USN :

CMR Institute of Technology, Bangalore
DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING
II - INTERNAL ASSESSMENT

Semester: 4-CBCS 2018
Subject: DATA COMMUNICATION (18CS46)
Faculty: Ms Dhanya Viswanath

Date: 24 Jun 2021
Time: 01:00 PM - 02:30 PM
Max Marks: 50

Instructions to Students :

Answer any 5 FULL questions

Answer any 5 question(s)

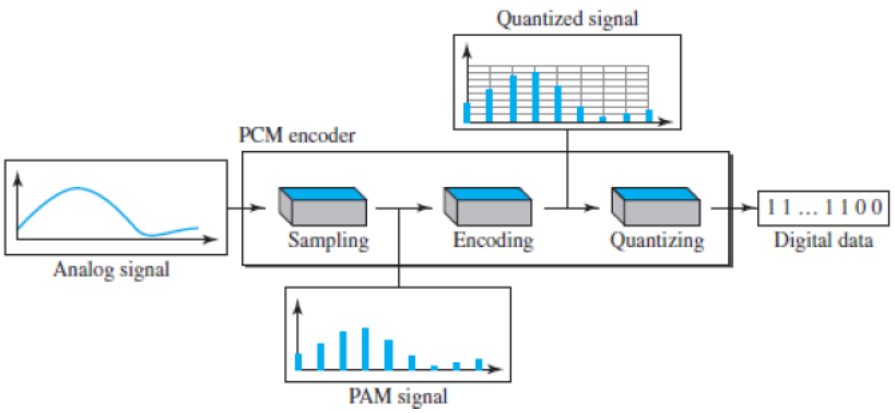
Q.No			Marks	CO	PO	BT/CL
1		Explain PCM encoder used for analog to digital conversion with steps and example.	10	CO2	PO1,PO4	L2
2	a	Briefly explain ASK & FSK modulation techniques with diagrams and specify the bandwidth requirements.	8	CO2	PO1,PO4	L2
	b	Find the bandwidth for a signal transmitting at 12 Mbps for QPSK. The value of d=0.	2	CO2	PO1,PO4	L3
3	a	State and explain the data rate management to handle disparity in input data rates in TDM.	6	CO2	PO1,PO4	L1
	b	Two channels, one with a bit rate of 100 kbps and another with a bit rate of 200 kbps, are to be multiplexed. How can this be achieved? Calculate the frame rate, frame duration and bit rate of the link.	4	CO2	PO1,PO4	L3
4	a	What is spread spectrum? Mention its advantages.	4	CO2	PO1,PO4	L1
	b	Explain FHSS and discuss its merits over FDM.	6	CO2	PO1,PO4	L2
5		Explain the 3 phases of switching at the data link layer with necessary diagrams. Also obtain an expression for total delay.	10	CO2	PO1,PO4	L2
6	a	Explain simple parity check code with a neat diagram.	5	CO2	PO1,PO4	L2
	b	If the generating polynomial for CRC code is x3+x+1 and dataword is 1101100100, generate the code word at the sender side.	5	CO2	PO1,PO4	L3

Question #	**Description**	**Marks Distribution**	**Max Marks**
1	Explain PCM encoder used for analog to digital conversion with steps and example. • Explain what is PCM with diagram • Explain steps involved in sampling, quantization & encoding with diagrams • Detailed example	2 M + 6 M + 2 M	10 M
2	(a) Briefly explain ASK & FSK modulation techniques with diagrams and specify the bandwidth requirements. • Explain ASK with diagram & implementation • Bandwidth expression with diagram for ASK • Explain FSK with diagram & implementation • Bandwidth expression with diagram for FSK	3 M + 1 M + 3 M + 1 M	10 M
	(b) Find the bandwidth for a signal transmitting at 12 Mbps for QPSK. The value of d=0. • Show formula and calculations	2 M	
3	(a) State and explain the data rate management to handle disparity in input data rates in TDM. • Explain the 3 methods with diagrams	2 M * 3	10 M
	(b) Two channels, one with a bit rate of 100 kbps and another with a bit rate of 200 kbps, are to be	1 M * 4	

	multiplexed. How can this be achieved? Calculate the frame rate, frame duration and bit rate of the link. • Show the calculation and mention the logic applied to deduce the answer.		
4	(a) What is spread spectrum? Mention its advantages. • Explain spread spectrum with diagram • Mention 2 advantages briefly	2 M * 2	10 M
	(b) Explain FHSS and discuss its merits over FDM. • Explain the FHSS technique in detail including its implementation • Show the block diagram & slot allocations • Compare with FDM using diagram	2 M * 3	
5	Explain the 3 phases of switching at the data link layer with necessary diagrams. Also obtain an expression for total delay. • 3 phases - Setup phase, Data transfer phase & Teardown phase explanation with diagrams • Expression for delay with diagram showing each delay component.	7 M + 3 M	10 M
6	(a) Explain simple parity check code with a neat diagram. • Block Diagram • Explanation of how parity checker works	2.5 M * 2	10 M
	(b) If the generating polynomial for CRC code is x^3+x+1 and dataword is 1101100100, generate the code word at the sender side. • Calculate value of n • Obtain augmented dataword • Perform modulo-2-division • Obtain check bits • Obtain codeword	1 M * 5	

1. Explain PCM encoder used for analog to digital conversion with steps and example.

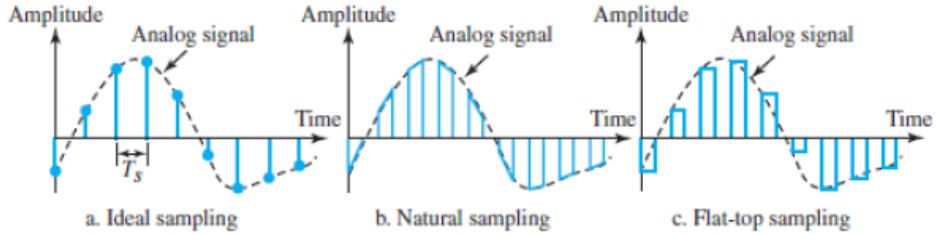
The most common technique to change an analog signal to digital data (digitization) is called pulse code modulation (PCM). A PCM encoder has three processes, as shown in the below figure.



