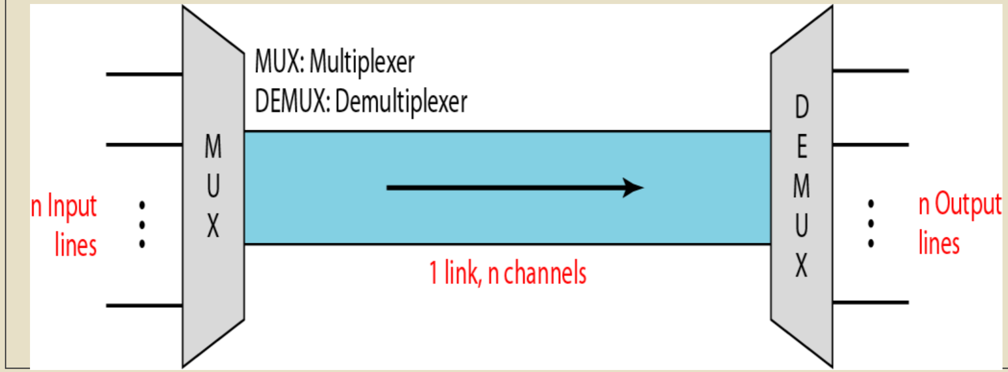
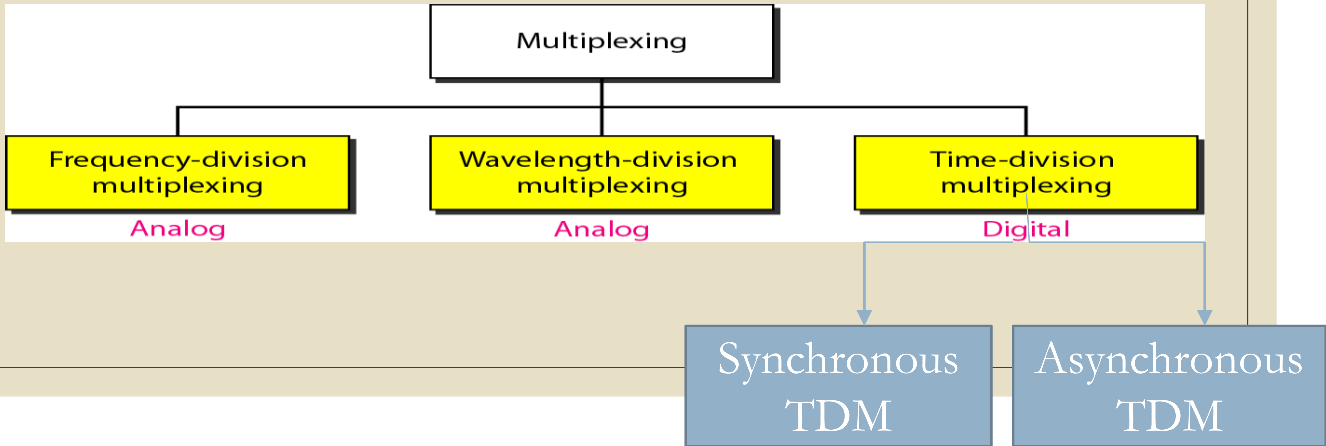
MULTIPLEXING & SPREADING

Annu James

MULTIPLEXING

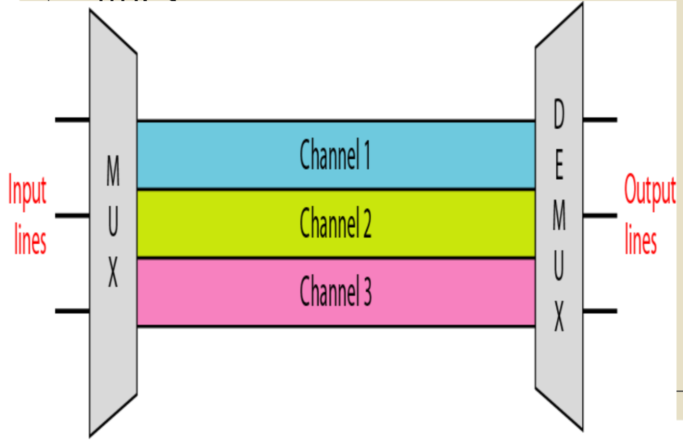
- Bandwidth utilization is the wise use of available bandwidth to achieve specific goals- Efficiency can be achieved by multiplexing; privacy and anti-jamming can be achieved by spreading.
- *Whenever the bandwidth of a medium linking two devices is greater than the bandwidth needs of the devices, the link can be shared. Multiplexing is the set of techniques that allows the simultaneous transmission of multiple signals across a single data link. As data and telecommunications use increases, so does traffic.*
- In a multiplexed system, *n* lines share the bandwidth of one link. Figure shows the basic format of a multiplexed system. The lines on the left direct their transmission streams to a multiplexer (MUX), which combines them into a single stream (many-to- one). At the receiving end, that stream is fed into a demultiplexer (DEMUX), which separates the stream back into its component transmissions (one-to-many) and directs them to their corresponding lines. In the figure, the word **link** refers to the **physical path**. The word **channel** refers to **the portion of a link** that carries a transmission between a given pair of lines. One link can have many *(n)* channels.
- 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

