Module 2

TRANSMISSION MODES

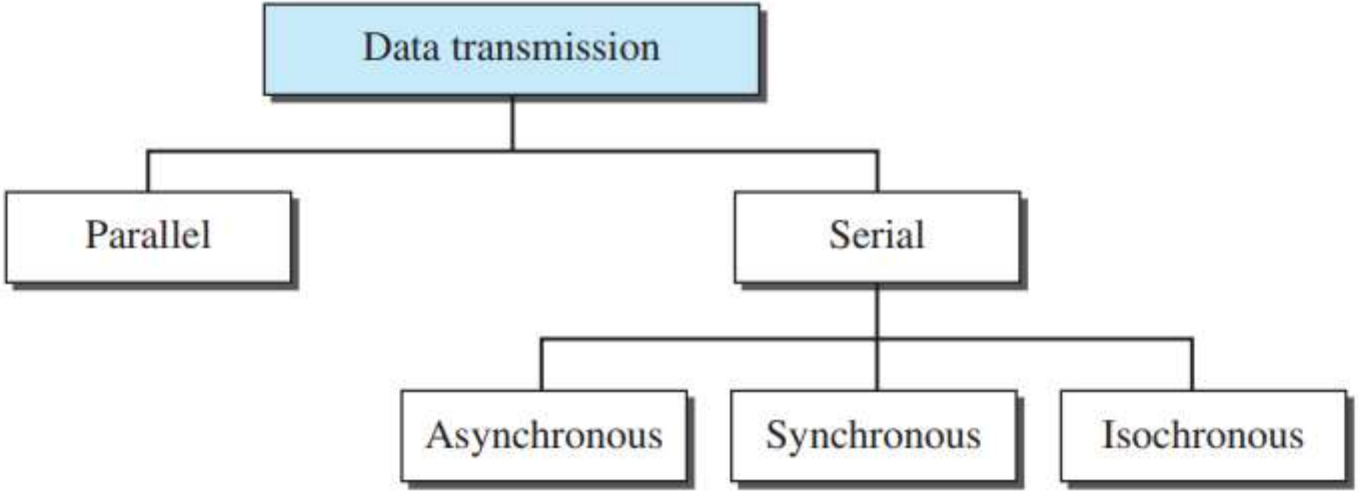
The transmission of binary data across a link can be accomplished in either parallel or serial mode.

In parallel mode, multiple bits are sent with each clock tick.

In serial mode, 1 bit is sent with each clock tick.

While there is only one way to send parallel data.

There are three subclasses of serial transmission: asynchronous, synchronous, and isochronous


Data transmission
Parallel
Serial
Asynchronous
Synchronous
Isochronous

Parallel Transmission

Binary data, consisting of 1s and 0s, may be organized into groups of n bits each.

