King Fahd University of Petroleum & Minerals Computer Engineering Dept

COE 540 –Computer Networks
Term 072
Dr. Ashraf S. Hasan Mahmoud
Rm 22-148-3
Ext. 1724
Email: ashraf@kfupm.edu.sa

Lecture Contents

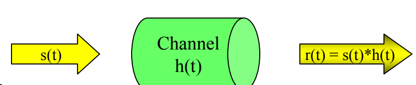
1. Channels and Models
2. Error Detection
3. ARQ: Retransmission Strategies
4. Framing
5. Standard DLCs

Reading Assignment #2

- You are required to read the following Sections:
  - 2.7, 2.8, 2.9 and 2.10 of Gallager's textbook
- The material is required for subsequent quizzes and exam

Channels and Models

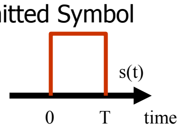
- Channels
  - Digital – accepts/generates bit stream
  - Analog – accepts waveforms
- Modem: a box that maps digital information into an analog waveform
- Conventionally,
  - s(t) – analog channel input
  - r(t) – analog channel output
    - Could be distorted, delayed, attenuated version of s(t)
- A good modulation/scheme maps the digital info into into s(t) such that the signal impairments are minimal!



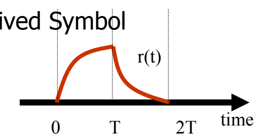
Filtering

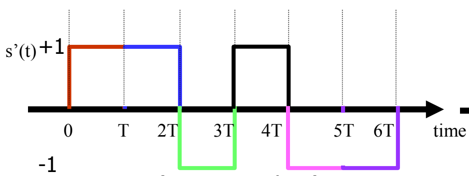
- The medium works as a filter – it has its own h(t)
- Properties of Linear-Time Invariant Filter:
  - If input s(t) yields output s(t), then for any τ, input s(t-τ) yields s(t-τ)
  - If s(t) yields r(t), then for any real number α, αs(t) yields αr(t), and
  - If s1(t) yields r1(t) and s2(t) yields r2(t), then s1(t)+s2(t) yields r1(t)+r2(t)

Transmitted Symbol



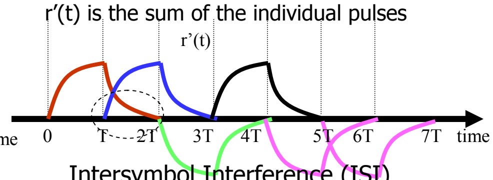
Received Symbol





(NRZ encoding)

r'(t) is the sum of the individual pulses



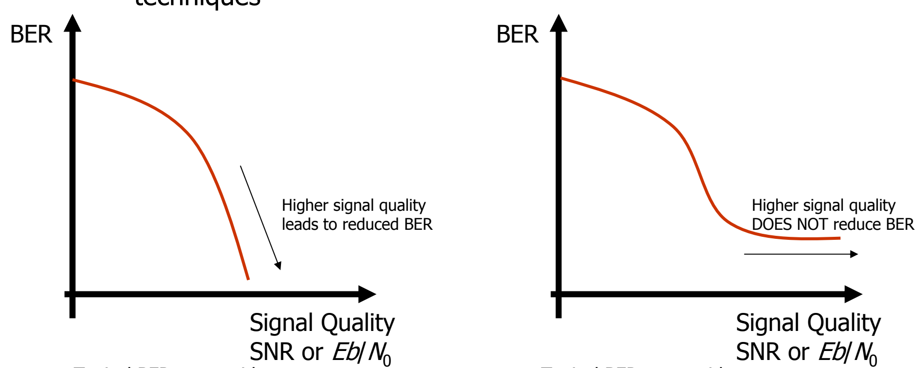
Intersymbol Interference (ISI)

Intersymbol Interference

- One symbol is being received while the tail(s) of the preceding symbols are not finished
  - A limit on channel bit rate
  - Irreducible error floor
- A similar phenomena appears if there are multiple delayed copies of the same single transmitted symbol
  - Multipath
  - A real-problem for high speed transmission over wireless links – Why?