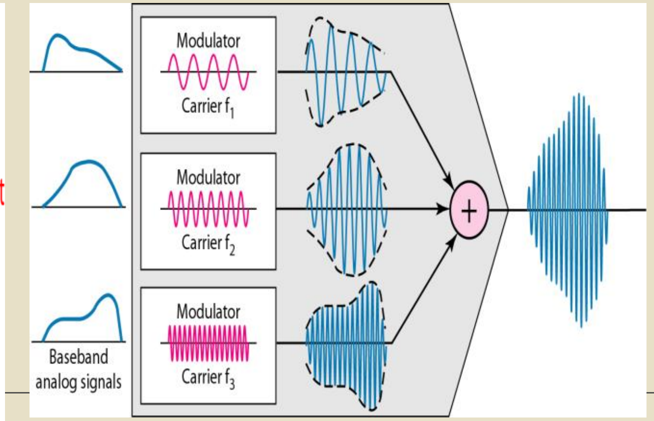


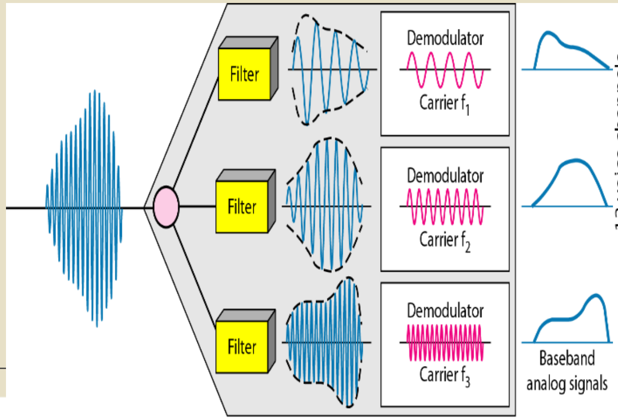


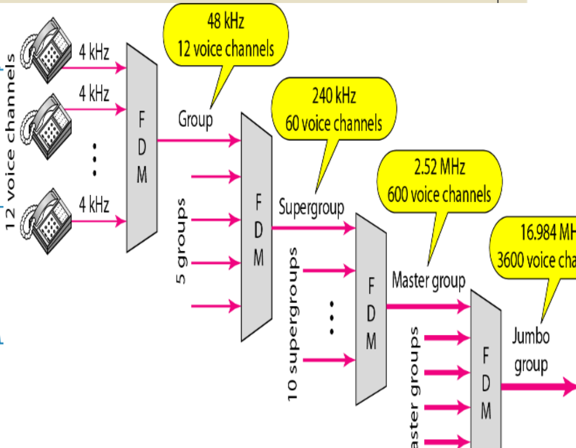
Frequency-Division Multiplexing

- 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. 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.
- Each source generates a signal of a similar frequency range. Inside the **multiplexer**, these similar signals modulates different carrier frequencies f1,f2,f3 The resulting modulated signals are then combined into a single composite signal that is sent out over a media link that has enough bandwidth to accommodate it.ie, The frequency spectrum is divided into frequency bands with each user having exclusive possession of some band. Eg AM radio 500 to 1500 HZ.
- **The demultiplexer** uses a series of filters to decompose the multiplexed signal into its constituent component signals. The individual signals are then passed to a demodulator that separates them from their carriers and passes them to the output lines.