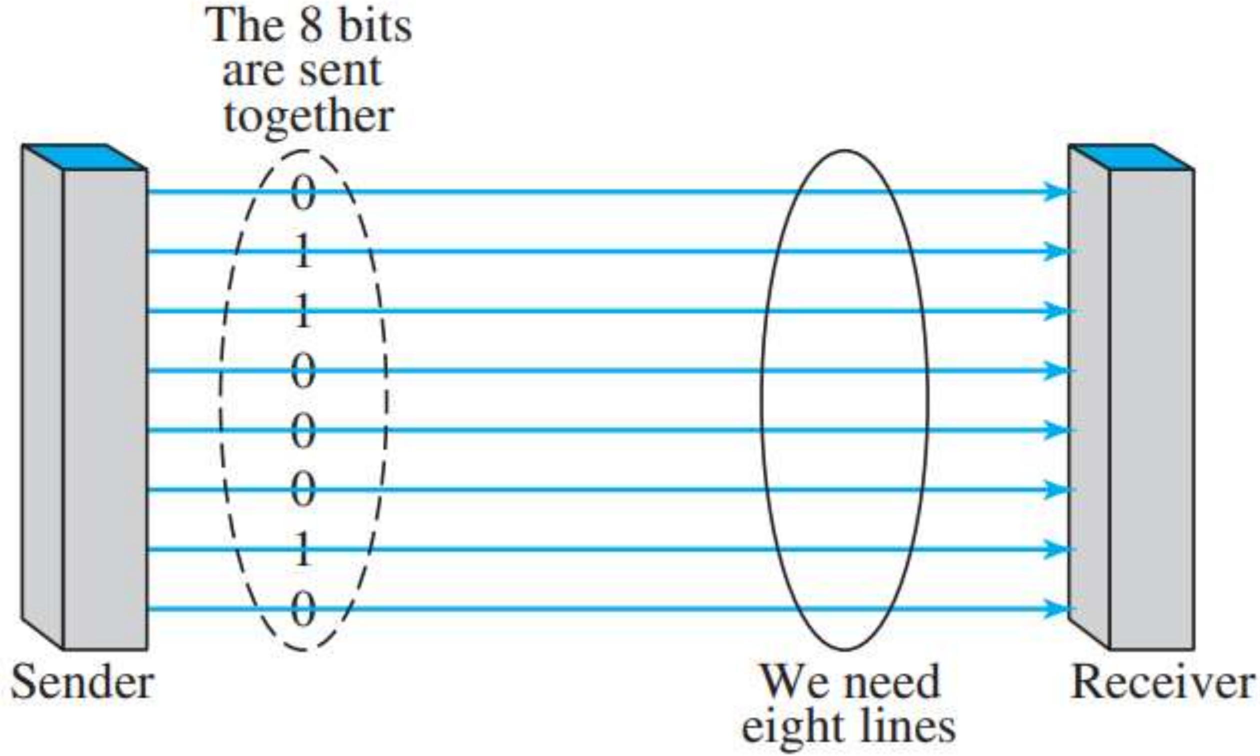
Computers produce and consume data in groups of bits much as we conceive of and use spoken language in the form of words rather than letters.

By grouping, we can send data n bits at a time instead of 1. This is called parallel transmission.

Parallel Transmission

The mechanism for parallel transmission is a conceptually simple one: Use n wires to send n bits at one time.

That way each bit has its own wire, and all n bits of one group can be transmitted with each clock tick from one device to another.


The 8 bits are sent together
0
1
1
0
0
0
1
0
Sender
We need eight lines
Receiver

Parallel Transmission

The advantage of parallel transmission is speed.

All else being equal, parallel transmission can increase the transfer speed by a factor of n over serial transmission.

But there is a significant disadvantage: cost.

Parallel transmission requires n communication lines (wires in the example) just to transmit the data stream.