1. The analog signal is sampled.
2. The sampled signal is quantized.
3. The quantized values are encoded as streams of bits.

i. **Sampling**

The first step in PCM is sampling. The analog signal is sampled every Ts s, where Ts is the sample interval or period. The inverse of the sampling interval is called the sampling rate or sampling frequency and denoted by fs, where fs = 1/Ts. There are three sampling methods—ideal, natural, and flat-top as shown in the next figure.



a. Ideal sampling b. Natural sampling c. Flat-top sampling

In ideal sampling, pulses from the analog signal are sampled. This is an ideal sampling method and cannot be easily implemented. In natural sampling, a high-speed switch is turned on for only the small period of time when the sampling occurs. The result is a sequence of samples that retains the shape of the analog signal. The most common sampling method, called sample and hold, however, creates flat-top samples by using a circuit.

The sampling process is sometimes referred to as pulse amplitude modulation (PAM). The result of sampling is still an analog signal with nonintegral values.

Sampling Rate

According to the Nyquist theorem, to reproduce the original analog signal, one necessary condition is that the sampling rate be at least twice the highest frequency in the original signal.

ii. **Quantization**

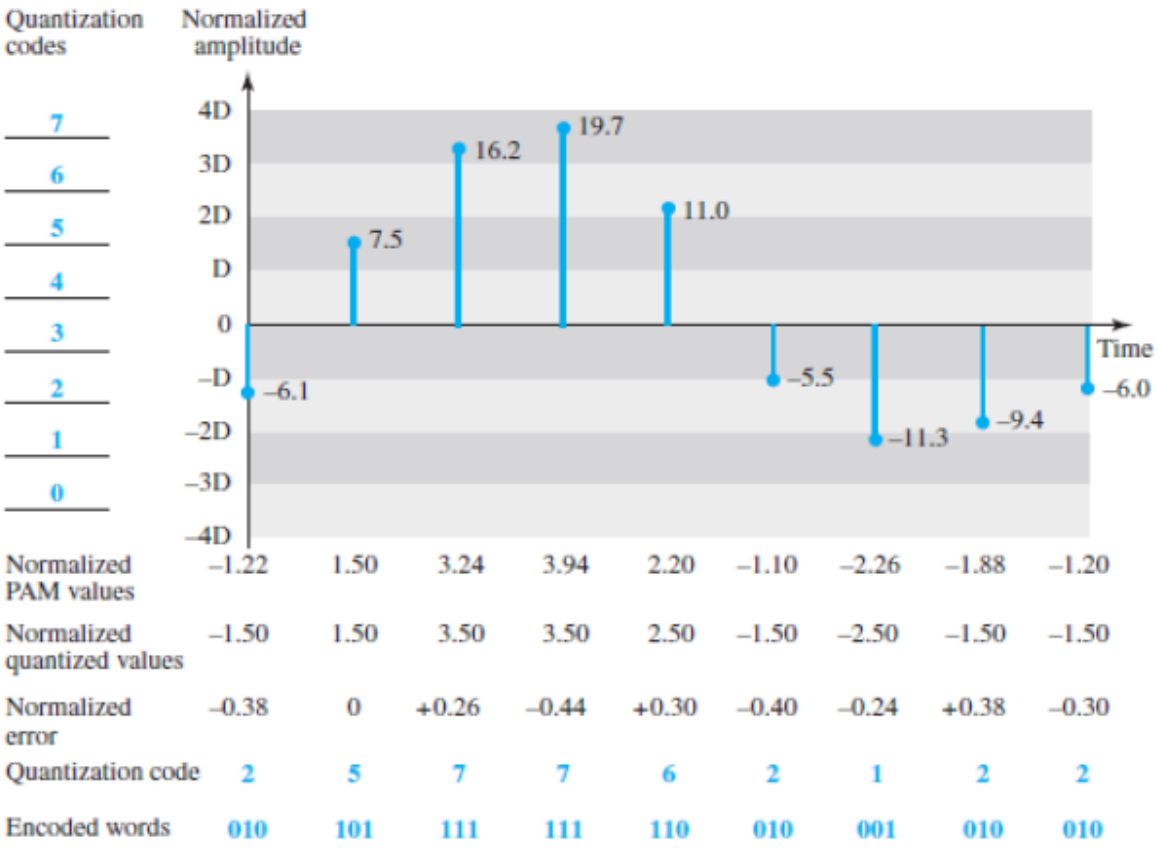
The result of sampling is a series of pulses with amplitude values between the maximum and minimum amplitudes of the signal. The set of amplitudes can be infinite with nonintegral values between the two limits. These values cannot be used in the encoding process. The following are the steps in quantization:

a. We assume that the original analog signal has instantaneous amplitudes between Vmin and Vmax.
b. We divide the range into L zones, each of height Δ (delta).

$$\Delta = \frac{V_{max} - V_{min}}{L}$$

c. We assign quantized values of 0 to L − 1 to the midpoint of each zone.
d. We approximate the value of the sample amplitude to the quantized values.

For example, assume that we have a sampled signal and the sample amplitudes are between −20 and +20 V. We decide to have eight levels (L = 8). This means that Δ = 5 V. Figure shows this example.



We have shown only nine samples using ideal sampling (for simplicity). The value at the top of each sample in the graph shows the actual amplitude. In the chart, the first row is the normalized value for each sample (actual amplitude/Δ). The quantization process selects the quantization value from the middle of each zone. This means that the normalized quantized values (second row) are different from the normalized amplitudes. The difference is called the normalized error (third row). The fourth row is the quantization code for each sample based on the quantization levels at the left of the graph. The encoded words (fifth row) are the final products of the conversion.