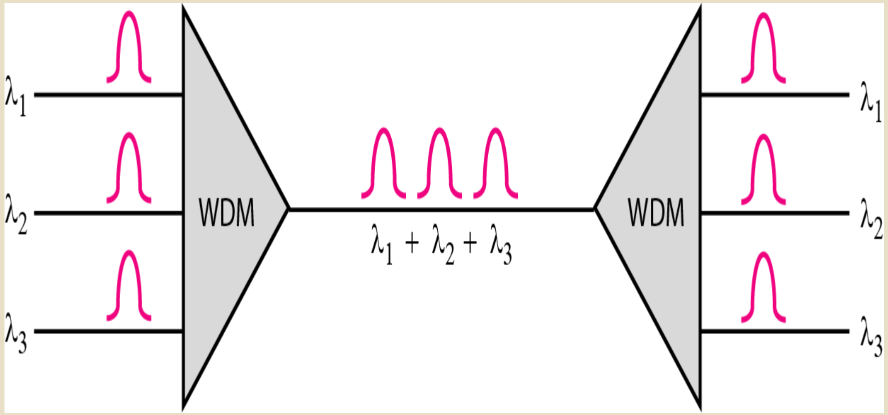


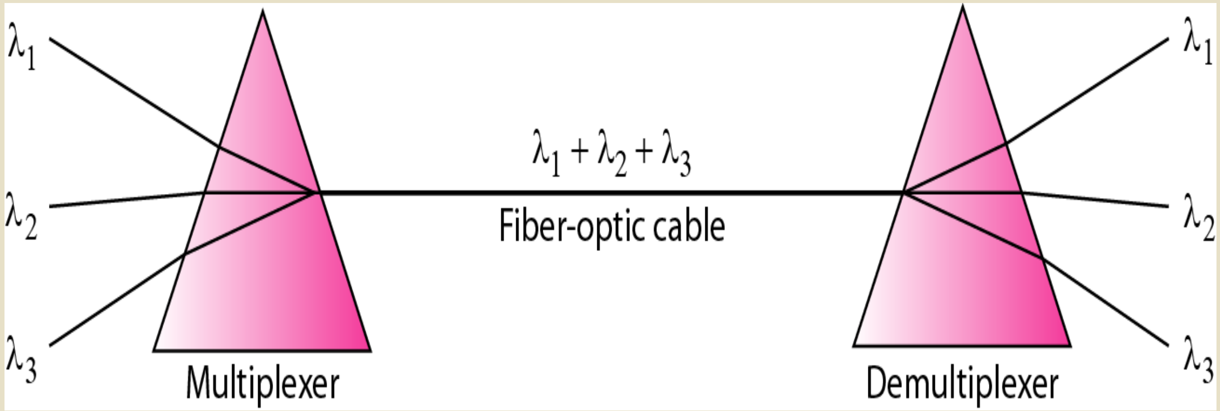




Wavelength-division multiplexing (WDM)

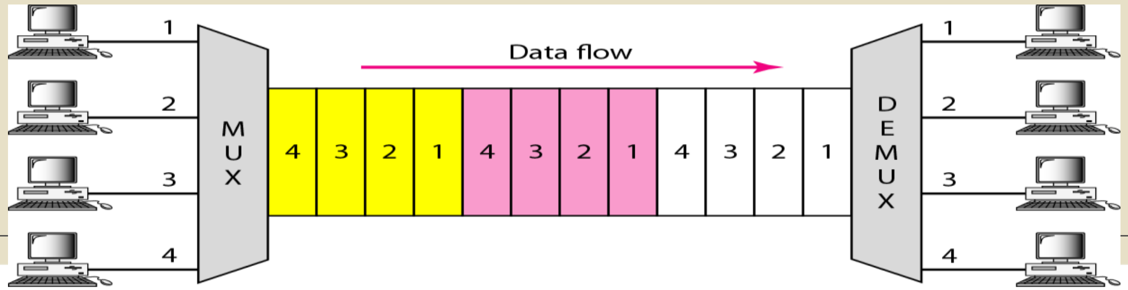
- WDM is conceptually the same as FDM, except that the multiplexing and demultiplexing involve optical signals transmitted through fiber-optic channels using prism.. The idea is the same: We are combining different signals of different frequencies. The difference is that the high frequency optical signals.
- WDM is designed to use the high-data-rate capability of fiber-optic cable. The optical fiber data rate is higher than the data rate of metallic transmission cable. Using a fiber-optic cable for one single line wastes the available bandwidth. Multiplexing allows us to combine several lines into one.
- Although WDM technology is very complex, the basic idea is very simple. We want to combine multiple light sources into one single light at the multiplexer and do the reverse at the demultiplexer. The combining and splitting of light sources are easily handled by a prism. Recall from basic physics that a prism bends a beam of light based on the angle of incidence and the frequency. Using this technique, a multiplexer can be made to combine several input beams of light, each containing a narrow band of frequencies, into one output beam of a wider band of frequencies. A demultiplexer can also be made to reverse the process

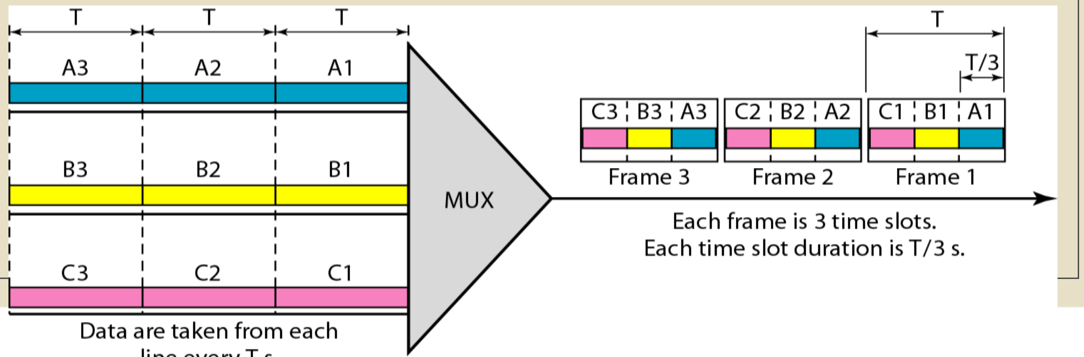




Time Division Multiplexing

- **Synchronous Time-Division Multiplexing**
- Time-division multiplexing (TDM) is a digital process that allows several connections to share the high bandwidth of a sharing a portion of the bandwidth as in FDM, time is shared. Each connection occupies a portion of time in the link. The data rate capacity of transmission medium is greater than sender and receiver. Note that the same link is used as in FDM; here, however, the link is shown sectioned by time rather than by frequency. In the figure, portions of signals 1,2,3, and 4 occupy the link sequentially.
- We also need to remember that TDM is, in principle, a digital multiplexing technique. Digital data from different sources are combined into one timeshared link. TDM is a digital multiplexing technique for combining several low-rate channels into one high-rate one.
- We can divide TDM into two different schemes: **synchronous and statistical**. We first discuss synchronous TDM and then show how statistical TDM differs. In synchronous TDM, each input connection has an allotment in the output even if it is not sending data.
- ***Time Slots and Frames***
- In synchronous TDM, the data flow of each input connection is divided into units, where each input occupies one input time slot. A unit can be 1 bit, one character, or one block of data. Each input unit becomes one output unit and occupies one output time slot. However, the duration of an output time slot is *n* times shorter than the duration of an input time slot. If an input time slot is *T* s, the output time slot is *Tin* s, where *n* is the number of connections.

