transfers suitable for applications w

data is generated irregularly.

Disadvantages of Asynchronous Data Transfer

There are also some disadvantages of using asynchronous data for transfer in computer organization, such as:

- The success of these transmissions depends on the start bits and their recognition. Unfortunately, this can be easily susc interference, causing these bits to be corrupted or distorted.
- A large portion of the transmitted data is used to control and identify header bits and thus carries no helpful information transmitted data. This invariably means that more data packets need to be sent.