Convolution Relation

- BER – a curve that determines the relation between signal power and bit error rate
  - Very important characterization tool for modulation/encoding techniques



Typical BER curve with no ISI or multipath

Typical BER curve with ISI or multipath (no equalization)

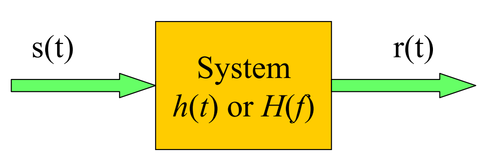
Convolution Integral

- **For linear Systems:**
  - **h(t) is the system's impulse response – i.e. r(t)= h(t) when s(t) = δ(t)**
  - **s(t) is system input signal**
  - **r(t) is system output signal**

$$r(t) = \int_{-\infty}^{\infty} s(\tau)h(t-\tau)d\tau$$

$$r(t) = s(t) * h(t)$$

$$R(f) = S(f)H(f)$$



convolution NOT multiplication

A good introduction into linear systems is found at
http://www.ece.utexas.edu/~bevans/courses/ee313/lectures/04_Convolution/lecture4.pdf

Example 1: Convolution

- If $h(t) = \alpha e^{-\alpha t}$ for $t > 0$
  $= 0$ otherwise

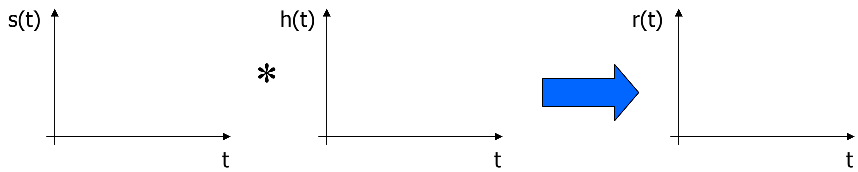
where $\alpha = 2/T$

A) Compute analytically and plot r(t) for $s(t) = \Pi((t-T/2)/T)$

B) Use Matlab to compute the required convolution – Plot the results and list your code

Hint: $\Pi(t/T)$ is the square pulse function of unit height, width equal to T, and centered around 0.

Solution:



Revision – Fourier Transform

- **A "transformation" between the time domain and the frequency domain**

Time (t)		**Frequency (f)**
s(t)	**← →**	**S(f)**

$$S(f) = \int_{-\infty}^{\infty} s(t) e^{-j2\pi ft} dt$$ Fourier Transform

$$s(t) = \int_{-\infty}^{\infty} S(f) e^{+j2\pi ft} df$$ Inverse Fourier Transform

Revision – Fourier Transform (2)

- **F.T. can be used to find the BANDWIDTH of a signal or system**
  - **Bandwidth - system: range of frequencies passed (perhaps scaled) by system**
  - **Bandwidth – signal: range of (+ve) frequencies contained in the signal**

Revision – Fourier Transform (3)

- **Remember for periodic signals (i.e. s(t) = s(t+T) where T is the period) ➔ Fourier Series expansion:**

$$s(t) = \frac{A_0}{2} + \sum_{n=1}^{\infty} \left[A_n \cos(2\pi n f_0 t) + B_n \sin(2\pi n f_0 t) \right]$$

$$A_0 = \frac{2}{T}\int_0^T s(t)dt \qquad B_n = \frac{2}{T}\int_0^T s(t)\sin(2\pi n f_0 t)dt$$

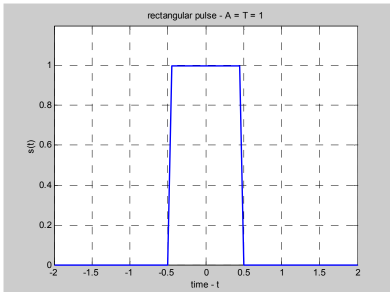
$$A_n = \frac{2}{T}\int_0^T s(t)\cos(2\pi n f_0 t)dt$$

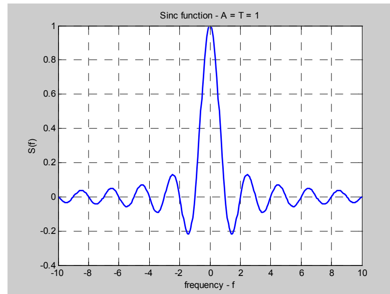
f_0 is the fundamental frequency and is equal to 1/T

Remember:
sinc(x) = sin(πx)/(πx)