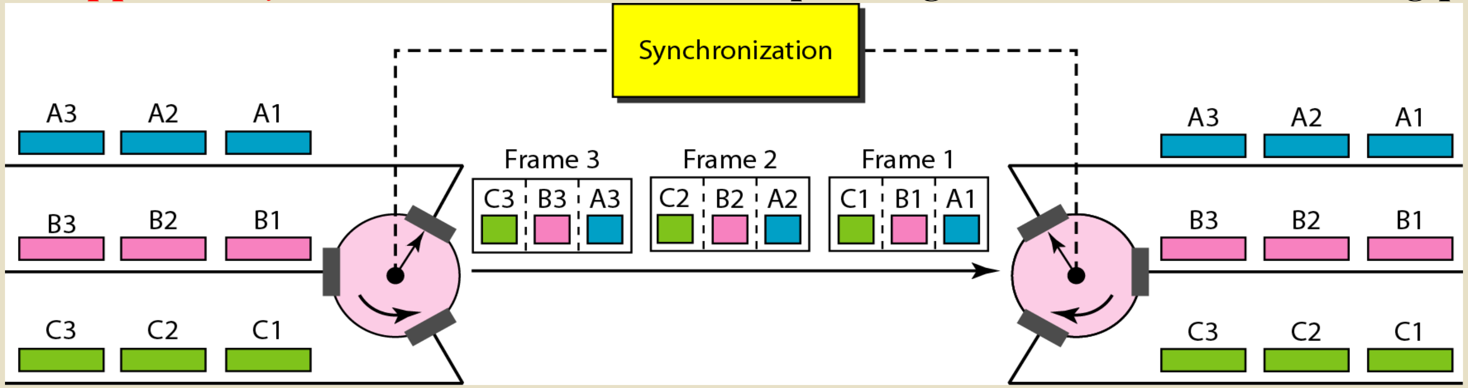


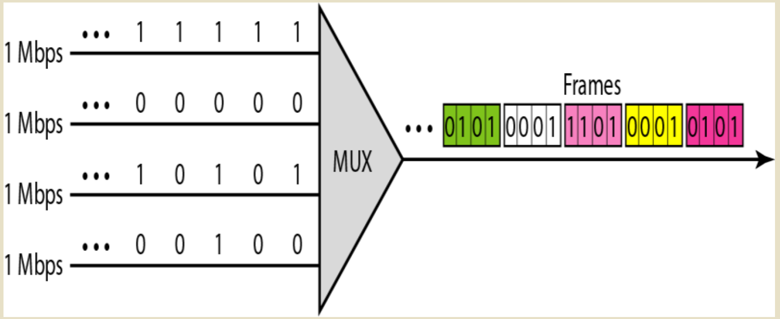


- In synchronous TDM, a round of data units from each input connection is collected into a frame. If we have *n* connections, a frame is divided into *n* time slots and one slot is allocated for each unit, one for each input line. If the duration of the input unit is *T*, the duration of each slot is *Tin* and the duration of each frame is *T*
- In synchronous TDM, the data rate of the link is *n* times faster, and the unit duration is *n* times shorter.
- Time slots are grouped into frames. A frame consists of one complete cycle of time slots, with one slot dedicated to each sending device. In a system with *n* input lines, each frame has *n* slots, with each slot allocated to carrying data from a specific input line.

Interleaving

- TDM can be visualized as two fast-rotating switches, one on the multiplexing side and the other on the demultiplexing side. The switches are synchronized and rotate at the same speed, but in opposite directions. On the multiplexing side, as the switch opens in front of a connection, that connection has the opportunity to send a unit onto the path. This process is called **interleaving.** On the demultiplexing side, as the switch opens in front of a connection, that connection has the opportunity to receive a unit from the path.Figure shows the interleaving process for the connection.