Quantization Levels

In the previous example, we showed eight quantization levels. The choice of L, the number of levels, depends on the range of the amplitudes of the analog signal and how accurately we need to recover the signal. If the amplitude of a signal fluctuates between two values only, we need only two levels; if the signal, like voice, has many amplitude values, we need more quantization levels. In audio digitizing, L is normally chosen to be 256; in video it is normally thousands. Choosing lower values of L increases the quantization error if there is a lot of fluctuation in the signal.

Quantization Error

One important issue is the error created in the quantization process. Quantization is an approximation process. The input values to the quantizer are the real values; the output values are the approximated values. The output values are chosen to be the middle value in the zone. If the input value is also at the middle of the zone, there is no quantization error; otherwise, there is an error. In the previous example, the normalized amplitude of the third sample is 3.24, but the normalized quantized value is 3.50. This means that there is an error of +0.26. The value of the error for any sample is less than Δ/2. In other words, we have $-\Delta/2 \leq \text{error} \leq \Delta/2$.

iii. Encoding

The last step in PCM is encoding. After each sample is quantized and the number of bits per sample is decided, each sample can be changed to an nb-bit code word. In the previous figure, the encoded words are shown in the last row. A quantization code of 2 is encoded as 010; 5 is encoded as 101; and so on. Note that the number of bits for each sample is determined from the number of quantization levels. If the number of quantization levels is L, the number of bits is nb = log2 L. In our example L is 8 and nb is therefore 3. The bit rate can be found from the formula

$$\text{Bit rate} = \text{sampling rate} \times \text{number of bits per sample} = f_s \times n_b$$

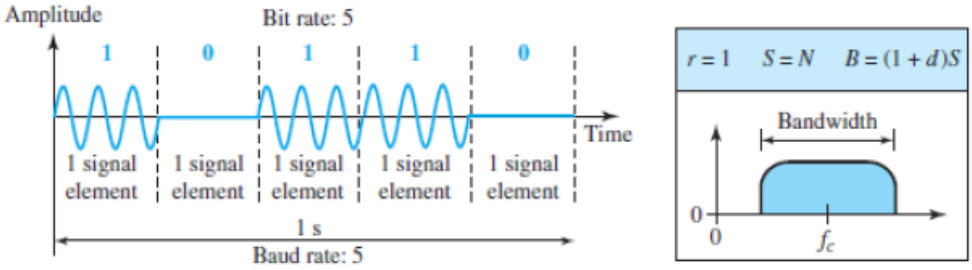
2a. Briefly explain ASK & FSK modulation techniques with diagrams and specify the bandwidth requirements.

Amplitude Shift Keying (ASK):

In amplitude shift keying, the amplitude of the carrier signal is varied to create signal elements. Both frequency and phase remain constant while the amplitude changes.

1. Binary ASK (BASK)

ASK is normally implemented using only two levels. This is referred to as binary amplitude shift keying or *on- off keying* (OOK). The peak amplitude of one signal level is 0; the other is the same as the amplitude of the carrier frequency. Figure 5.3 gives a conceptual view of binary ASK.



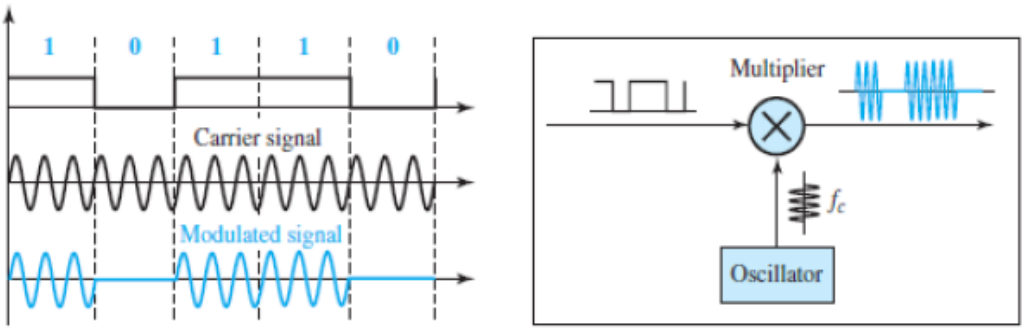
Bandwidth for ASK The above figure also shows the bandwidth for ASK. Although the carrier signal is only one simple sine wave, the process of modulation produces a nonperiodic composite signal.As we expect, the bandwidth is proportional to the signal rate (baud rate). However, there is normally another factor involved, called *d*, which depends on the modulation and filtering process. The value of *d* is between 0 and 1. This means that the bandwidth can be expressed as shown, where 5 is the signal rate and the *B* is the bandwidth.

$B = (1 + d) \times S$

The formula shows that the required bandwidth has a minimum value of S and a maximum value of 2S. The most important point here is the location of the bandwidth.

The middle of the bandwidth is where f_c, the carrier frequency, is located. This means if we have a bandpass channel available, we can choose our f_c so that the modulated signal occupies that bandwidth.

The next figure shows how we can simply implement binary ASK.



If digital data are presented as a unipolar NRZ digital signal with a high voltage of I V and a low voltage of 0 V, the implementation can achieved by multiplying the NRZ digital signal by the carrier signal coming from an oscillator. When the amplitude of the NRZ signal is 1, the amplitude of the carrier frequency is held; when the amplitude of the NRZ signal is 0, the amplitude of the carrier frequency is zero.

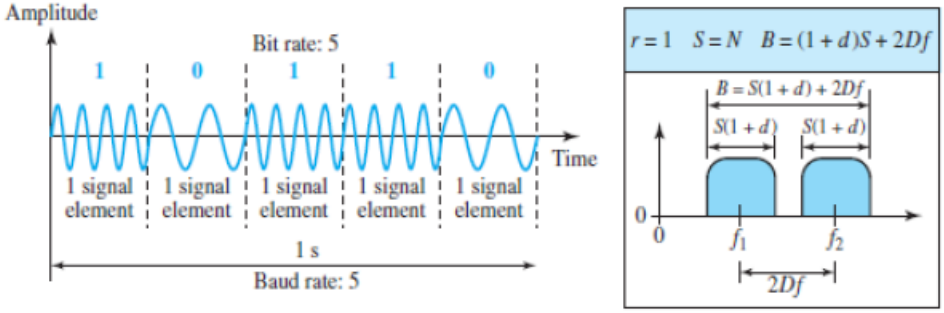
We can have multilevel ASK in which there are more than two levels. We can use 4,8, 16, or more different amplitudes for the signal and modulate the data using 2, 3, 4, or more bits at a time. In these cases, r = 2, r = 3, r =4, and so on.