Revision – Fourier Transform (4-a)

- **Famous pairs – rectangular pulse (A =T = 1)**





$$s(t) = \prod(t/T)$$

$$S(f) = AT\frac{\sin(\pi fT)}{\pi fT}$$

$S(f) = AT$ for $f = 0$
$= 0$ for $f = n/T$; n=+/- 1,2, …

Revision – Fourier Transform (4-b)

- **Famous pairs – sinc pulse (A =T = 1)**
- **The plots for the s(t) and the corresponding S(f) are the blue curves on the next slide**
- **The sinc pulse is a special case of the raised cosine pulse!**
- **Note T = 1/W**

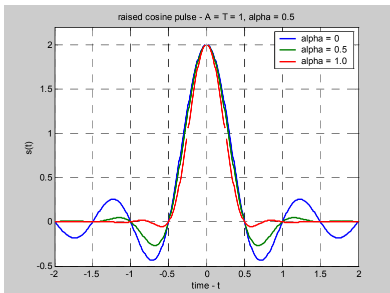
$$s(t) = A\frac{\sin\left(\pi Wt\right)}{\left(\pi Wt\right)}$$

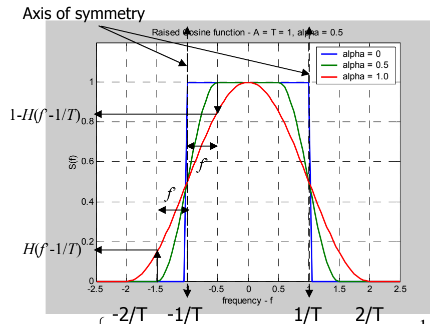
$$S(f) = \frac{A}{W}\Pi\left(f/W\right)$$

$S(f) = A/W$ for $|f| <= W/2$
$= 0$ for $|f| > W/2$

Revision – Fourier Transform (5)

- **Famous pairs – Raised Cosine pulse (A =T = 1), as a function of α**





$$s(t) = \frac{(2A)}{T}\frac{\cos(2\pi\alpha t)}{1-(4\alpha t)^2}\frac{\sin(2\pi t/T)}{2\pi t/T}$$

$$S(f) = \begin{cases} A & |f| < \frac{1}{T} - \alpha \\ A\cos^2\left(\frac{\pi}{4\alpha}\left(|f| - \frac{1}{T} + \alpha\right)\right) & \frac{1}{T} - \alpha < |f| < \frac{1}{T} + \alpha \\ 0 & |f| > \frac{1}{T} + \alpha \end{cases}$$

Revision – Fourier Transform (6)

- **Raised Cosine Pulse: 0 < α < 1/T**
- **Note that s(t) = 0 for t = nT/2 where n = +/- 1,2, …**
  - **Very good for forming pulses**
  - **ZERO ISI for ideal situation**
- **BW for s(t) = 1/T + α**
  - **Maximum = 2 X 1/T (for α = 1/T)**
  - **Minimum = 1/T (for α = 0)**

Revision – Fourier Transform (7)

- **Matlab code: Raised Cosine Pulse**

```
clear all  % clear all variables

A = 1;
T = 1;
alphas = [0 0.5 1];

for k = 1:length(alphas)
    alpha = alphas(k);

    t = -2:0.01:2;          % define the time axis
    s_t(k,:) = ((2*A)/T) * (cos(2*pi*alpha*t)./ …
               (1-(4*alpha*t).^2)) .*(sin(2*pi*t/T)./ …
               (2*pi*t/T));   % define s(t)

    f = -2.5:0.05:2.5;         % define the freq axis
    S_f(k,:) = zeros(size(f));
    i  = find(abs(f) <= (1/T-alpha));
    S_f(k,i) = A;
    i  = find((abs(f) <= (1/T+alpha)) & …
              (abs(f) > (1/T-alpha)));
    S_f(k,i) = A*(cos(pi/(4*alpha)* …
              (abs(f(i))-1/T+alpha))).^2;% define S(f)
end
```