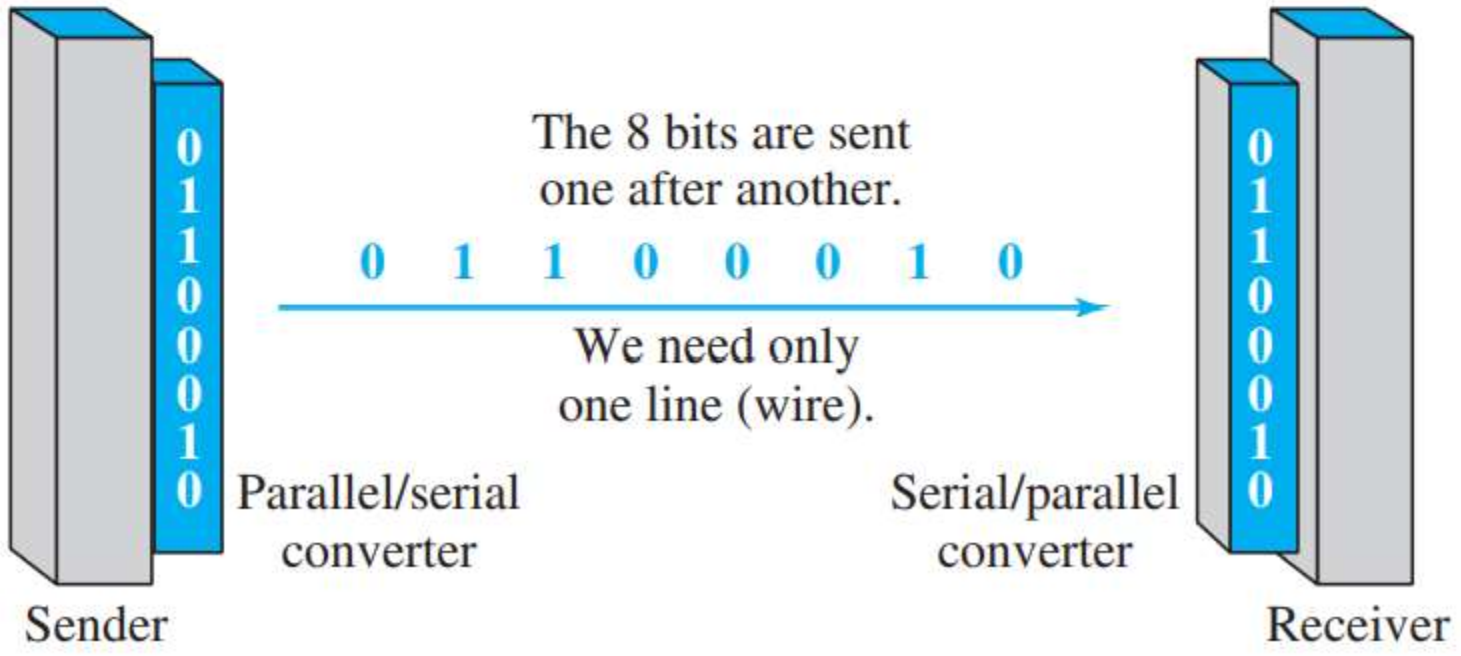
Because this is expensive, parallel transmission is usually limited to short distances.

Serial Transmission

In serial transmission ***one bit follows another.***

So we need ***only one communication channel*** rather than n to transmit data between two communicating devices.

The advantage of serial over parallel transmission is that with only one communication channel, serial transmission reduces the cost of transmission over parallel by roughly a factor of n.



The 8 bits are sent one after another.
0 1 1 0 0 0 1 0
We need only one line (wire).
Parallel/serial converter
Serial/parallel converter
Sender
Receiver
0 1 1 0 0 0 1 0
0 1 1 0 0 0 1 0

Serial Transmission

Since communication within devices is parallel, conversion devices are required at the interface between the sender and the line (parallel-to-serial) and between the line and the receiver (serial-to-parallel).

Serial transmission occurs in one of three ways:

1. Asynchronous
2. Synchronous
3. Isochronous.

Asynchronous Transmission

Asynchronous transmission is so named because the ***timing of a signal is unimportant***.

Instead, ***information is received and translated by agreed upon patterns.***

As long as those patterns are followed, the receiving device can retrieve the information without regard to the rhythm in which it is sent.

Asynchronous Transmission

Patterns are based on grouping the bit stream into bytes.

Each group, usually 8 bits, is sent along the link as a unit.

The sending system handles each group independently, relaying it to the link whenever ready, without regard to a timer.

Without synchronization, the receiver cannot use timing to predict when the next group will arrive.

To alert the receiver to the arrival of a new group, therefore, an extra bit is added to the beginning of each byte.

This bit, usually a 0, is called the start bit.