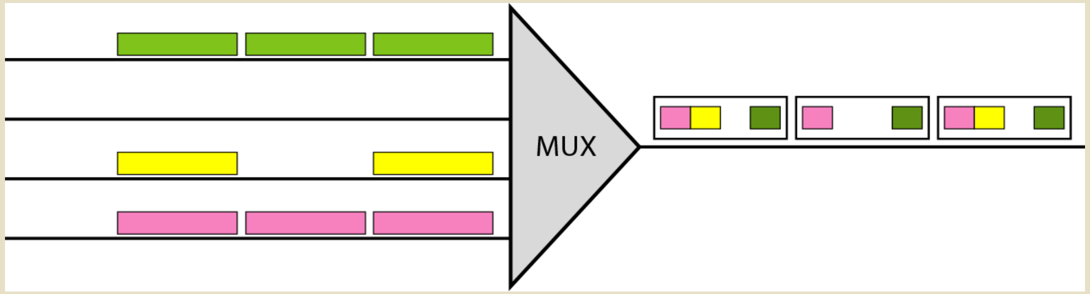


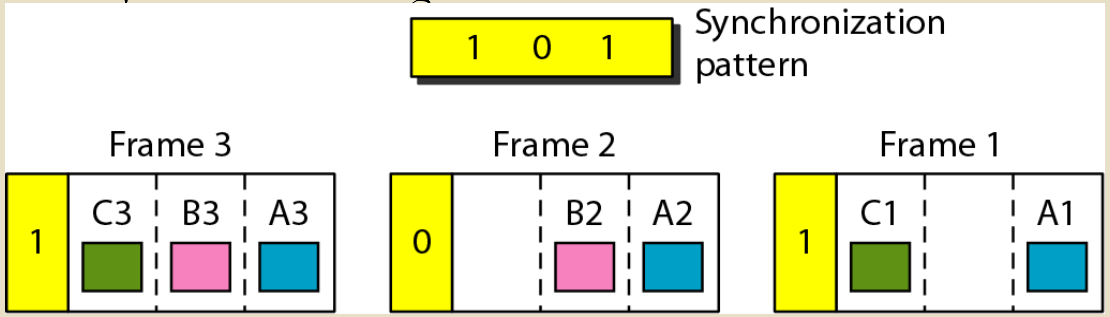


Empty Slot- Synchronous TDM is not as efficient as it could be. If a source does not have data to send, the corresponding slot in the output frame is empty. Figure shows a case in which one of the input lines has no data to send and one slot in another input line has discontinuous data.The first output frame has three slots filled, the second frame has two slots filled, and the third frame has three slots filled. No frame is full. We learn in the next section that statistical TDM can improve the efficiency by removing the empty slots from the frame.

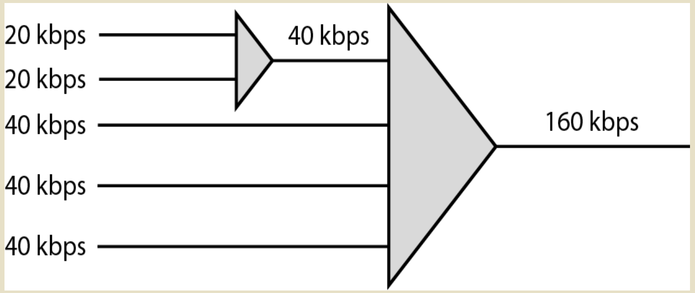
- ***Data Rate Management-*** One problem with TDM is how to handle a disparity in the input data rates. In all our discussion so far, we assumed that the data rates of all input lines were the same. However, If data rates are not the same, three strategies, or a combination of them, can be used. We call these three strategies **multilevel multiplexing, multiple-slot allocation,** and **pulse stuffing.**

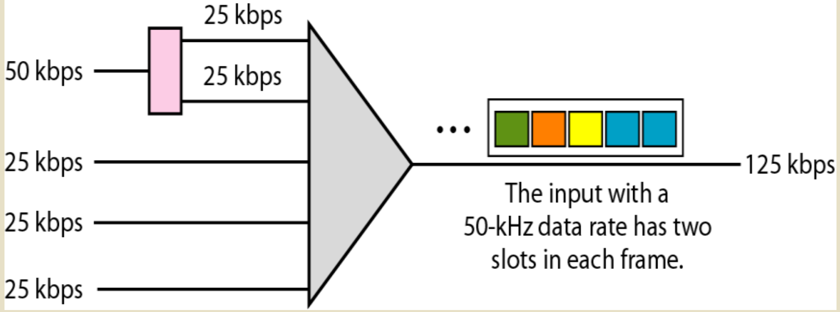
- **Frame Synchronization-** The implementation of TDM is not as simple as that of FDM. Synchronization between the multiplexer and demultiplexer is a major issue. If the. multiplexer and the demultiplexer are not synchronized, a bit belonging to one channel may be received by the wrong channel. For this reason, one or more synchronization bits are usually added to the beginning of each frame. These bits, called framing bits, follow a pattern, frame to frame, that allows the demultiplexer to synchronize with the incoming stream so that it can separate the time slots accurately. In most cases, this synchronization information consists of 1 bit per frame, alternating between 0 and I, as shown in Figure