Frequency Shift Keying (FSK):

In frequency shift keying, the frequency of the carrier signal is varied to represent data. The frequency of the modulated signal is constant for the duration of one signal element, but changes for the next signal element if the data element changes. Both peak amplitude and phase remain constant for all signal elements

Binary FSK (BFSK) One way to think about binary FSK (or BFSK) is to consider two carrier frequencies. In Figure 5.6, we have selected two carrier frequencies,f1 and f2. We use the first carrier if the data element is 0; we use the second if the data element is 1.

Figure 5.6 *Binary frequency shift keying*



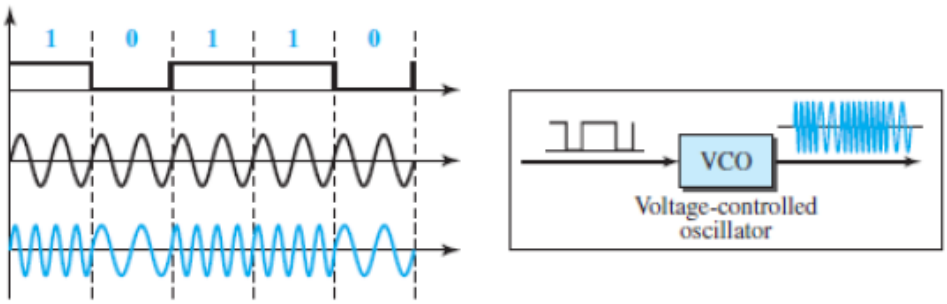
As Figure 5.6 shows, the middle of one bandwidth is fl and the middle of the other is *f2*. Both fl and *f2* are Δf apart from the midpoint between the two bands. The difference between the two frequencies is *2*Δf.

Bandwidth for BFSK Figure 5.6 also shows the bandwidth of FSK. Again the carrier signals are only simple sine waves, but the modulation creates a nonperiodic composite signal with continuous frequencies. We can think of FSK as two ASK signals, each with its own carrier frequency fl and f2*)*. If the difference between the two frequencies is *2*Δf, then the required bandwidth is

$$B = (1+d) \times S + 2\Delta f$$

Implementation There are two implementations of BFSK: **noncoherent and coherent**. In noncoherent BFSK, there may be discontinuity in the phase when one signal element ends and the next begins. In coherent BFSK, the phase continues through the boundary of two signal elements. Noncoherent BFSK can be implemented by treating BFSK as two ASK modulations and using two carrier frequencies. Coherent BFSK can be implemented by using one *voltage-controlled oscillator* (VeO) that changes its frequency according to the input voltage. Figure 5.7 shows the simplified idea behind the second implementation. The input to the oscillator is the unipolar NRZ signal. When the amplitude of NRZ is zero, the oscillator keeps its regular frequency; when the amplitude is positive, the frequency is increased.

Figure 5.7 *Implementation ofBFSK*



Multilevel modulation (MFSK) is not uncommon with the FSK method. We can use more than two frequencies. For example, we can use four different frequencies f1,f2,f3, and f4 to send 2 bits at a time. To send 3 bits at a time, we can use eight frequencies. However, we need to remember that the frequencies need to be *2*Δf apart. For the proper operation of the modulator and demodulator, it can be shown that the minimum value of *2*Δf needs to be S. We can show that the bandwidth with *d* =0 is

$B = (1+d) \times S + (L-1)\,2\Delta f \rightarrow B = L \times S$

2b. Find the bandwidth for a signal transmitting at 12 Mbps for QPSK. The value of d=0.

Solution
For QPSK, 2 bits is carried by one signal element. This means that *r* =2. So the signal rate (baud rate) is S =*N* x *(1/r)* =6 Mbaud. With a value of *d* =0, we have *B* =S =6 MHz.

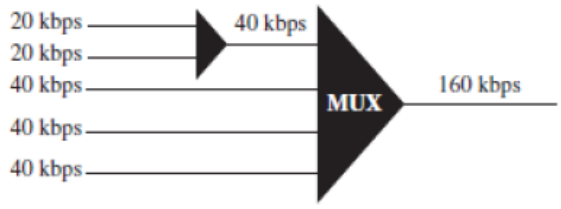
3a. State and explain the data rate management to handle disparity in input data rates in TDM.

If data rates are not the same, three strategies, or a combination of them, can be used. We call these 3 strategies multilevel multiplexing, multiple-slot allocation, and pulse stuffing.

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 6.19, 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.

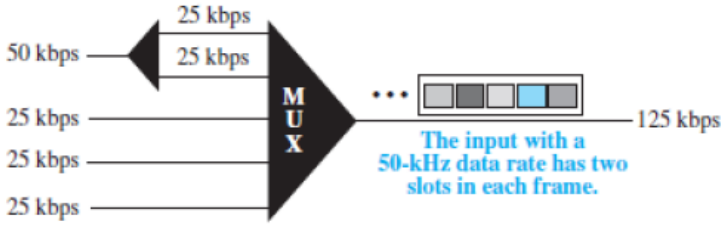
Figure 6.19 *Multilevel multiplexing*



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 6.20, 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.

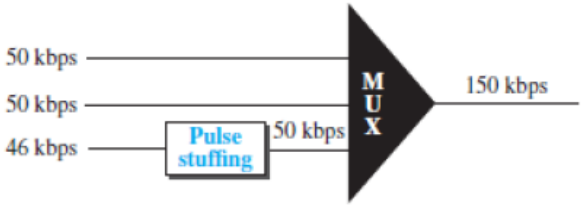
Figure 6.20 *Multiple-slot multiplexing*



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 6.21. The input with a data rate of 46 is pulse-stuffed to increase the rate to 50 kbps. Now multiplexing can take place.

