Module 3: Physical Layer

Dr. Natarajan Meghanathan
Associate Professor of Computer Science
Jackson State University
Jackson, MS 39217
Phone: 601-979-3661
E-mail: natarajan.meghanathan@jsums.edu

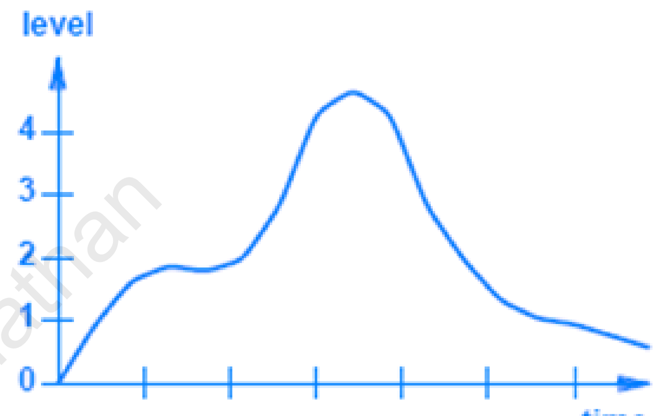
Topics

- 3.1 Signal Levels: Baud rate and Bit rate
- 3.2 Channel Encoding Standards
  - RS-232 and Manchester Encoding
  - Delay during transmission
- 3.3 Transmission Order of Bits and Bytes
- 3.4 Modulation Techniques
  - Amplitude, Frequency and Phase modulation
- 3.5 Multiplexing Techniques
  - TDMA, FDMA, Statistical Multiplexing and CDMA

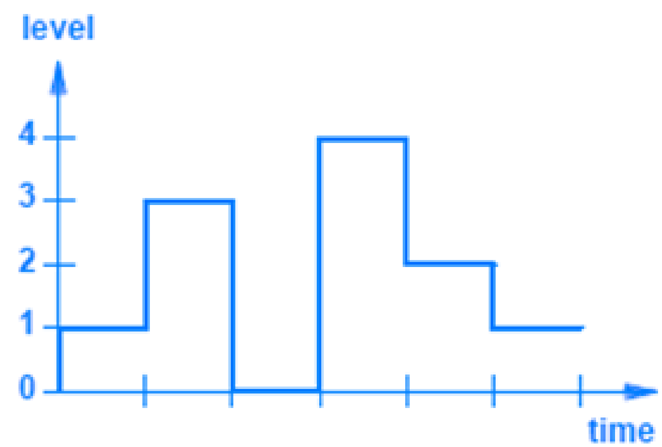
3.1 Signal Levels: Baud Rate and Bit Rate

Analog and Digital Signals

- Data communications deals with two types of information:
  - analog
  - digital
- An analog signal is characterized by a continuous mathematical function
  - when the input changes from one value to the next, it does so by moving through all possible intermediate values
- A digital signal has a fixed set of valid levels
  - each change consists of an instantaneous move from one valid level to another



(a)



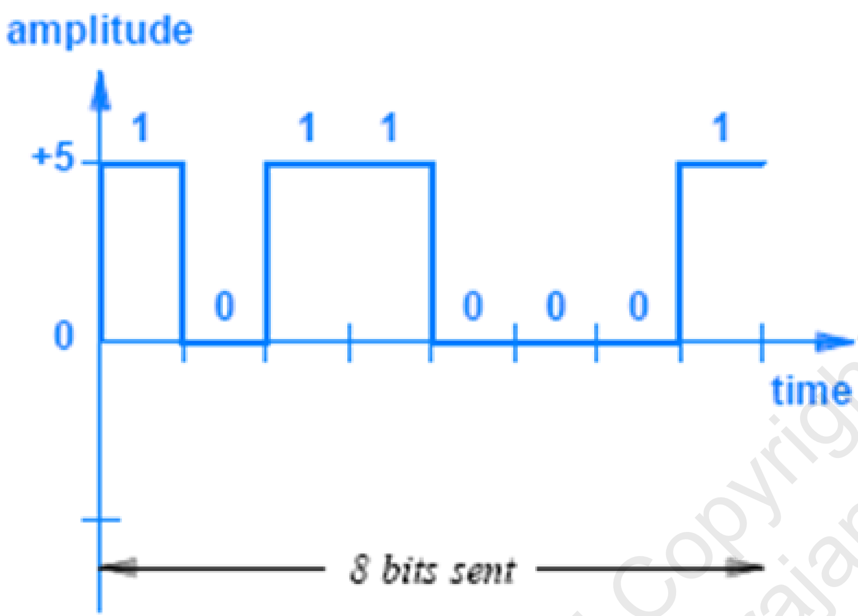
Digital Signals and Signal Levels

- Some systems use voltage to represent digital values
  - by making a positive voltage correspond to a logical one
  - and zero voltage correspond to a logical zero
- For example, ***+5*** volts can be used for a logical ***one*** and ***0*** volts for a logical ***zero***
- If only two levels of voltage are used
  - each level corresponds to one data bit (0 or 1).
- Some physical transmission mechanisms can support more than two signal levels
- When multiple digital levels are available
  - each level can represent multiple bits
- For example, consider a system that uses four levels of voltage:

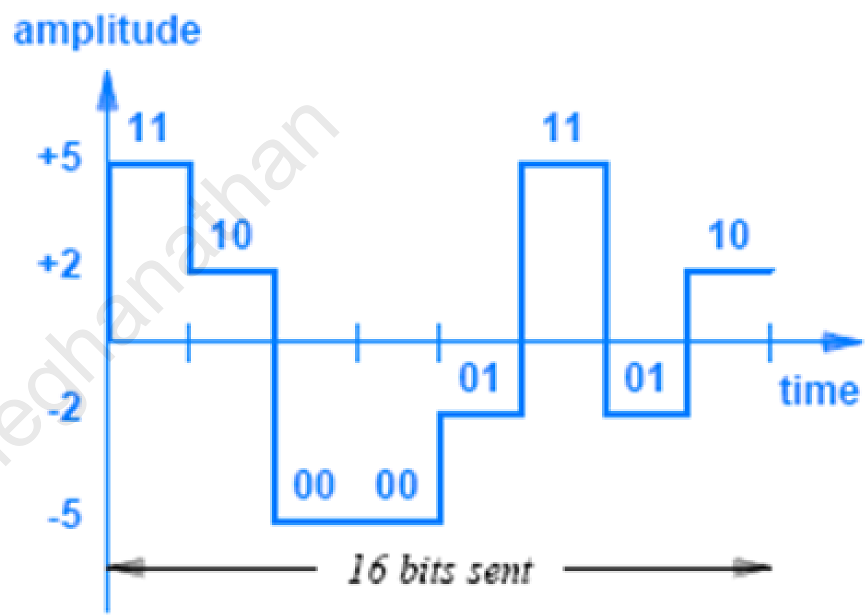
  -5 volts, -2 volts, +2 volts, and +5 volts