Asynchronous Transmission

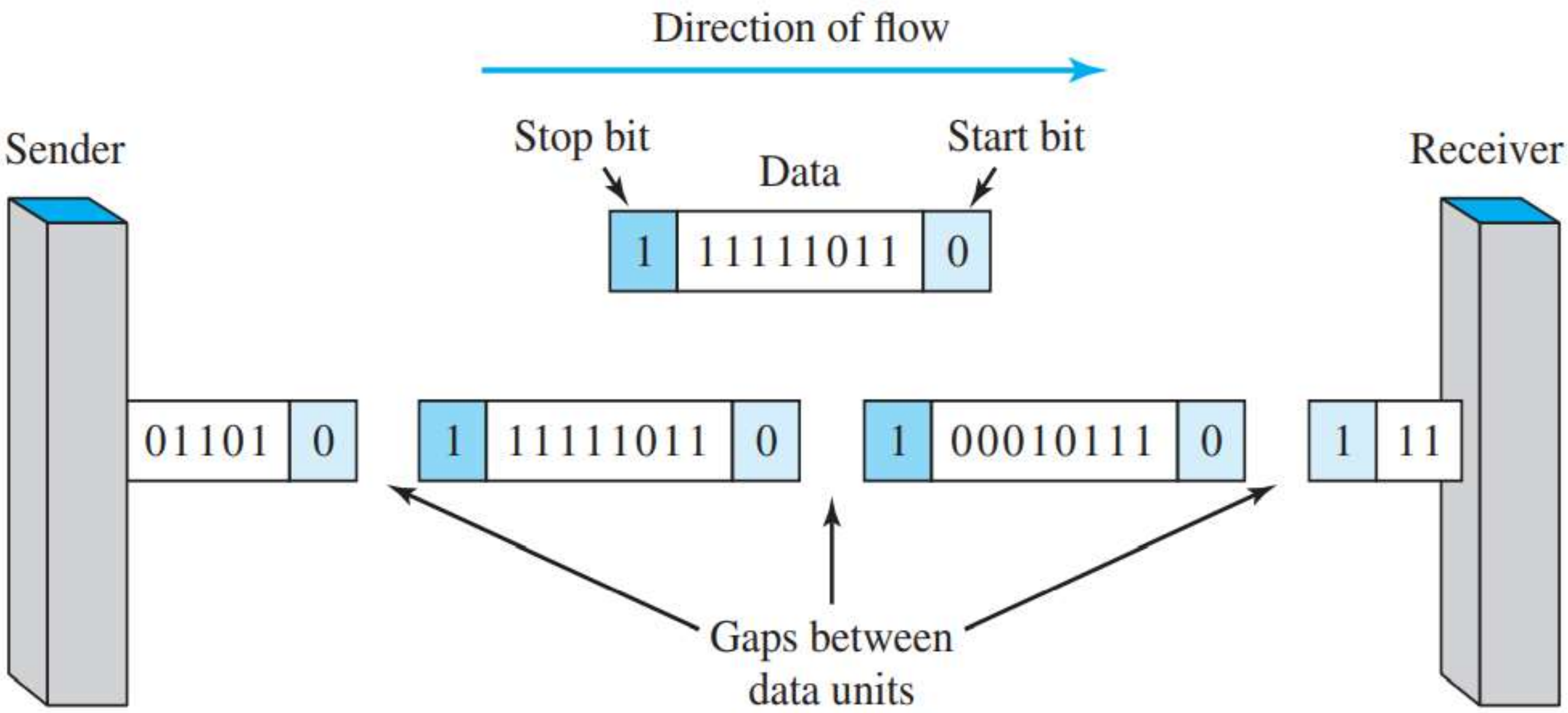
To let the receiver know that the byte is finished, 1 or more additional bits are appended to the end of the byte.

These bits, usually 1s, are called stop bits.

By this method, each byte is increased in size to at least 10 bits, of which 8 bits is information and 2 bits or more are signals to the receiver.

In addition, the transmission of each byte may then be followed by a gap of varying duration. This gap can be represented either by an idle channel or by a stream of additional stop bits.

In asynchronous transmission, we send 1 start bit (0) at the beginning and 1 or more stop bits (1s) at the end of each byte. There may be a gap between bytes.



Direction of flow
Stop bit
Data
Start bit
Sender
Receiver
1
11111011
0
01101
0
1
11111011
0
1
00010111
0
1
11
Gaps between
data units

Synchronous Transmission

In synchronous transmission, the bit stream is combined into ***longer "frames,"*** which may contain multiple bytes.

Each byte, however, is introduced onto the transmission link without a gap between it and the next one.

It is left to the receiver to separate the bit stream into bytes for decoding purposes.

In other words, data are transmitted as an unbroken string of 1s and 0s, and the receiver separates that string into the bytes, or characters, it needs to reconstruct the information.