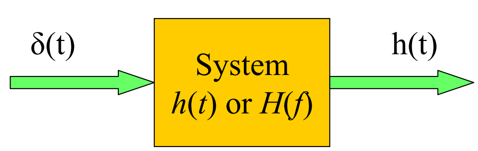
```
figure(1);
plot(t, s_t);            % plot s(t)
title('raised cosine pulse - A = T = 1');
xlabel('time - t');
ylabel('s(t)');
legend('alpha = 0', 'alpha = 0.5', 'alpha = 1.0');
axis([-2 2 -0.5 2.2]);
grid

figure(2);
plot(f, S_f);            % plot S(f)
title('Raised Cosine function - A = T = 1');
xlabel('frequency - f');
ylabel('S(f)');
legend('alpha = 0', 'alpha = 0.5', 'alpha = 1.0');
axis([-2.5 2.5 0 1.2]);
grid
```


Frequency Response

- **H(f) is known as the frequency response of the channel or system**
- **h(t) is known as the impulse response of the channel or system**

$$h(t) = \int_{-\infty}^{\infty} \delta(\tau) h(t-\tau) d\tau$$



$$h(t) = \delta(t) * h(t)$$

$$H(f) = \Delta(f) H(f)$$

This means $\Delta(f) = 1 \ \forall f$

Example 2: Frequency Response

A) For s(t) = Π(t/T), compute S(f) – Use Matlab to plot |S(f)|
B) For h(t) = $\alpha e^{-\alpha t}$ for t > 0 and equal to 0 otherwise, compute H(f) – Use Matlab to plot |H(f)|

Hint: (A) is solved on slide 13 – Part (B)'s answer is in the textbook equation (2.3). For these two parts you have to be able to derive the results.

Solution:

Sampling Theorem

- **<u>Theorem</u>: if a waveform s(t) is low-pass limited to frequencies at most W (i.e. S(f) = 0 for |f| > W], then s(t) is completely determined by its values each 1/(2W) seconds**
- **One can write**

$$s(t) = \sum_{i=-\infty}^{\infty} s\left(\frac{i}{2W}\right) \frac{\sin\left[2\pi W\left(t - i/(2W)\right)\right]}{2\pi W\left(t - i/(2W)\right)}$$

More on Sinc and Raised Cosine Pulses

- Consider the sinc pulse and the raised cosine pulse shown on slides 14 and 15
- Both of these s(t)s (the ideal sinc function and the raised cosine function) satisfies Nyquist criterion – i.e. zero ISI
  - i.e. $s(i/(2W)) = 0 \ \forall \ i \neq 0$
- However, raised cosine is a more "practical pulse" – can be easily generated in the lab!
- Figure 2.6 (Gallager) – shows that s(t) is equal to weighted shifted copies of the sinc function – graphical representation of the sampling theorem

More on Sinc and Raised Cosine Pulses – cont'd

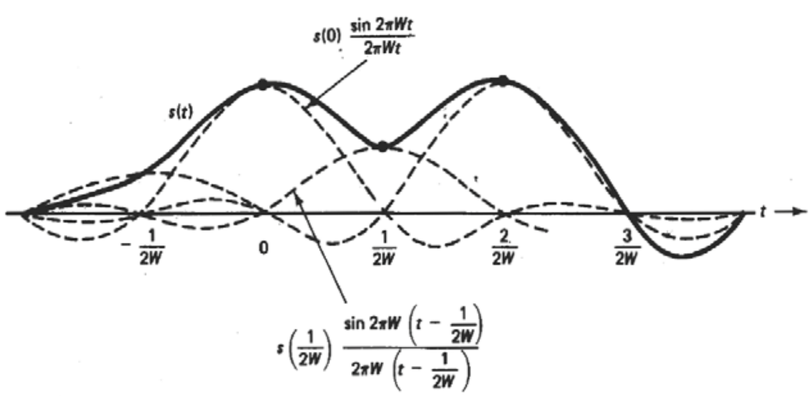


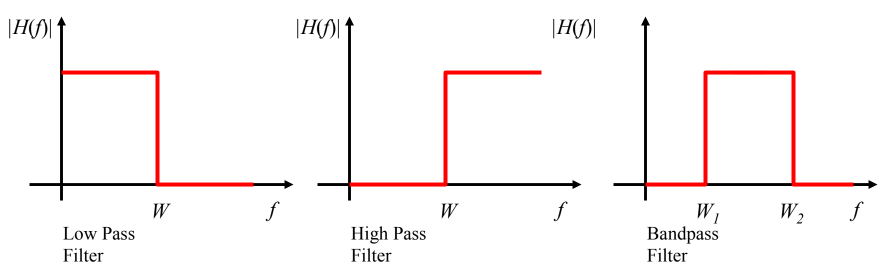
Figure 2.6 Sampling theorem, showing a function $s(t)$ that is low-pass limited to frequencies at most W. The function is represented as a superposition of $(\sin x)/x$ functions. For each sample, there is one such function, centered at the sample and with a scale factor equal to the sample value.