  - Each level can correspond to two bits of data as Figure 6.8 illustrates

Digital Signals and Signal Levels



Digital signal using
2-voltage levels



Digital signal using
4-voltage levels

Digital Signals and Signal Levels

- The relationship between the number of levels required and the number of bits to be sent is straightforward
- There must be a signal level for each possible combination of bits
- There are 2^n combinations possible with n bits
  - a communication system must use 2^n levels to represent n bits
- One could achieve arbitrary numbers of levels by dividing voltage into arbitrarily small increments
  - Mathematically, one could create a million levels between 0 and 1 volts merely by using 0.0000001 volts for one level, 0.0000002 for the next level, and so on
- Practical electronic systems cannot distinguish between signals that differ by arbitrarily small amounts
  - Thus, practical systems are restricted to a few signal levels

Baud and Bits Per Second

- How much data can be sent in a given time?
  - The answer depends on two aspects of the communication system.
- The rate at which data can be sent depends on
  - the number of signal levels
  - the amount of time the system remains at a given level before moving to the next
- As with signal levels, the hardware in a practical system places limits on how short the time can be
  - if the signal does not remain at a given level long enough, the receiving hardware will fail to detect it
- The accepted measure of a communication system does not specify a length of time
  - how many times the signal can change per second, which is defined as the baud
  - for example, if a system requires the signal to remain at a given level for .001 seconds, we say that the system operates at 1000 baud
- Both baud and number of signal levels control bit rate

Baud and Bits Per Second

- If a system with two signal levels operates at 1000 baud
  - the system can transfer exactly 1000 bits per second
- If a system that operates at 1000 baud has four signal levels
  - the system can transfer 2000 bits per second (because four signal levels can represent two bits)
- Equation below expresses the relationship between baud, signal levels, and bit rate

$$bits\ per\ second = baud \times \left\lfloor \log_2(\ levels\ ) \right\rfloor$$

3.2 Channel Encoding Standards

Asynchronous Transmission

- It is asynchronous if the system allows the physical medium to be idle for an arbitrary time between two transmissions
- The asynchronous style of communication is well-suited to applications that generate data at random
  - (e.g., a user typing on a keyboard or a user that clicks on a link)
- The disadvantage of asynchrony arises from the lack of coordination between sender and receiver
  - While the medium is idle, a receiver cannot know how long the medium will remain idle before more data arrives
- Asynchronous technologies usually arrange for a sender to transmit a few extra bits before each data item
  - to inform the receiver that a data transfer is starting
  - extra bits allow the receiver to synchronize with the incoming signal
  - the extra bits are known as a preamble or start bits

RS-232 Asynchronous Character Transmission

- Consider the transfer of characters across copper wires between a computer and a device such as a keyboard
  – each data item represents one character
- It is standardized by the Electronic Industries Alliance (EIA)
  - It has become the most widely used for character communication
  - Known as RS-232-C, and commonly abbreviated RS-232
- EIA standard specifies the details, such as
  – physical connection size (max cable length 50 feet long)
  – electrical details (range between -15v +15v)
  – the line coding being used
  – It can be configured to control the exact number of bits per second
  – It can be configured to send 7-bit or 8-bit characters

RS-232 Asynchronous Character Transmission

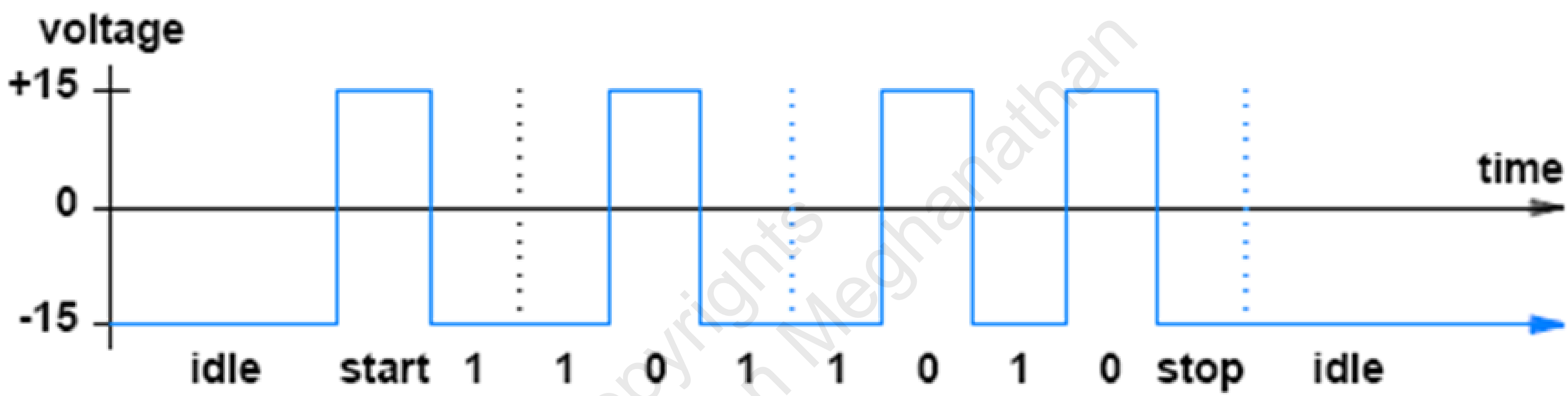
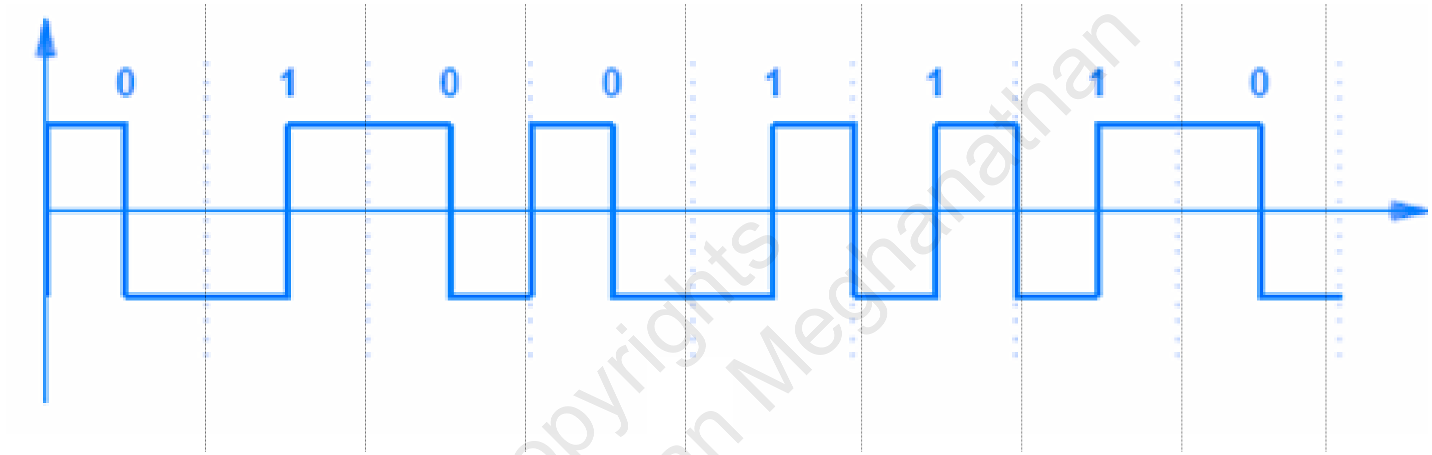