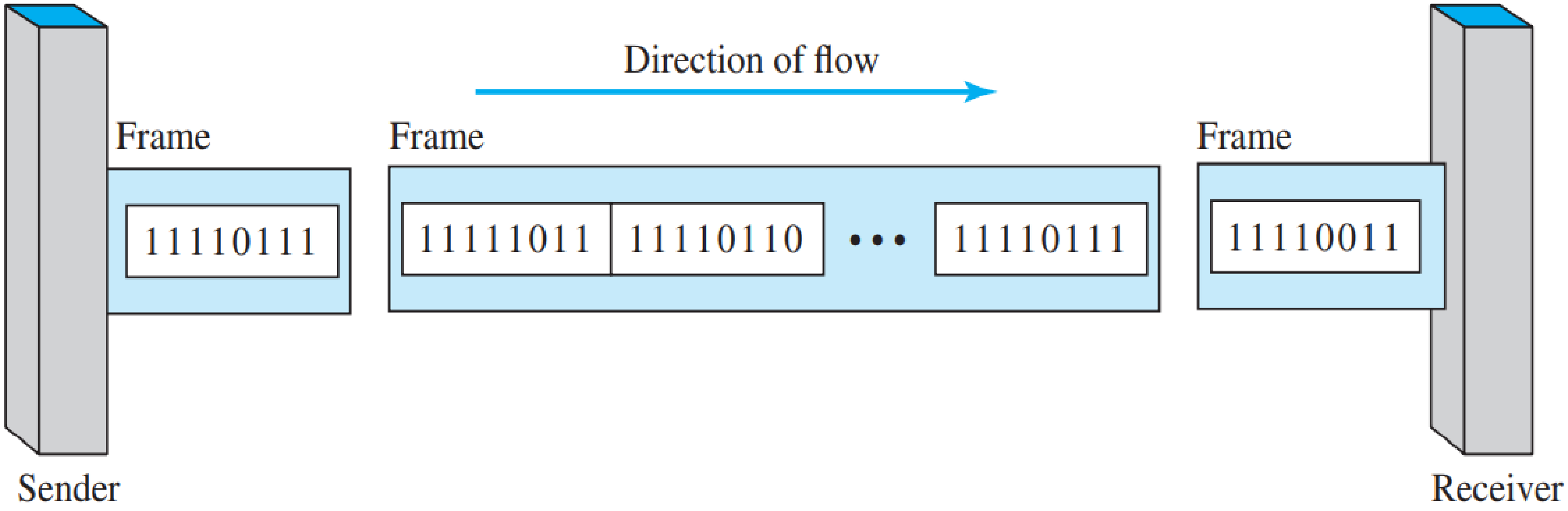
In synchronous transmission, we send bits one after another without start or stop bits or gaps. It is the responsibility of the receiver to group the bits.

Synchronous Transmission

The receiver counts the bits as they arrive and groups them in 8-bit units.

Without gaps and start and stop bits, there is no built-in mechanism to help the receiving device adjust its bit synchronization midstream.

Timing becomes very important, therefore, because the accuracy of the received information is ***completely dependent on the ability of the receiving device to keep an accurate count of the bits as they come in.***


Direction of flow
Frame
11110111
Frame
11111011
11110110
• • •
11110111
Frame
11110011
Sender
Receiver

Synchronous Transmission

The advantage of synchronous transmission is speed.

With no extra bits or gaps to introduce at the sending end and remove at the receiving end, and, by extension, with fewer bits to move across the link, synchronous transmission is faster than asynchronous transmission.

For this reason, it is more useful for high-speed applications such as the transmission of data from one computer to another.

Byte synchronization is accomplished in the data-link layer.

Isochronous Transmission

In real-time audio and video, in which uneven delays between frames are not acceptable, synchronous transmission fails.

For example, TV images are broadcast at the rate of 30 images per second; they must be viewed at the same rate.

If each image is sent by using one or more frames, there should be no delays between frames. ***For this type of application, synchronization between characters is not enough; the entire stream of bits must be synchronized.***

The isochronous transmission guarantees that the data arrive at a fixed rate.