Bandpass Channels

- **Definition: ?**
- **This means** $H(f) = \int_{-\infty}^{\infty} h(t)dt = 0$
- **The impulse response for these channels fluctuates around 0 – i.e. +ve area = -ve area**
- **This phenomenon is called "ringing"**
- **NRZ is not appropriate for bandpass channels**
  - **Manchester encoding is a better option**
- **Another way of looking at this: NRZ has a DC component which DOES NOT pass through the bandpass channel**

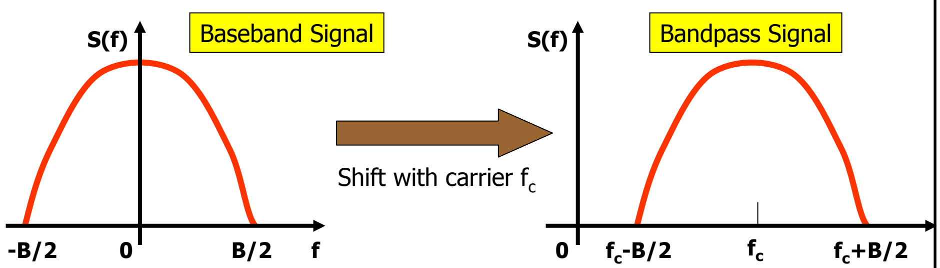
Signals and Systems

- **System bandwidth is determined by examining the Fourier transfer of the system function h(t), H(f)**
- **Example (transmission) systems:**



Baseband vs. Bandband

- **Baseband Signal:**
  - **Spectrum not centered around non zero frequency**
  - **May have a DC component**
- **Bandpass Signal:**
  - **Does not have a DC component**
  - **Finite bandwidth around or at f_c**



Modulation

- **Is used to shift the frequency content of a baseband signal**
  - **Basis for AM modulation**
  - **Basis for Frequency Division Multiplexing (FDM)**

Analog Communications

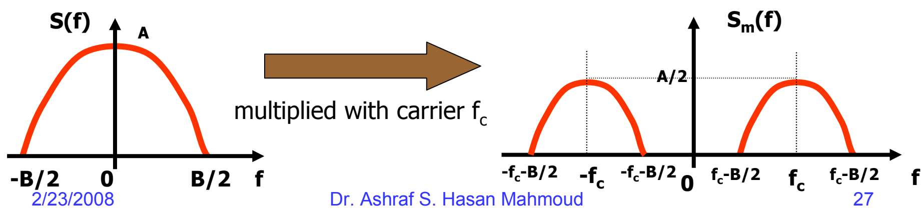
Modulation

- **Consider the signal s(t),**

$$s_m(t) = s(t) \times \cos(2\pi ft)$$

The spectrum for $s_m(t)$ is given by

$$S_m(f) = ½ \times \{S(f-f_c) + S(f+f_c)\}$$

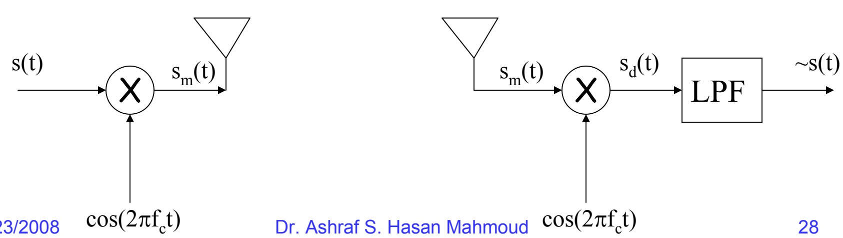


Analog Communications

Modulation – Txer/Rxer

- **At the receiver side:**

$$\begin{aligned} s_d(t) &= s_m(t) \times \cos(2\pi f_c t) \\ &= s(t) \times \cos(2\pi f_c t) \times \cos(2\pi f_c t) \\ &= \underbrace{½\, s(t)}_{\text{desired term}} + \underbrace{½\, s(t) \times \cos(2\pi 2 \times f_c t)}_{\text{undesired term – signal centered around } 2f_c \text{ filtered out using the LPF}} \end{aligned}$$