Figure above illustrates how voltage varies at different stages when a start bit, eight bits of a character, and a stop bit are sent

Manchester Encoding

- In addition to the RS-232 standard, one particular standard for line coding is especially important for networks:
  - Manchester Encoding used with Ethernet
- Detecting a transition in signal level is easier than measuring the signal level
- This fact explains why Manchester Encoding uses transitions rather than levels to define bits
- In Manchester Encoding, a 1 corresponds to a transition from negative voltage level to a positive voltage level
  - Correspondingly, a 0 corresponds to a transition from a positive voltage level to a negative level
  - The transitions occur in the "middle" of the time slot of a bit
- Figure 6.13a illustrates the concept

Manchester Encoding



Manchester Encoding uses a 64-bit preamble (alternating 0s and 1s)

Delays

- In network communication, the delay incurred to send data from a source to destination is a combination of the following components:
  - Transmission Delay
  - Propagation Delay
  - Queuing Delay (including the Switching Delay)
- Transmission Delay – the delay incurred to insert the bits of a packet onto a channel of a given bandwidth

$$\text{Transmission Delay} = \frac{\text{Packet Size (bits)}}{\text{Channel Bandwidth (bits/sec)}}$$

- Propagation Delay – the delay incurred for a packet to propagate (for the signal to move) on a channel of a particular length

$$\text{Propagation Delay} = \frac{\text{Channel Length (m)}}{\text{Speed of the Signal (m/s)}}$$

Sample Question: RS-232 Std. & Delay

- Determine the transmission delay, propagation delay and the total delay incurred to transmit data of size 2000 characters using the RS-232 standard. Assume the channel bandwidth is 40000 bits/sec and length is $2*10^6$m. Assume the speed of the signal on the channel is 60% of the speed of light.

- Solution:
  - To transmit a character according to the RS-232 standard, 10 bits (8 data bits plus 1 start and 1 stop bits) are needed. Hence, to transmit 2000 characters, we need 2000 * 10 = 20,000 bits
  - Transmission Delay = Data Size/ Channel Bandwidth

    = (20,000 bits)/ (40,000 bits/sec) = 0.50 sec
  - Propagation Delay = Channel length/ speed of the signal on the channel

    = $(2 * 10^6$ m)/ $(0.6 * 3 * 10^8$ m/s) = 0.011 sec
  - Total Delay = Transmission Delay + Propagation Delay = 0.511 sec

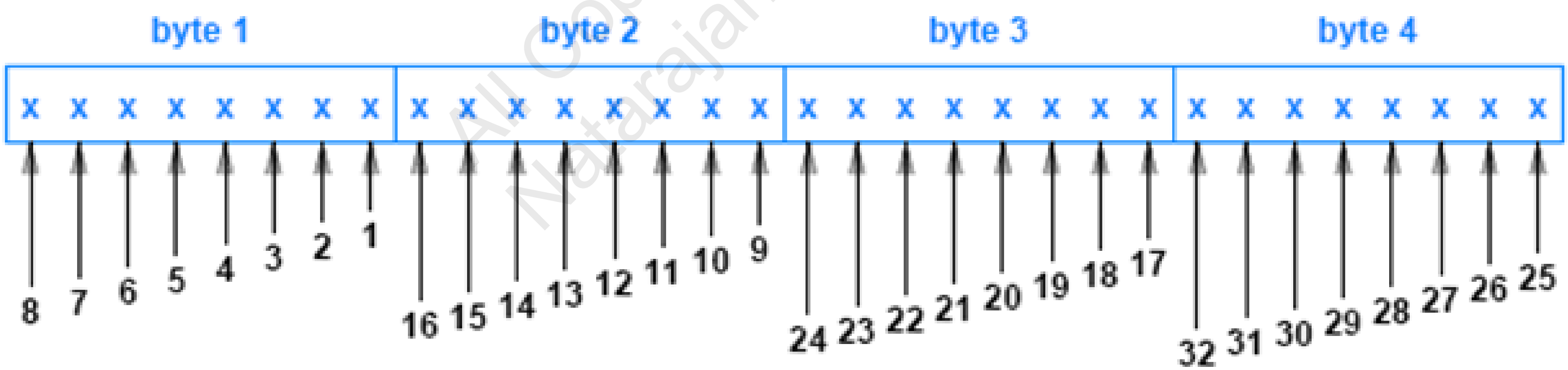
3.3 Transmission Order

Transmission Order: Bits and Bytes

- In serial mode, when sending bits, which bit should be sent across the medium first?
- Consider an integer: Should a sender transmit
  - the Most Significant Bit (MSB)
  - or the Least Significant Bit (LSB) first?
- We use the term little-endian to describe a system that sends the LSB first
- We use the term big-endian to describe a system that sends the MSB first
- Either form can be used, but the sender and receiver must agree