Figure 6.21 *Pulse stuffing*



3b. Two channels, one with a bit rate of 100 kbps and another with a bit rate of 200 kbps, are to be multiplexed. How can this be achieved? Calculate the frame rate, frame duration and bit rate of the link.

Solution

We can allocate one slot to the first channel and two slots to the second channel. Each frame carries 3 bits. The frame rate is 100,000 frames per second because it carries 1 bit from the first channel. The frame duration is 1/100,000 s, or 10 ms. The bit rate is 100,000 frames/s × 3 bits per frame, or 300 kbps. Note that because each frame carries 1 bit from the first channel, the bit rate for the first channel is preserved. The bit rate for the second channel is also preserved because each frame carries 2 bits from the second channel.

4a. What is spread spectrum? Mention its advantages.

Spread spectrum is designed to be 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 (in military operations, for example).

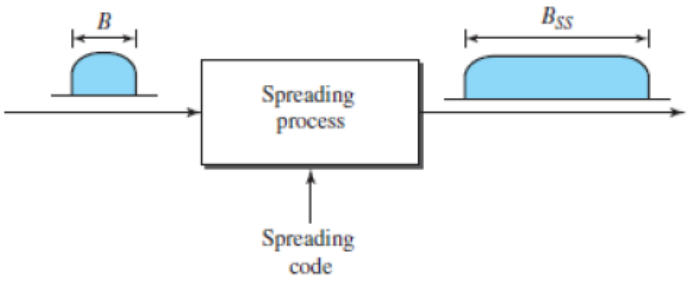
To achieve these goals, spread spectrum techniques add redundancy; they spread the original spectrum needed for each station. If the required bandwidth for each station is B, spread spectrum expands it to Bss, such that Bss >> B. The expanded bandwidth allows the source to wrap its message in a protective envelope for a more secure transmission.

Spread spectrum achieves its goals through two principles:

1. The bandwidth allocated to each station needs to be, by far, larger than what is needed. This allows redundancy.

2. 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.

Figure 6.27 *Spread spectrum*



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).

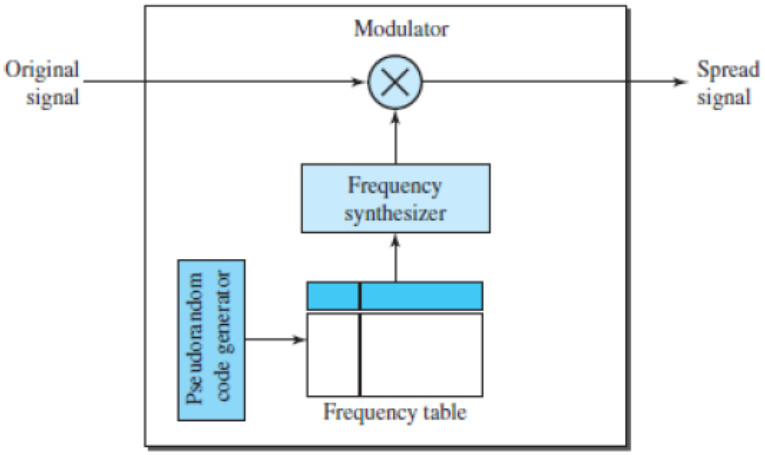
4b. Explain FHSS and discuss its merits over FDM.

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.*

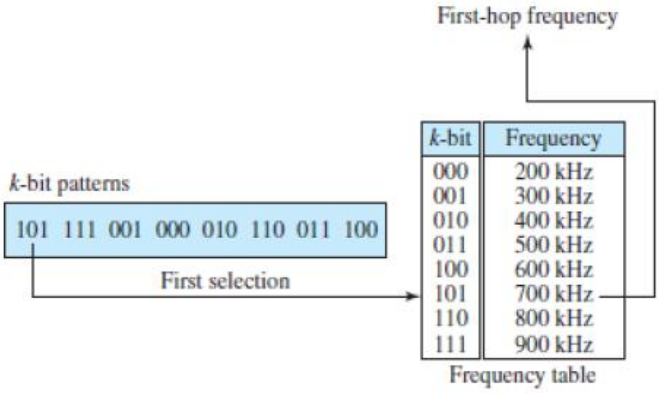
Figure 6.28 shows the general layout for FHSS. A pseudorandom code generator, called pseudorandom noise (PN), creates a k-bit pattern for every hopping period *Th•* The frequency table uses the pattern to find the frequency to be used for this hopping period and passes it to the frequency synthesizer. The

frequency synthesizer creates a carrier signal of that frequency, and the source signal modulates the carrier signal.

Figure 6.28 *Frequency hopping spread spectrum (FHSS)*

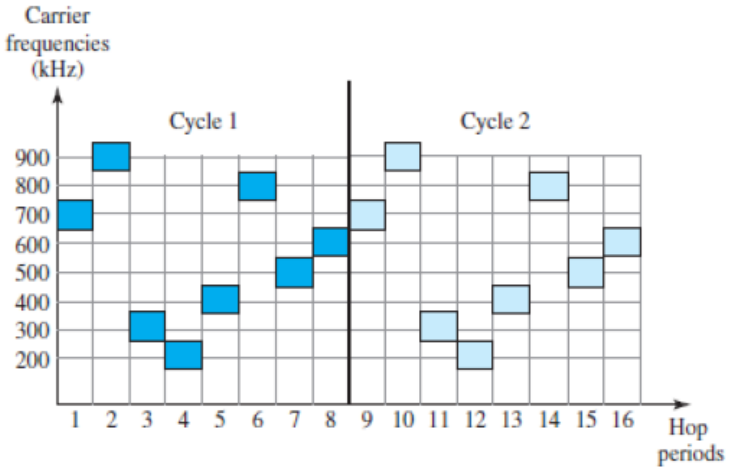


Suppose we have decided to have eight hopping frequencies. This is extremely low for real applications and is just for illustration. In this case, *M is* 8 and *k* is 3. The pseudorandom code generator will create eight different 3-bit patterns. These are mapped to eight different frequencies in the frequency table (see Figure 6.29).