Nyquist Bandwidth

Digital Communications

- For a noiseless channels of bandwidth B, the maximum attainable bit rate (or capacity) is given by

$$C = 2B \log_2(M)$$

Where M is the size of the signaling set

Shannon Capacity

Digital Communications

- Capacity of a channel of bandwidth B, in the presence of noise is given by

$$C = B \log_2(1 + SNR)$$

where SNR is the ratio of signal power to noise power – a measure of the signal quality

Digital Communications

Example 3: Shannon Capacity

- **Consider a GSM system with BW = 200 kHz. If SNR is equal to 15 dB, find the channel capacity?**
- **Solution:**

SNR = 15 dB = 10^(15/10) = 31.6

C = $200X10^3$ X $\log_2(1+31.6)$

= 1005.6 kb/s

Note GSM operates at 273 kb/s which is ~27% of maximum capacity at SNR = 30 dB.

Digital Communications

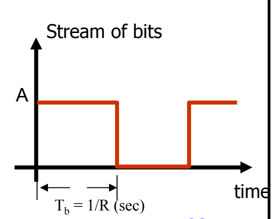
Eb/No Expression

- **An alternative representation of SNR**
- **Consider the bit stream shown in figure – for bit of rate R, then each bit duration is equal to T_b = 1/R seconds**
- **Energy of signal for the bit duration is equal to A^2X T_b, where its power is equal to bit energy / T_b or A^2.**
- **Noise power is equal to N_0 X B (refer to thermal noise section)**
- **Hence, SNR is given by signal power / noise power or**

$$SNR = \frac{signalpower}{N_0 B} = \frac{E_b}{N_0} \times \frac{R}{B}$$

- **One can also write**

$$\left(\frac{E_b}{N_0}\right)_{dB} = SignalPower(dBW) - 10\log R - 10\log k - 10\log T$$

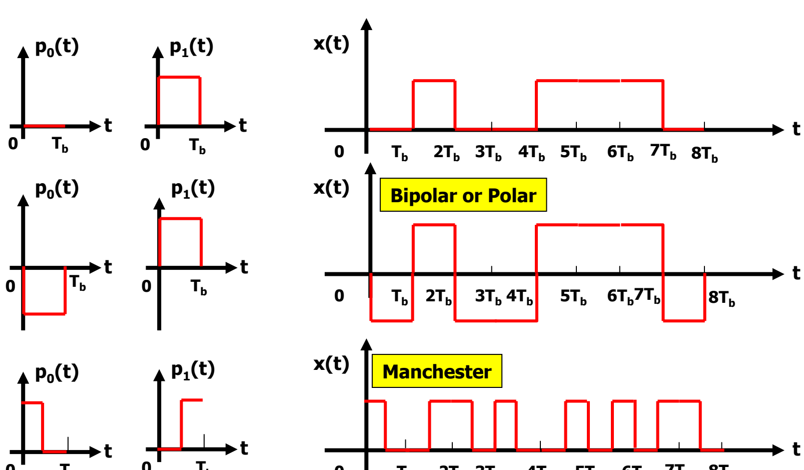


Signal Elements or Pulses

- Unit of transmission – repeated to form the overall signal
- *Shape* of pulse determines the bandwidth of the transmitted signal
- Digital data is mapped or encoded to the different pulses or units of transmission
- Baud/Modulation or Symbol Rate (R_s)
  - The bit rate $R_b = R_s \log_2(M)$
- Please refer to earlier examples of pulses and the corresponding BW