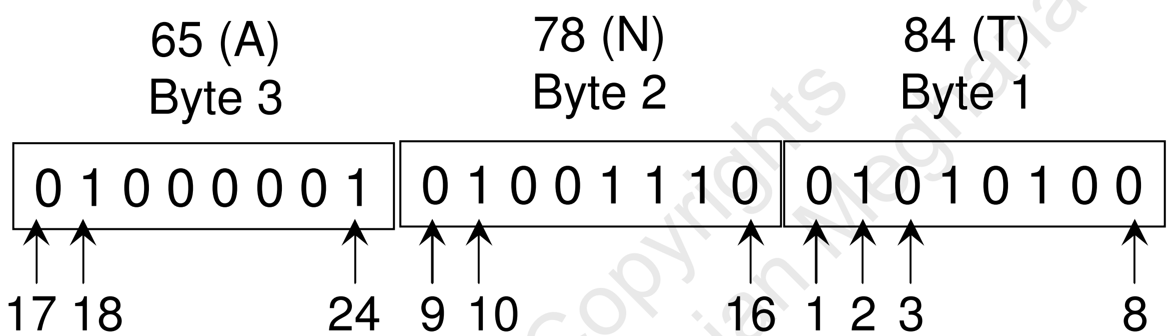
Transmission Order: Bits and Bytes

- The order in which bits are transmitted does not settle the entire question of transmission order
  - Data in a computer is divided into bytes, and each byte is further divided into bits (typically 8 bits per byte)
  - Thus, it is possible to choose a byte order and a bit order independently
  - For example, Ethernet technology specifies that data is sent byte big-endian and bit little-endian, as shown below for a 32-bit data

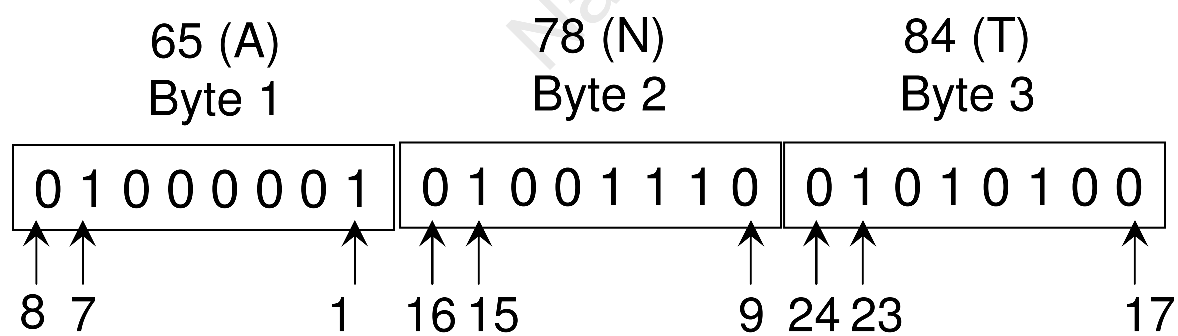


Sample Question: Transmission Order of Bits and Bytes

- Consider the word ‘ANT’ with the ASCII values of ‘A’, ‘N’ and ‘T’ being 65, 78 and 84 respectively. How would this word be transmitted if the transmission order is:
  - Byte little-endian and bit big-endian?



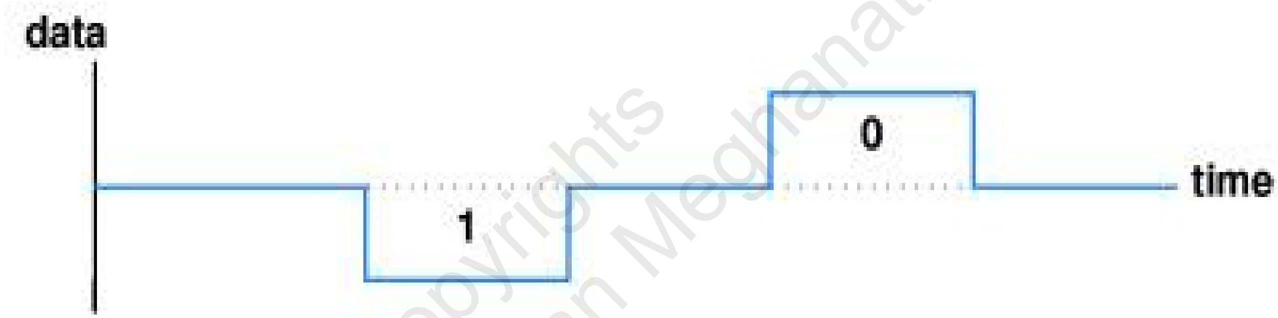
  - Byte big-endian and bit little-endian?



3.4 Modulation Techniques

Amplitude Modulation

- The amplitude of the carrier is modulated to encode data
- The strength of the carrier signal in (b) is reduced to 2/3rd of its Amplitude to encode a 1 bit and to 1/3rd of the Amplitude to encode 0 bit



(a)

(b)

Frequency Modulation

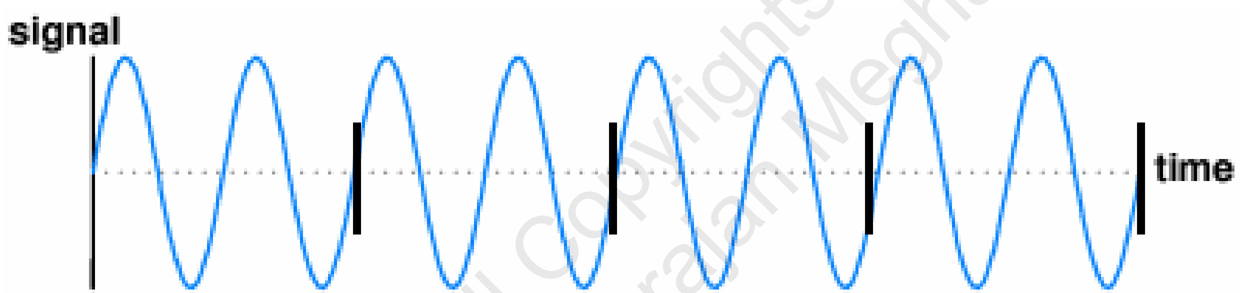
- The frequency of the carrier wave is changed to encode the transmission of the bits.