Multiplexing

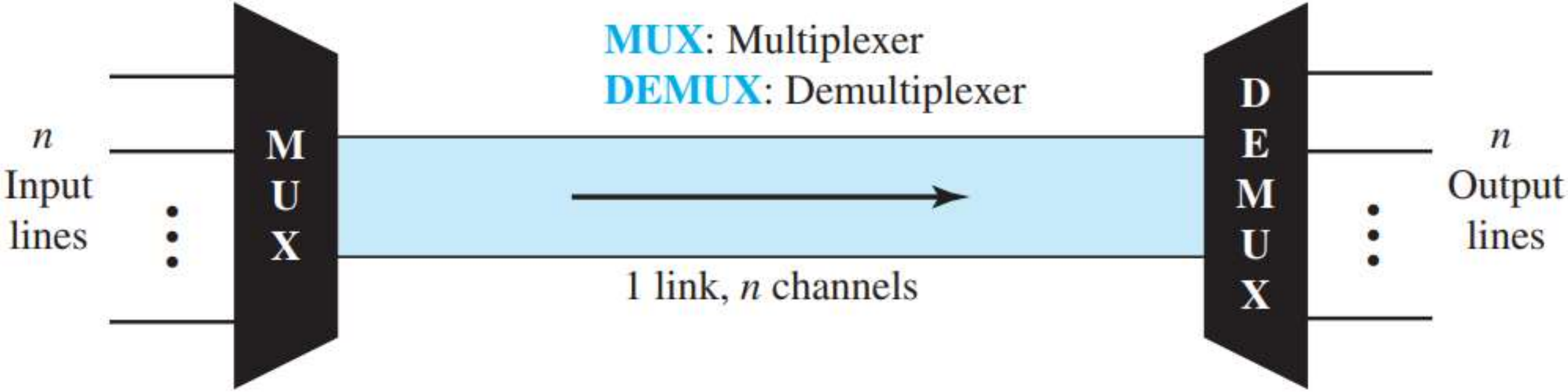
Whenever the bandwidth of a medium linking two devices is greater than the bandwidth needs of the devices, the link can be shared.

Multiplexing is the set of techniques that allow the simultaneous transmission of multiple signals across a single data link.

If the bandwidth of a link is greater than the bandwidth needs of the devices connected to it, the bandwidth is wasted.

An efficient system maximizes the utilization of all resources; bandwidth is one of the most precious resources we have in data communications.