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.

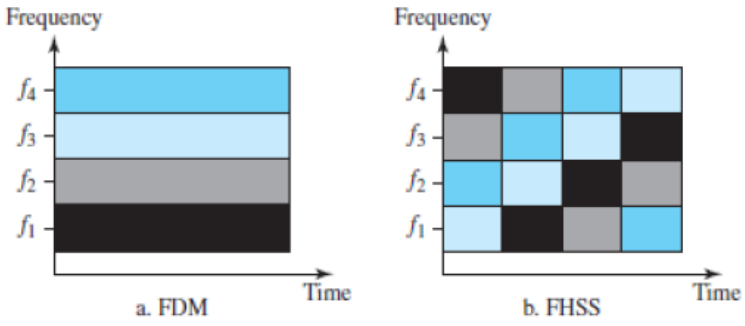
Figure 6.30 *FHSS cycles*



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 **antijamming** 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

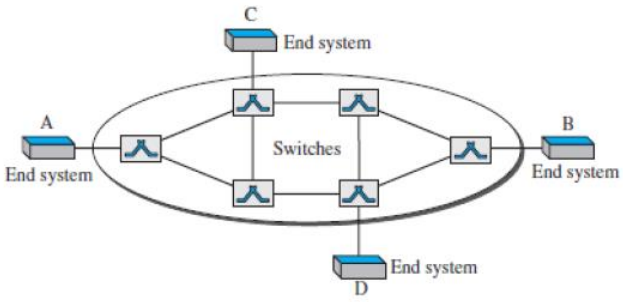
If the number of hopping frequencies is *M*, we can multiplex *M* channels into one by using the same Bss bandwidth. This is possible because a station uses just one frequency in each hopping period; *M* - 1 other frequencies can be used by other *M* - 1 stations. In other words, *M* different stations 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 6.31.Figure 6.31 shows an example of four channels using FDM and four channels using FHSS.



5. Explain the 3 phases of switching at the data link layer with necessary diagrams. Also obtain an expression for total delay.

A virtual-circuit network is normally implemented in the data-link layer.

Figure 8.10 *Virtual-circuit network*



As in a circuit-switched network, a source and destination need to go through three phases in a virtual-circuit network: setup, data transfer, and teardown. In the setup phase, the source and destination use their global addresses to help switches make table entries for the connection. In the teardown phase, the source and destination inform the switches to delete the corresponding entry. Data transfer occurs between these two phases.

1. Data-Transfer Phase

To transfer a frame from a source to its destination, all switches need to have a table entry for this virtual circuit. The table, in its simplest form, has four columns. This means that the switch holds four pieces of information for each virtual circuit that is already set up.

Figure 8.12 shows such a switch and its corresponding table.

Figure 8.12 *Switch and tables in a virtual-circuit network*

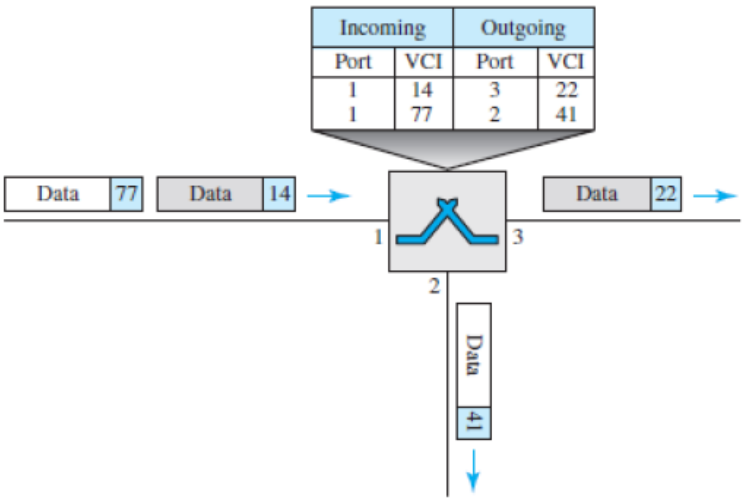
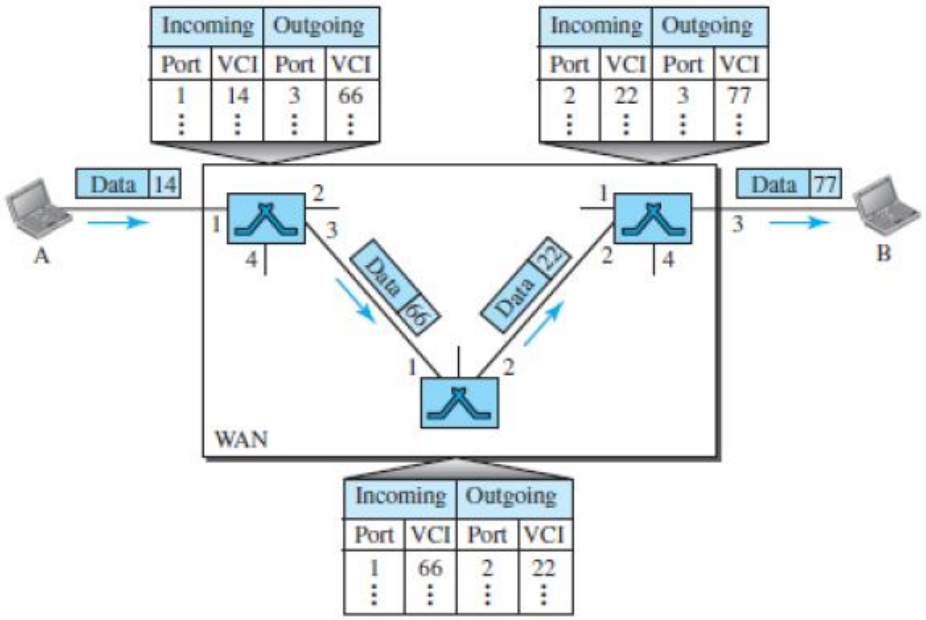


Figure 8.12 shows a frame arriving at port 1 with a VCI of 14. When the frame arrives, the switch looks in its table to find port 1 and a VCI of 14. When it is found, the switch knows to change the VCI to 22 and send out the frame from port 3.