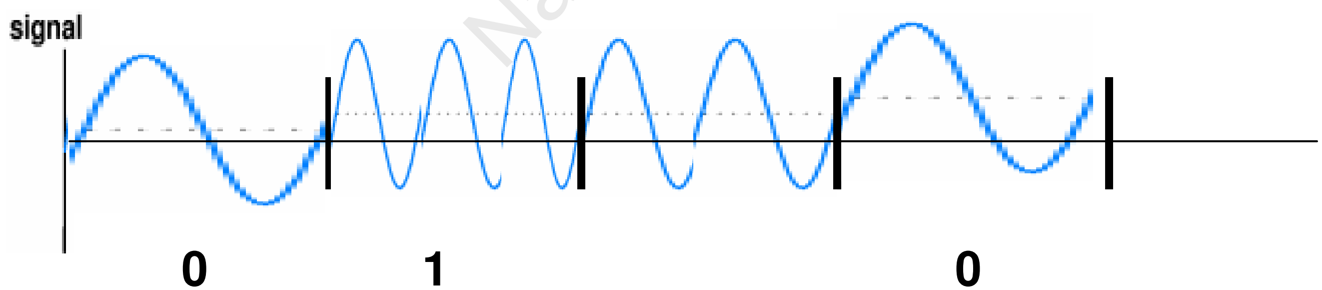
- The two binary digits are represented by two different frequencies that are offset from the carrier frequency by equal but opposite amount.

- Frequency of the carrier wave = f_c,
- Frequency of binary digit 0 = f_0,
- Frequency of binary digit 1 = f_1.
- $f_0 < f_c < f_1$; $f_c - f_1 = f_c - f_0$

Frequency Modulation - Example

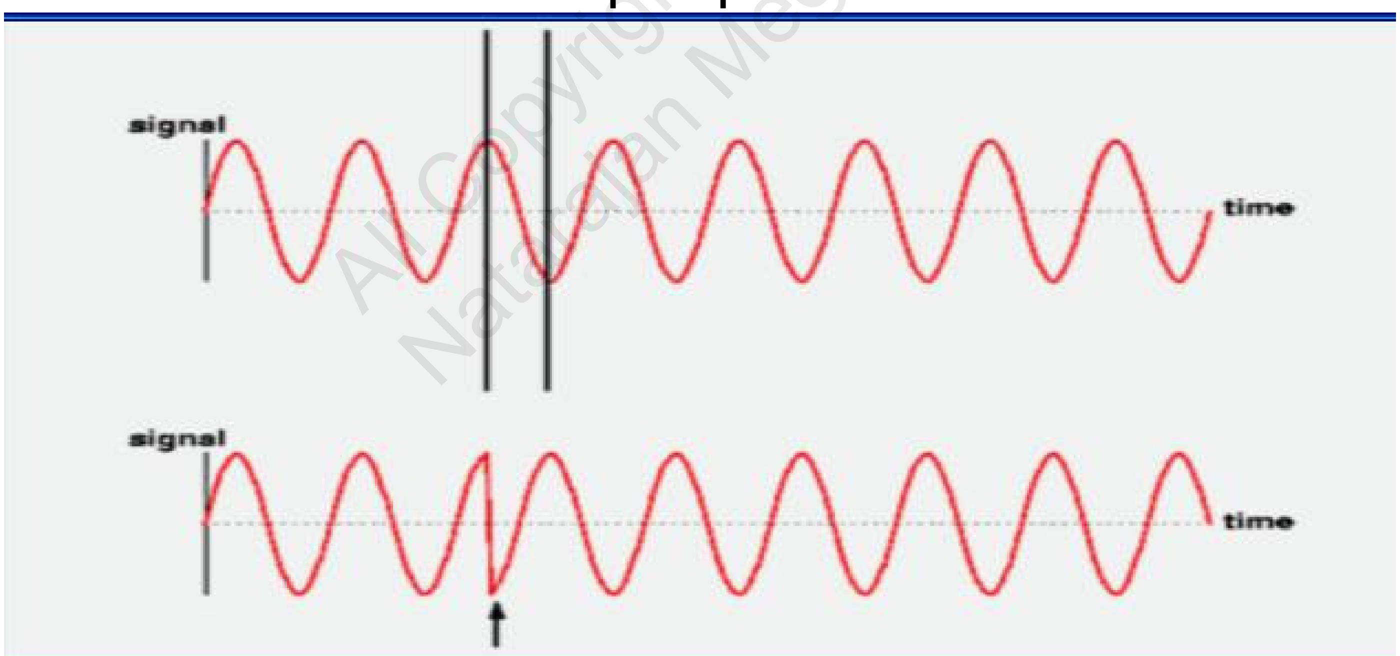
- Assume the bits are encoded for every two cycles (i.e., two cycles per unit time). Then, to encode data, we could change the frequency to one cycle (for bit 0) and three cycles (for bit 1)per unit time respectively.
- Assume the sequence of bits to be transmitted is: 0 1 _ 0, where _ indicates the channel is idle.





Phase Modulation

- The timing of the carrier wave is abruptly changed with a phase shift.
- Section of the wave is omitted at phase shift.
- The number of possible shifts that can be detected by the hardware within a cycle of the carrier wave determines the number of bits encoded per phase shift.



Phase Modulation

- The timing of the carrier wave is abruptly changed with a phase shift.