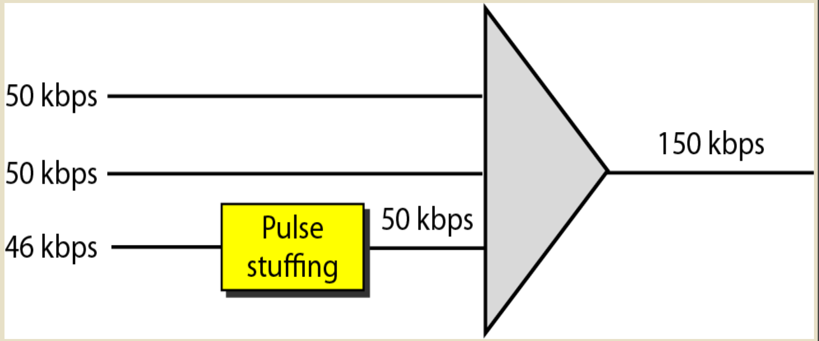




- **Multilevel Multiplexing-** Multilevel multiplexing is a technique used when the data rate of an input line is a multiple of others. For example, in Figure, we have two inputs of 20 kbps and three inputs of 40 kbps. The first two input lines can be multiplexed together to provide a data rate equal to the last three. A second level of multiplexing can create an output of 160 kbps.
- **Multiple-Slot Allocation** Sometimes it is more efficient to allot more than one slot in a frame to a single input line. For example, we might have an input line that has a data rate that is a multiple of another input. In Figure, the input line with a SO-kbps data rate can be given two slots in the output. We insert a serial-to-parallel converter in the line to make two inputs out of one.
- **Pulse Stuffing** Sometimes the bit rates of sources are not multiple integers of each other. Therefore, neither of the above two techniques can be applied. One solution is to make the highest input data rate the dominant data rate and then add dummy bits to the input lines with lower rates. This will increase their rates. This technique is called pulse stuffing, bit padding, or bit stuffing. The idea is shown in Figure . The input with a data rate of 46 is pulse-stuffed to increase the rate to 50 kbps. Now multiplexing can take place.

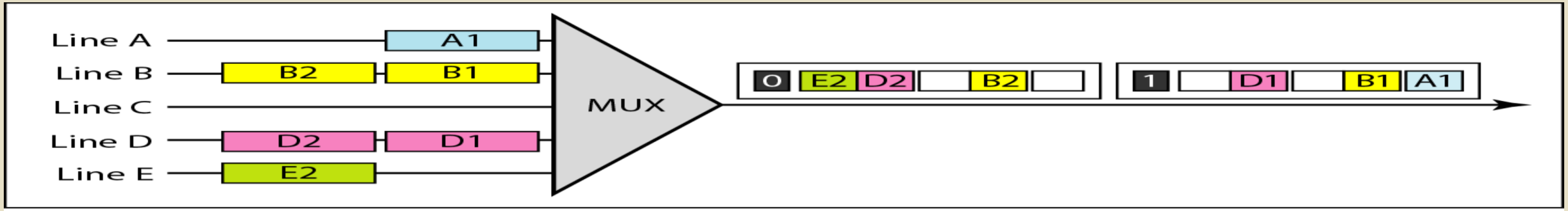




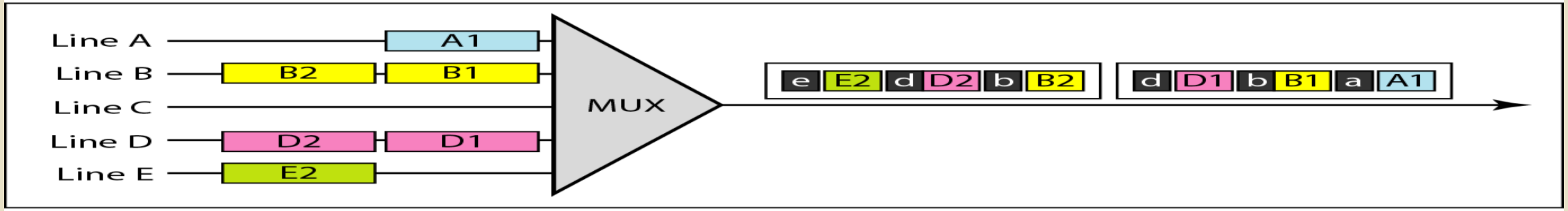


- **Statistical /Asynchronous Time-Division Multiplexing**
- As we saw in the previous section, in synchronous TDM, each input has a reserved slot in the output frame. This can be inefficient if some input lines have no data to send. In statistical time-division multiplexing, slots are dynamically allocated to improve bandwidth efficiency. Only when an input line has a slot's worth of data to send is it given a slot in the output frame. In statistical multiplexing, the number of slots in each frame is less than the number of input lines. The multiplexer checks each input line in round robin fashion; it allocates a slot for an input line if the line has data to send; otherwise, it skips the line and checks the next line.
- Figure shows a synchronous and a statistical TDM example. In the former, some slots are empty because the corresponding line does not have data to send. In the latter, however, no slot is left empty as long as there are data to be sent by any input line.
- ***Addressing***
- Figure also shows a major difference between slots in synchronous TDM and statistical TDM. An output slot in synchronous TDM is totally occupied by data; in statistical TDM, a slot needs to carry data as well as the address of the destination. In synchronous TDM, there is no need for addressing; synchronization and pre assigned relationships between the inputs and outputs serve as an address. We know, for example, that input 1 always goes to input 2. If the multiplexer and the demultiplexer are synchronized, this is guaranteed. In statistical multiplexing, there is no fixed relation ship between the inputs and outputs because there are no pre assigned or reserved slots. We need to include the address of the receiver inside each slot to show where it is to be delivered.

- ***Slot Size***- Since a slot carries both data and an address in statistical TDM, the ratio of the data size to address size must be reasonable to make transmission efficient. For example, it would be inefficient to send 1 bit per slot as data when the address is 3 bits. This would mean an overhead of 300 percent. In statistical TDM, a block of data is usually many bytes while the address is just a few bytes.
- ***No Synchronization Bit*** -There is another difference between synchronous and statistical TDM, but this time it is at the frame level. The frames in statistical TDM need not be synchronized, so we do not need synchronization bits.
- ***Bandwidth***- In statistical TDM, the capacity of the link is normally less than the sum of the capacities of each channel. The designers of statistical TDM define the capacity of the link based on the statistics of the load for each channel. If on average only *x* percent of the input slots are filled, the capacity of the link reflects this. Of course, during peak times, some slots need to wait.