Multiplexing

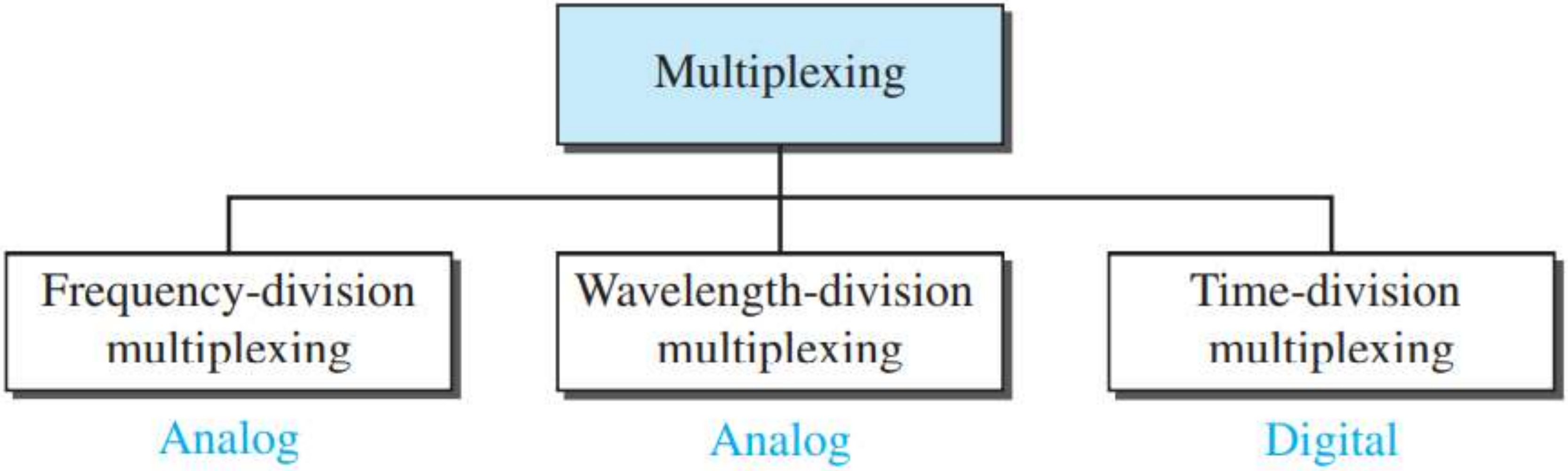


Multiplexing

There are three basic multiplexing techniques: ***frequency-division multiplexing, wavelength-division multiplexing, and time-division multiplexing.***

The first two are techniques designed for analog signals, the third, for digital signals

Multiplexing



Frequency-Division Multiplexing

Frequency-division multiplexing (FDM) is an analog technique that can be applied when the bandwidth of a link (in hertz) is greater than the combined bandwidths of the signals to be transmitted.

In FDM, signals generated by each sending device modulate different carrier frequencies.

These modulated signals are then combined into a single composite signal that can be transported by the link.

Frequency-Division Multiplexing

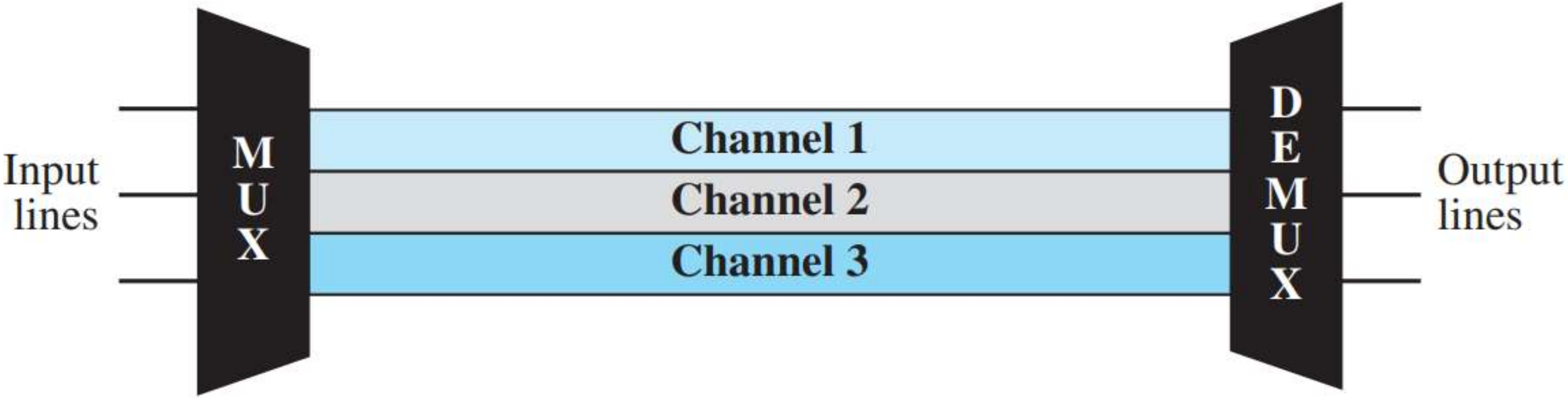
Carrier frequencies are separated by sufficient bandwidth to accommodate the modulated signal.

These bandwidth ranges are the channels through which the various signals travel.

Channels can be separated by strips of unused bandwidth—***guard bands***—to prevent signals from overlapping.

In addition, carrier frequencies must not interfere with the original data frequencies.

Frequency-Division Multiplexing



Frequency-Division Multiplexing

Carrier frequencies are separated by sufficient bandwidth to accommodate the modulated signal.

These bandwidth ranges are the channels through which the various signals travel.

Channels can be separated by strips of unused bandwidth—***guard bands***—to prevent signals from overlapping.

In addition, carrier frequencies must not interfere with the original data frequencies.