Two-level Phase Shift Keying (BPSK)

Use two phases to represent binary digits

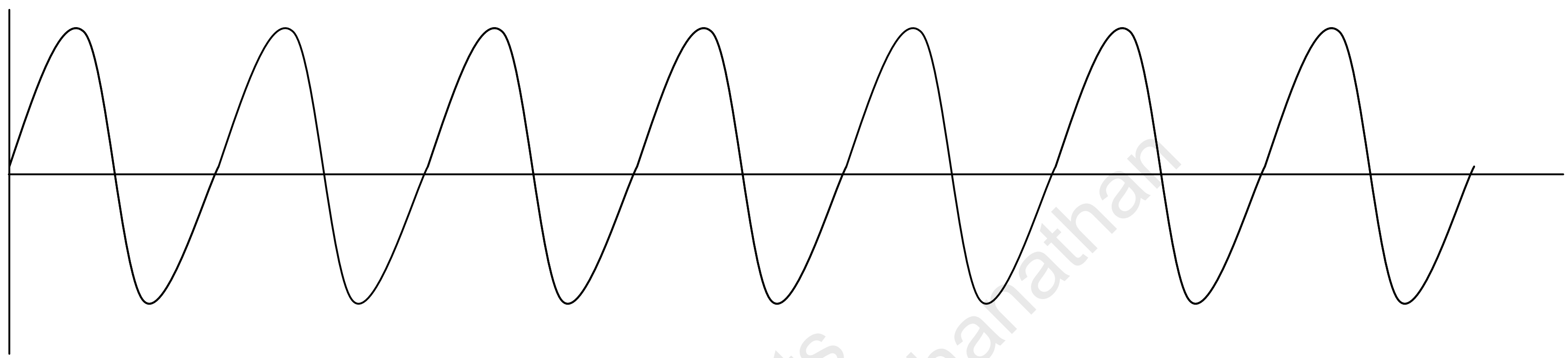
- Signal strength s(t) =

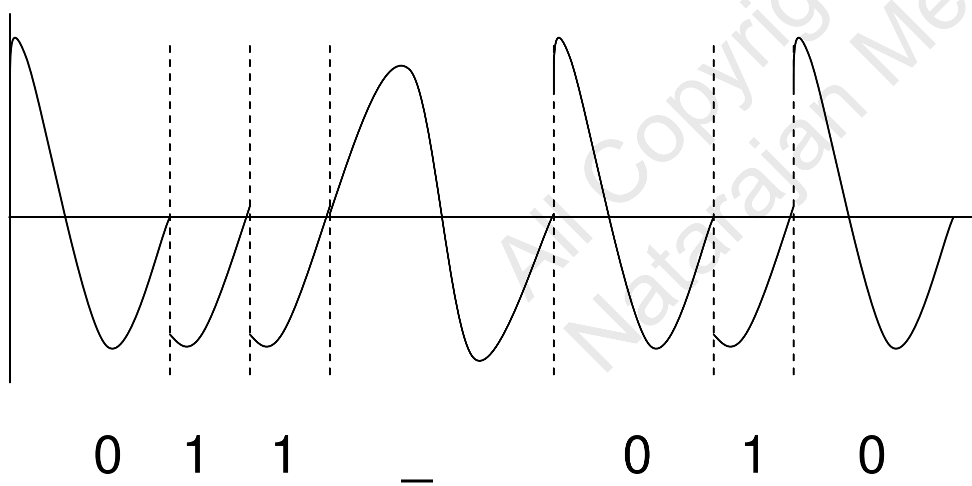
$$s(t) = \begin{cases} A \sin(2 \pi f t) & \text{Carrier} \\ A \sin(2 \pi f t + \pi/2) & \text{Bit 0} \\ A \sin(2 \pi f t + 3\pi/2) & \text{Bit 1} \end{cases}$$

- Can we do better? (i.e., can we encode multiple bits in a single cycle of the carrier?)
- Answer: Yes, using Multiple Phase Shift Keying and Multiple Frequency Shift Keying.

Not possible using amplitude modulation. Why?

Phase Modulation: Example





<u>Phase Shifts</u>
0 – π/2
1 - 3π/2

<u>Strength:</u> Less bandwidth
<u>Weakness:</u> The sender and receiver need to be frequently resynchronized to correctly detect the phase shifts

Phase Modulation

Example

Each signal element represents two bits

$$\text{Signal strength } S(t) = \begin{cases} A \sin(2 \pi f t) & \text{Carrier} \\ A \sin(2 \pi f t + \pi/4) & \text{bits 00} \\ A \sin(2 \pi f t + 3\pi/4) & \text{bits 01} \\ A \sin(2 \pi f t + 5\pi/4) & \text{bits 10} \\ A \sin(2 \pi f t + 7\pi/4) & \text{bits 11} \end{cases}$$

3.5 Multiplexing Techniques

Multiplexing

- **Broadband Technology** - Technology that uses a large bandwidth of the electromagnetic spectrum to achieve higher throughput (data transmitted per second). E.g., FDM, WDM (Wavelength Division Multiplexing – FDM when applied optical transmission systems and systems that use radio frequencies)

- **Baseband Technology** - Technology that uses a smaller bandwidth of the electromagnetic spectrum and sends only one signal at a time are called baseband technologies. E.g., TDM, Statistical multiplexing

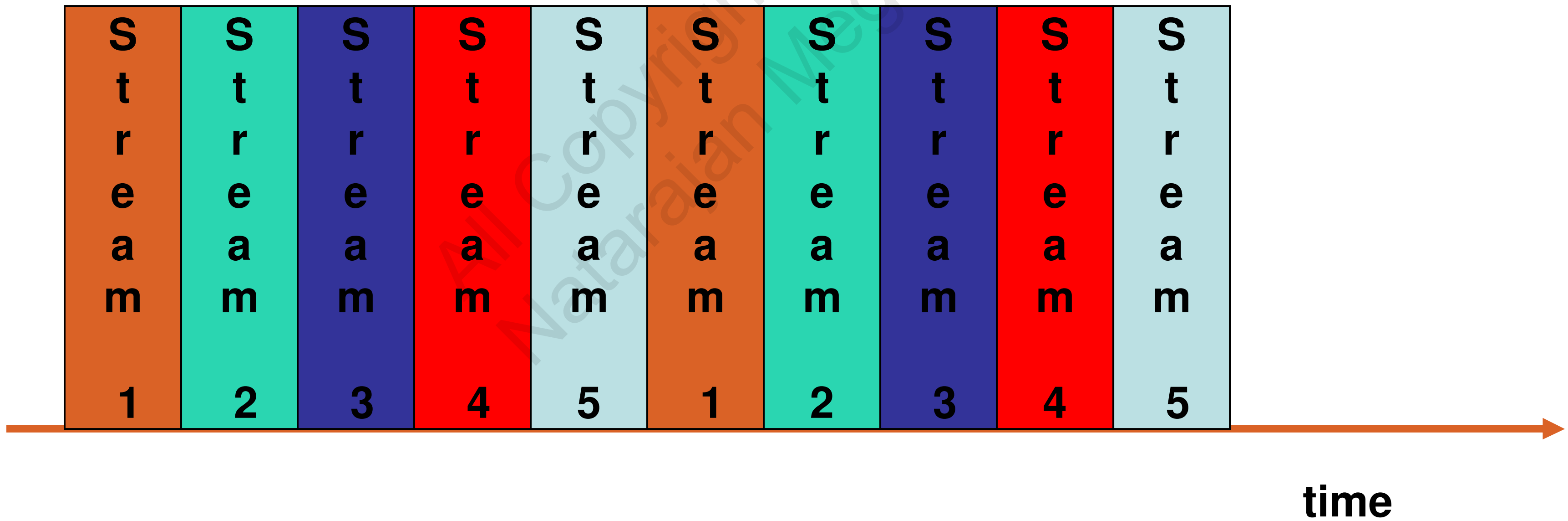
- **Spread spectrum multiplexing**

  Transfer the same data using multiple signals at the same frequency or different frequency.