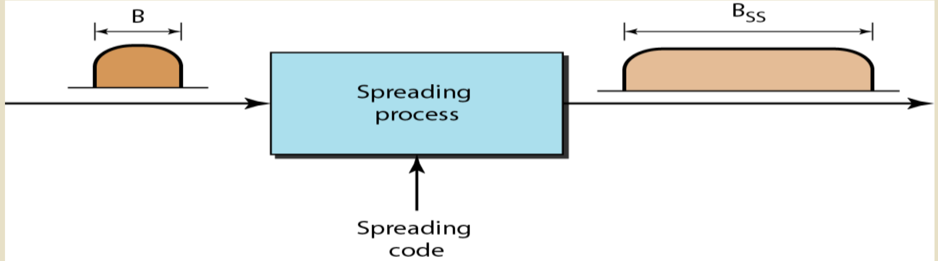
a. Synchronous TDM



b. Statistical TDM

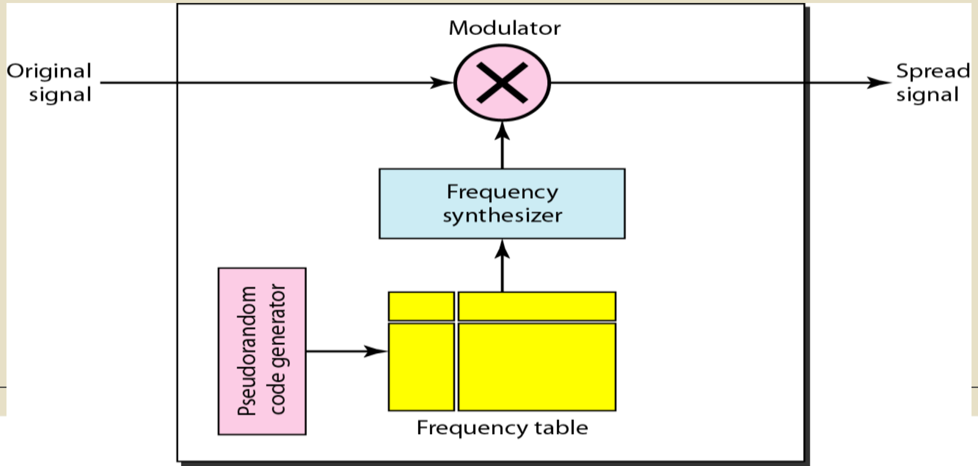
SPREAD SPECTRUM

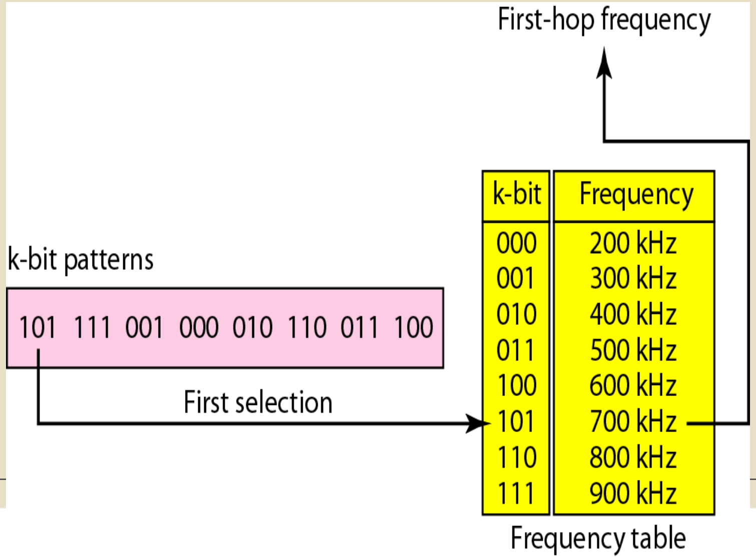
- Multiplexing combines signals from several sources to achieve bandwidth efficiency; the available bandwidth of a link is divided between the sources. In spread spectrum (ss), we also combine signals from different sources to fit into a larger bandwidth, but our goals are somewhat different. Spread spectrum is designed to used in wireless applications (LANs and WANs). In these types of applications, we have some concerns that outweigh bandwidth efficiency. In wireless applications, all stations use air (or a vacuum) as the medium for communication. Stations must be able to share this medium without interception by an eavesdropper and without being subject to jamming from a malicious intruder
- To achieve these goals, spread spectrum techniques add redundancy; they spread the original spectrum needed for each station. If the required bandwidth for each station is *B*, spread spectrum expands it to Bss' such that Bss » *B*. The expanded bandwidth allows the source to wrap its message in a protective envelope for a more secure transmission. An analogy is the sending of a delicate, expensive gift. We can insert the gift in a special box to prevent it from being damaged during transportation, and we can use a superior delivery service to guarantee the safety of the package.
- Figure shows the idea of spread spectrum. Spread spectrum achieves its goals through two principles:
  - The bandwidth allocated to each station needs to be, by far, larger than what is needed. This allows redundancy.
  - The expanding of the original bandwidth *B* to the bandwidth Bss must be done by a process that is independent of the original signal. In other words, the spreading process occurs after the signal is created by the source.
- After the signal is created by the source, the spreading process uses a spreading code and spreads the bandwidth. The figure shows the original bandwidth *B* and the spreaded bandwidth *Bss*. The spreading code is a series of numbers that look random, but are actually a pattern. There are two techniques to spread the bandwidth: frequency hopping spread spectrum (FHSS) and direct sequence spread spectrum (DSSS).