Figure 8.13 shows how a frame from source A reaches destination B and how its VCI changes during the trip. Each switch changes the VCI and routes the frame. The data-transfer phase is active until the source sends all its frames to the destination. The procedure at the switch is the same for each frame of a message. The process creates a virtual circuit, not a real circuit, between the source and destination.

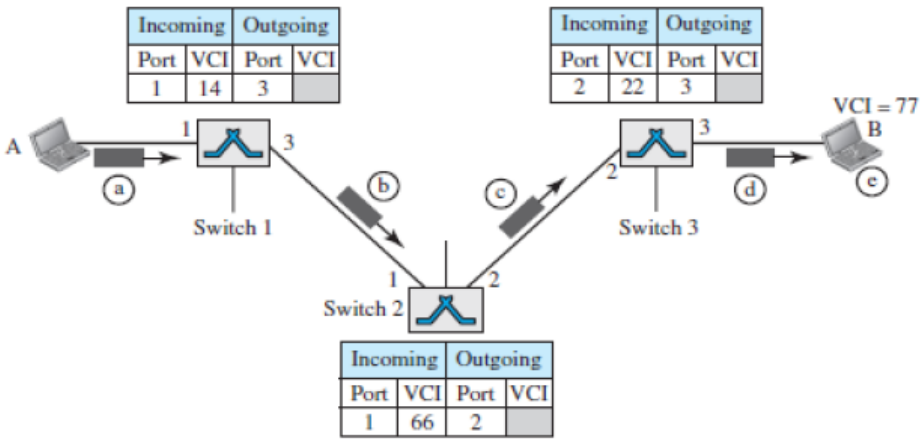
Figure 8.13 *Source-to-destination data transfer in a virtual-circuit network*



3. Setup Request

A setup request frame is sent from the source to the destination. Figure 8.14 shows the process.

Figure 8.14 *Setup request in a virtual-circuit network*

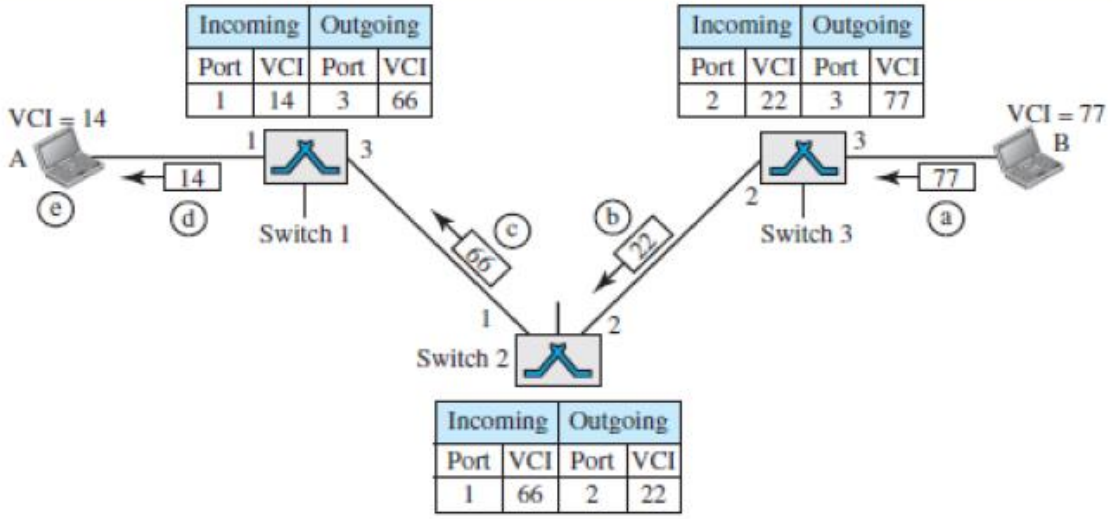


a. Source A sends a setup frame to switch 1.
b. Switch 1 receives the setup request frame. It knows that a frame going from A to B goes out through port 3. The switch, in the setup phase, acts as a packet switch; it has a routing table which is different from the switching table. For the moment, assume that it knows the output port. The switch creates an entry in its table for this virtual circuit, but it is only able to fill three of the four columns. The switch assigns the incoming port (1) and chooses an available incoming VCI (14) and the outgoing port (3). It does not yet know the outgoing VCI, which will be found during the acknowledgment step. The switch then forwards the frame through port 3 to switch 2.
c. Switch 2 receives the setup request frame. The same events happen here as at switch 1; three columns of the table are completed: in this case, incoming port (1),incoming VCI (66), and outgoing port (2).
d. Switch 3 receives the setup request frame. Again, three columns are completed: incoming port (2), incoming VCI (22), and outgoing port (3).
e. Destination B receives the setup frame, and if it is ready to receive frames from A,it assigns a VCI to the incoming frames that come from A, in this case 77. This VCI lets the destination know that the frames come from A, and not other sources.

Acknowledgment

A special frame, called the *acknowledgment frame,* completes the entries in the switching tables. Figure 8.15 shows the process.