  Goal: To provide reliability if certain carriers encounter interference.

Time Division Multiplexing (TDM)

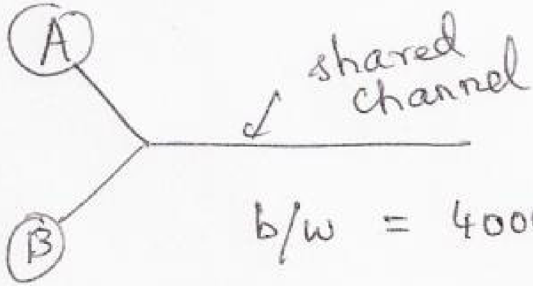
- Use a single carrier to carry the data of multiple transmitter/receiver pairs in a round-robin fashion.



Sample Question: TDM

- Assume two computers using time division multiplexing to take turns in sending 512 byte packets over a shared channel that operates at 40000 bits/second. If the hardware takes 10 msec after one computer stops sending before the other can begin, how long it will take for each of the two computers to send 1 MB of a file?

TDM Solution



b/w = 40000 bits/second

Packet size = 512 bytes.

Transmission delay per packet = $\frac{\text{Packet size}}{\text{b/w}} = \frac{512 \text{ bytes}}{40000 \text{ bits/sec}}$

$= \frac{512 \times 8}{40000} = 0.1024 \text{ sec}$

Hardware switching delay = 10 msec = 0.01 sec

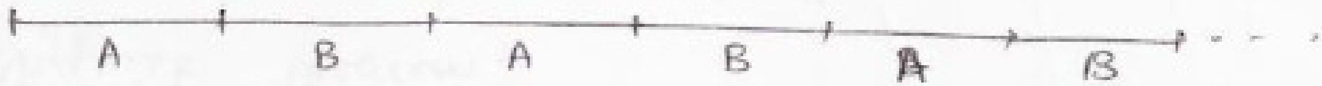
Transfer ~~Total~~ delay per packet per node = Transmission delay per packet + Hardware switching delay

$= 0.1024 + 0.01$

$= 0.1124 \text{ sec}$

TDM Solution (continued…)

Timeline



Node A spends 0.1124 sec in transmitting a packet and another 0.1124 sec in waiting for the next turn.

Hence, total delay per packet per node

= Transfer delay per packet + waiting time for another turn

= 0.1124 + 0.1124 = 0.2248 sec

No. of packets to be sent per node = $\frac{\text{File size}}{\text{Packet size}}$