Frequency Hopping Spread Spectrum (FHSS)

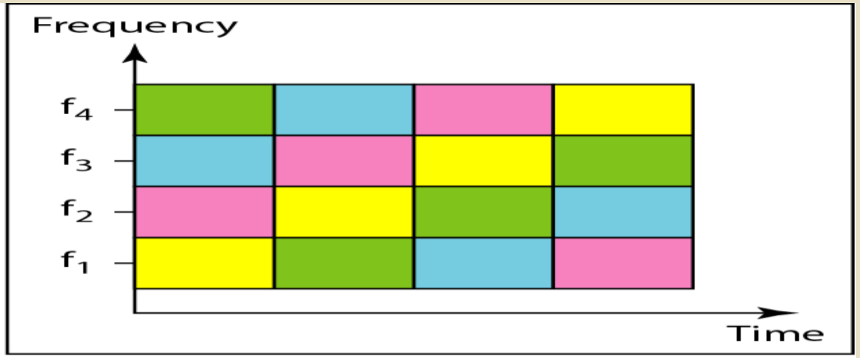
- The frequency hopping spread spectrum (FHSS) technique uses *M* different carrier frequencies that are modulated by the source signal. At one moment, the signal modulates one carrier frequency; at the next moment, the signal modulates another carrier
- frequency. Although the modulation is done using one carrier frequency at a time, *M* frequencies are used in the long run. The bandwidth occupied by a source after spreading is *BpHSS »B*. Figure shows the general layout for FHSS. A pseudorandom code generator, called pseudorandom noise (PN), creates a k-bit pattern for every hopping period
- The frequency table uses the pattern to find the frequency to be used for this hopping period and passes it to the frequency synthesizer. The frequency synthesizer creates a carrier signal of that frequency, and the source signal modulates the carrier signal.
- Suppose we have decided to have eight hopping frequencies. This is extremely low for real applications and is just for illustration. In this case, *Mis* 8 and *k* is 3. The pseudo random code generator will create eight different 3-bit patterns. These are mapped to eight different frequencies in the frequency table
- The pattern for this station is 101, 111, 001, 000, 010, all, 100. Note that the pattern is pseudorandom it is repeated after eight hoppings. This means that at hopping period 1, the pattern is 101. The frequency selected is 700 kHz; the source signal modulates this carrier frequency. The second k-bit pattern selected is 111, which selects the 900-kHz carrier; the eighth pattern is 100, the frequency is 600 kHz. After eight hoppings, the pattern repeats, starting from 101 again. Figure 6.30 shows how the signal hops around from carrier to carrier. We assume the required bandwidth of the original signal is 100 kHz.





- It can be shown that this scheme can accomplish the previously mentioned goals. If there are many k-bit patterns and the hopping period is short, a sender and receiver can have privacy. If an intruder tries to intercept the transmitted signal, she can only access a small piece of data because she does not know the spreading sequence to quickly adapt herself to the next hop. The scheme has also an anti jamming effect. A malicious sender may be able to send noise to jam the signal for one hopping period (randomly), but not for the whole period.
- *Bandwidth Sharing*
- If tpossible because a station uses just one frequency in each hopping periodhe number of hopping frequencies is *M,* we can multiplex *M* channels into one by using the same Bss bandwidth. This is; *M* - 1 other frequencies can be used by other *M* - 1 stations. In other words, *M* different stations can use the same Bss if an appropriate modulation technique such as multiple FSK (MFSK) is used. FHSS is similar to FDM, as shown in Figure .a Figure b shows an example of four channels using FDM and four channels using FHSS. In FDM, each station uses *11M* of the bandwidth, but the allocation is fixed; in FHSS, each station uses *11M* of the bandwidth, but the allocation changes hop to hop.

a. FDM



b. FHSS

- **Direct Sequence Spread Spectrum**
- The direct sequence spread spectrum (DSSS) technique also expands the bandwidth of the original signal, but the process is different. In DSSS, we replace each data bit with bits using a spreading code. In other words, each bit is assigned a code of bits, called chips, where the chip rate is times that of the data bit. Figure shows the concept of DSSS.
- As an example, let us consider the sequence used in a wireless LAN, the famous Barker sequence where is 11. We assume that the original signal and the chips in the chip generator use polar NRZ encoding. Figure shows the chips and the result of multiplying the original data by the chips to get the spread signal.
- In Figure, the spreading code is 11 chips having the pattern 10110111000 (in this case). If the original signal rate is *N*, the rate of the spread signal is *11N*. This means that the required bandwidth for the spread signal is 11 times larger than the bandwidth of the original signal. The spread signal can provide privacy if the intruder does not know the code. It can also provide immunity against interference if each station uses a different code.