Figure 8.15 *Setup acknowledgment in a virtual-circuit network*



a. The destination sends an acknowledgment to switch 3. The acknowledgment carries the global source and destination addresses so the switch knows which entry in the table is to be completed. The frame also carries VCI 77, chosen by the destination as the incoming VCI for frames from A. Switch 3 uses this VCI to complete the outgoing VCI column for this entry. Note that 77 is the incoming VCI for destination B, but the outgoing VCI for switch 3.
b. Switch 3 sends an acknowledgment to switch 2 that contains its incoming VCI in the table, chosen in the previous step. Switch 2 uses this as the outgoing VCI in the table.

c. Switch 2 sends an acknowledgment to switch 1 that contains its incoming VCI in the table, chosen in the previous step. Switch 1 uses this as the outgoing VCI in the table.
d. Finally switch 1 sends an acknowledgment to source A that contains its incoming VCI in the table, chosen in the previous step.
e. The source uses this as the outgoing VCI for the data frames to be sent to destination B

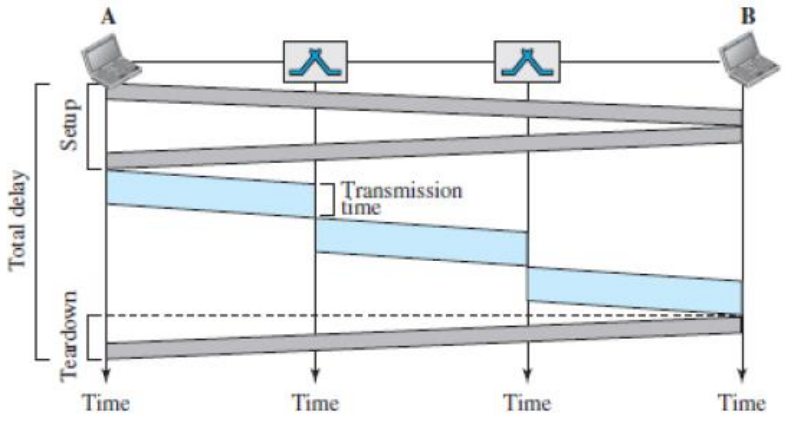
Teardown Phase

In this phase, source A, after sending all frames to B, sends a special frame called a *teardown request*. Destination B responds with a teardown confirmation frame. All switches delete the corresponding entry from their tables.

Delay in Virtual-Circuit Networks

In a virtual-circuit network, there is a one-time delay for setup and a one-time delay for teardown. If resources are allocated during the setup phase, there is no wait time for individual packets. Figure 8.16 shows the delay for a packet traveling through two switches in a virtual-circuit network.

Figure 8.16 *Delay in a virtual-circuit network*



The packet is traveling through two switches (routers). There are three transmission times ($3T$), three propagation times (3τ), data transfer depicted by the sloping lines, a setup delay (which includes transmission and propagation in two directions), and a teardown delay (which includes transmission and propagation in one direction). The total delay time is

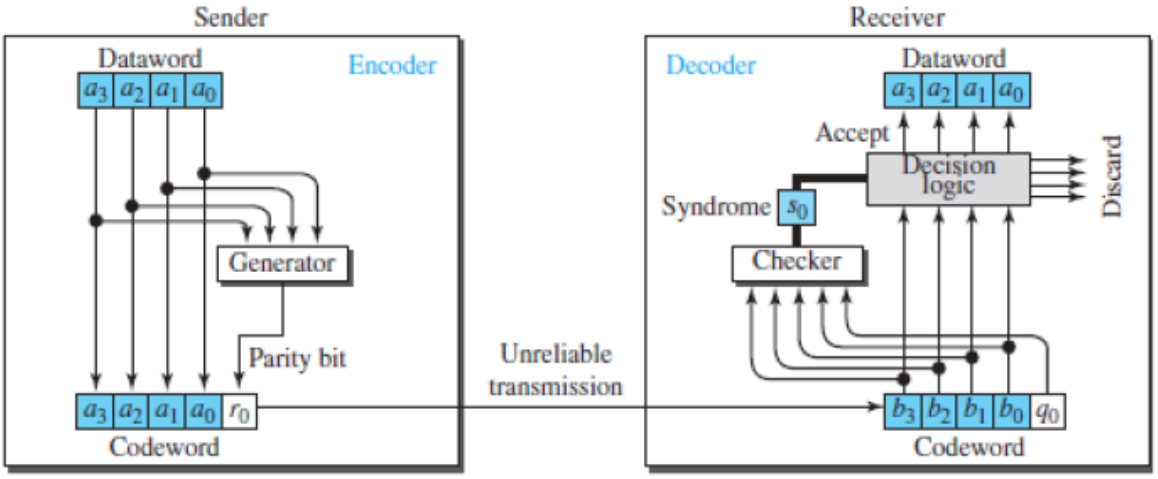
$$\text{Total Delay} = 3T + 3\tau + \text{setup delay} + \text{teardown delay}$$

6a. Explain simple parity check code with a neat diagram.

In parity check code, a k-bit data word is changed to an n-bit codeword where n = k + 1. The extra bit, called the parity bit, is selected to make the total number of 1s in the codeword even.

The minimum Hamming distance for this category is dmin =2, which means that the code is a single-bit error-detecting code.

Figure 10.4 *Encoder and decoder for simple parity-check code*



The calculation is done in modular arithmetic. The encoder uses a generator that takes a copy of a 4-bit dataword (a0, a1, a2, and a3) and generates a parity bit r0. The dataword bits and the parity bit create the 5-bit codeword. The parity bit that is added makes the number of 1s in the codeword even. This is normally done by adding the 4 bits of the dataword (modulo-2); the result is the parity bit. In other words,

$$r_0 = a_3 + a_2 + a_1 + a_0 \quad \text{(modulo-2)}$$

If the number of 1s is even, the result is 0; if the number of 1s is odd, the result is 1. In both cases, the total number of 1s in the codeword is even. The sender sends the codeword which may be corrupted during transmission. The receiver receives a 5-bit word. The checker at the receiver does the same thing as the generator in the sender with one exception: The addition is done over all 5 bits. The result, which is called the syndrome, is just 1 bit. The syndrome is

0 when the number of 1s in the received codeword is even; otherwise, it is 1.

$$s_0 = b_3 + b_2 + b_1 + b_0 + q_0 \quad \text{(modulo-2)}$$

The syndrome is passed to the decision logic analyzer. If the syndrome is 0, there is no error in the received codeword; the data portion of the received codeword is accepted as the data word; if the syndrome is 1, the data portion of the received codeword is discarded. The data word is not created.

6b. If the generating polynomial for CRC code is x^3+x+1 and dataword is 1101100101, generate the code word at the sender side.