$= \frac{1024 \times 1024 \text{ bytes}}{512 \text{ bytes/packet}}$

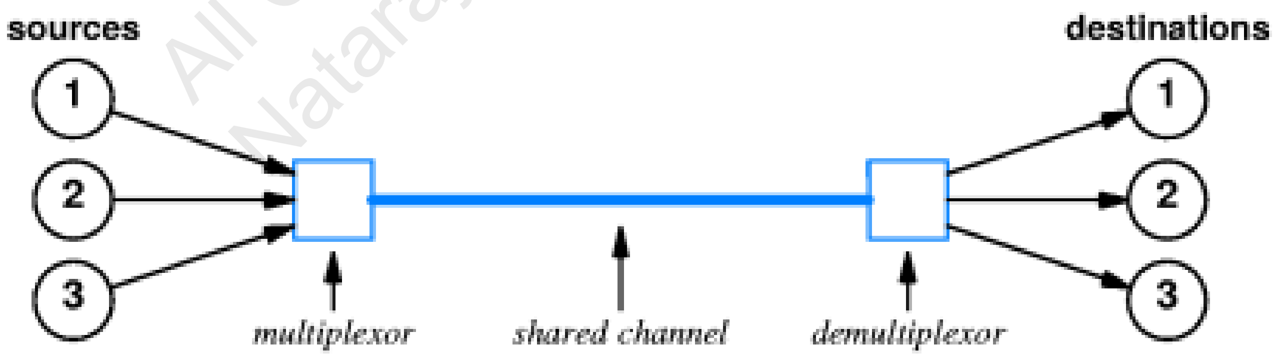
= 2048 packets

∴ Delay to transfer the whole file per node

= 2048 × 0.2248 = 460 sec

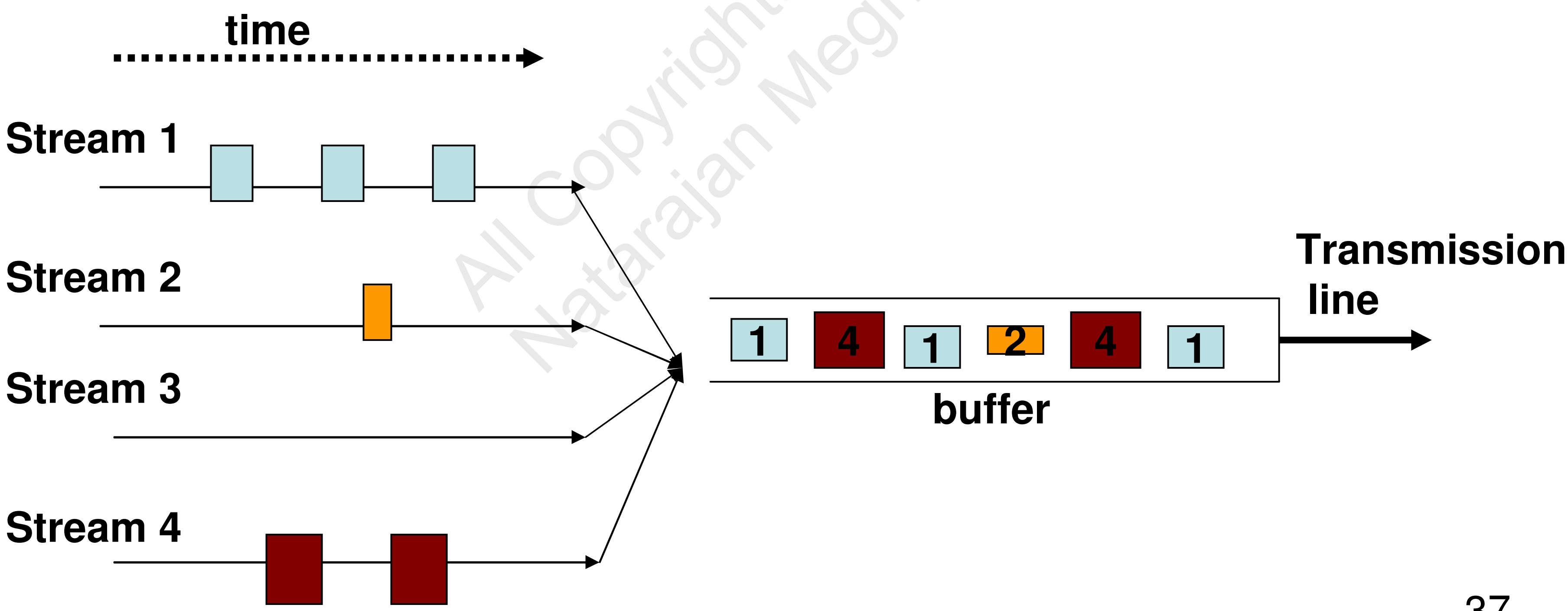
Frequency Division Multiplexing (FDM)

- FDM – technique of using multiple carrier frequencies to allow independent signals to travel through a medium.
- To avoid interference,
  - A minimum separation between the frequencies of the carriers is needed.
  - The carrier frequencies should not be multiples of each other. Hence, FDM is used only on high-bandwidth transmission systems.



Statistical Multiplexing

- Store the arriving packets in a buffer and send them one at a time in First-in First-out (FIFO) fashion.
- Most effective way of using the available bandwidth compared to TDM/ FDM as the user generating no packets is not allotted any of the channel resources



Code Division Multiplexing (CDM)

- CDM used in parts of the cellular telephone system and for some satellite communication
  - The specific version of CDM used in cell phones is known as Code Division Multi-Access (CDMA)
- CDM does not rely on physical properties
  - such as frequency or time
- CDM relies on an interesting mathematical idea
  - values from orthogonal vector spaces can be combined and separated without interference
- Each sender is assigned a unique binary code C_i
  - that is known as a chip sequence
  - chip sequences are selected to be orthogonal vectors
  - (i.e., the dot product of any two chip sequences is zero)
- At any point in time, each sender has a value to transmit, V_i
  - The senders each multiply $C_i \times V_i$ and transmit the results
- The senders transmit at the same time
  - and the values are added together
- To extract value V_i, a receiver multiplies the sum by C_i

CDM Example - Question

- Sample Question: Use Code Division Multiplexing to transmit the data values 0 1 1 0 and 1 0 1 0 from two senders S1 and S2 respectively. The 2-bit Chip Sequence of S1 and S2 are 0 1 and 1 1 respectively.
  - Show the resulting signal values when the two above two data signals are transmitted simultaneously
  - Show how the receiver for the data signals sent by S1 and S2 are able to decode the data?

CDM Example - Solution

- Let the Senders be represented as S1 and S2.
- Data D1 = [0 1 1 0] and D2 = [1 0 1 0]
- Chip Sequence C1 = [0 1] and C2 = [1 1]
- In all our computations, we will replace ‘0’ with ‘-1’ and then revert back to 0 at the end of the computation.
- Accordingly, D1 = [-1 1 1 -1]; D2 = [1 -1 1 -1]; C1 = [-1 1]; C2 = [1 1]

- Initial Step: To check if the chip sequences are orthogonal
- C1*C2 = [-1 1] * [1 1] = (-1)(1) + (1)(1) = -1 + 1 = 0
- Since the product is 0, we say C1 and C2 are orthogonal.
- Encoding at S1: The signal transmitted by the sender S1 is
- $S1 = C1^T * D1$

$$\begin{bmatrix} -1 \\ 1 \end{bmatrix}_{2x1} * \begin{bmatrix} -1 & 1 & 1 & -1 \end{bmatrix}_{1x4} = \begin{bmatrix} 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & -1 \end{bmatrix}_{2x4}$$

CDM Example – Solution (contd..)

- Encoding at S2: The signal transmitted by the sender S2 is
- $S2 = C2^T * D2$

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix}_{2x1} * \begin{bmatrix} 1 & -1 & 1 & -1 \end{bmatrix}_{1x4} = \begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 \end{bmatrix}_{2x4}$$

- Composite Signal Received at each Destination, S = S1 + S2

$$S = \begin{bmatrix} 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & -1 \end{bmatrix}_{2x4} + \begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 2 & -2 & 0 & 0 \\ 0 & 0 & 2 & -2 \end{bmatrix}_{2x4}$$

- Retrieving the Data at the Receiver for S1: = C1 * S

$$= \begin{bmatrix} -1 & 1 \end{bmatrix}_{1x2} * \begin{bmatrix} 2 & -2 & 0 & 0 \\ 0 & 0 & 2 & -2 \end{bmatrix}_{2x4} = \begin{bmatrix} -2 & 2 & 2 & -2 \end{bmatrix}_{1x4}$$

Dividing by 2, the data received from S1 = [-1 1 1 -1] ➔ [0 1 1 0]

CDM Example – Solution (contd..)

- Retrieving the Data at the Receiver for S2: = C2 * S

$$= [1 \quad 1]_{1x2} \quad * \quad \begin{bmatrix} 2 & -2 & 0 & 0 \\ 0 & 0 & 2 & -2 \end{bmatrix}_{2x4} \quad = \quad [2 \quad -2 \quad 2 \quad -2]_{1x4}$$

Dividing by 2, the data received from S1 = [1 -1 1 -1] ➔ [1 0 1 0]