Signal Elements or Pulses

Definitions of Pulses **Encoded Signal: 0 1 0 0 1 1 1 0**

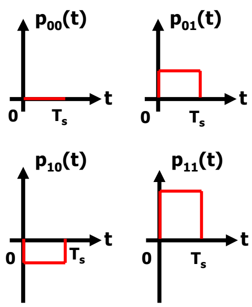


Examples of Digital Signaling

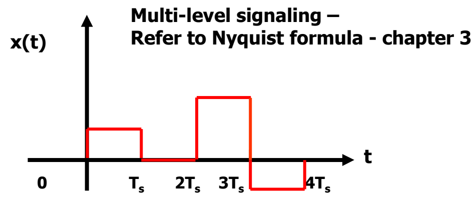
Digital Communications

Signal Elements or Pulses

Pluses Definitions



Encoded Signal: 0 1 0 0 1 1 1 0



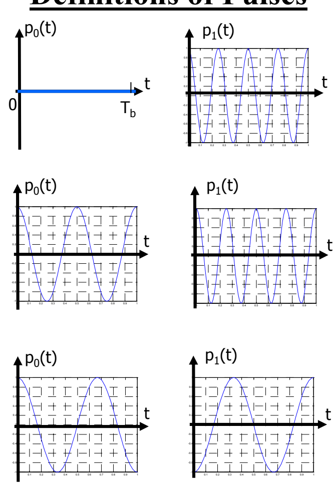
- Note that each symbol or pulse caries 2 bits
- Symbol duration is $T_s = 2T_b$
- Bit rate R equal to $1/T_b$
- Symbol rate or *baud rate* R_s equal to $1/T_s$ ➔ $R = 2R_s$
- In general to encode n bits per pulse, you need 2^n pulses

Example of Digital Signaling

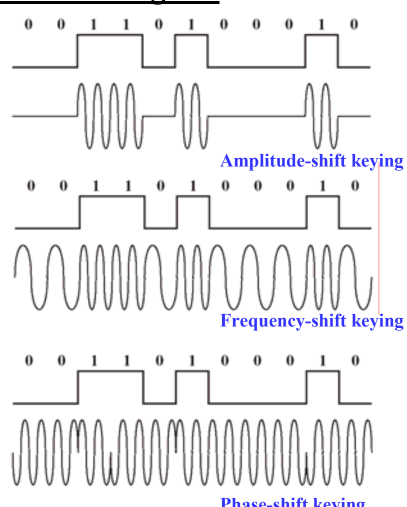
Digital Communications

Signal Elements or Pulses

Definitions of Pulses



Encoded Signal:

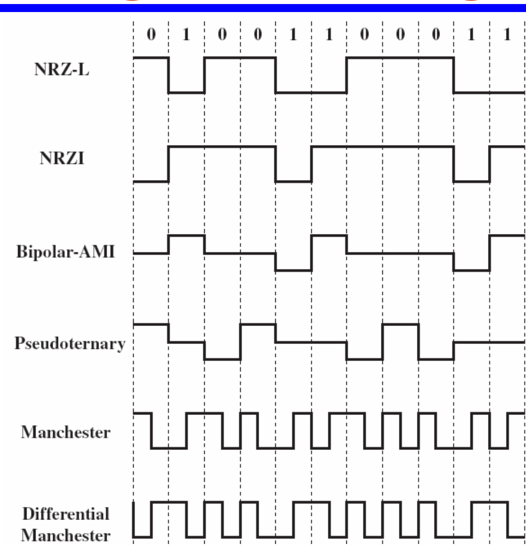


Example of Analog Signaling

Digital Signal Encoding Formats

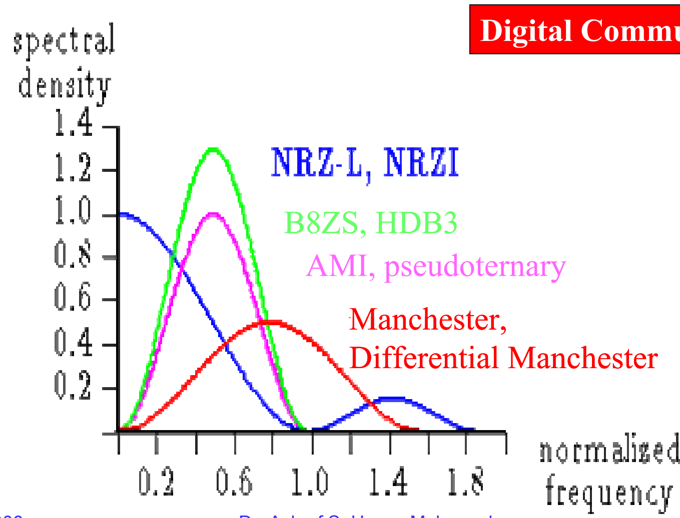
- Nonreturn to Zero-Level (NRZ-L)
  - 0 = high level
  - 1 = low level
- Nonreturn to Zero Inverted (NRZI)
  - 0 = no transition at beginning of interval
  - 1 = transition at beginning of interval
- Bipolar-AMI
  - 0 = no line signal
  - 1 = +ve or –ve level; alternating successive ones
- Pseudoternary
  - 0 = +ve or –ve level; alternating for successive ones
  - 1 = no line signal
- Doubinary
  - 0 = no line signal
  - 1 = +ve or –ve level; depending on number of separating 0s (even – same polarity, odd – opposite polarity)
- Manchester
  - 0 = transition from high to low in middle of interval
  - 1 = transition from low to high in middle of interval
- Differential Manchester: Always transition in middle of interval
  - 0 = transition at beginning of interval
  - 1 = no transition at beginning of interval

Digital Signal Encoding Formats



Spectrum Characteristics of Digital Encoding Schemes

Digital Communications



Digital Data – Analog Signals

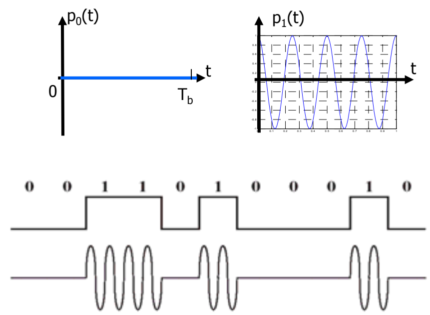
- **Digital data (bits) transmitted using analog signals:**
  - **E.g. computer-modem-PSTN**
- **Subscriber–to–PSTN connection designed to carry analog (voice) signal from 300 Hz to 3400 Hz**
- **56K Modem – encodes data and generates a signal occupying the same range for voice signals ➔ one line - one signal**
- **DSL Modem – encodes data and generates signal occupying higher range than that usually occupied by voice ➔ one line – two signals**

Amplitude Shift Keying (ASK)

- **Analog pulses (signal elements) used are:**

$$s(t) = \begin{cases} A\cos(2\pi f_c t) & bit = 1 \\ 0 & bit = 0 \end{cases}$$

- **Spectrum of overall signal is centered around f_c**
- **Application: on voice-grade lines used up to 1200 bps**



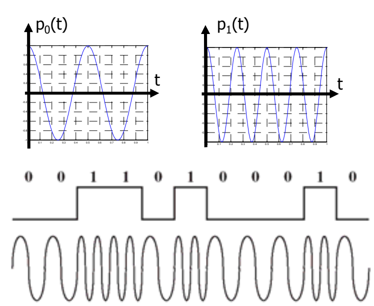
This is called BASK

Frequency Shift Keying (FSK)

- **Analog pulses (signal elements) used are:**

$$s(t) = \begin{cases} A\cos(2\pi f_1 t) & bit = 1 \\ A\cos(2\pi f_2 t) & bit = 0 \end{cases}$$

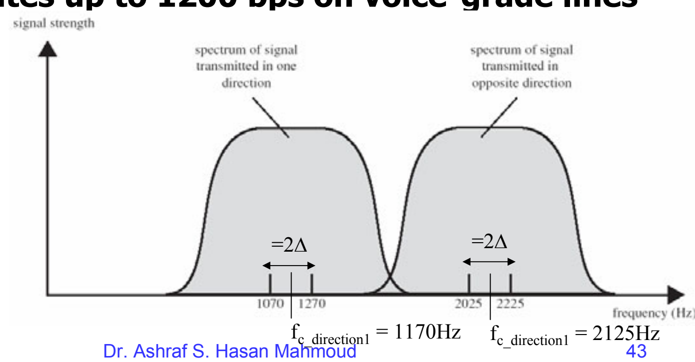
- **Spectrum of overall signal is centered around f_1 and f_2**



This is called BFSK

Frequency Shift Keying (FSK) (2)

- **Application: full duplex**
  - **Direction 1: f1 = 1070 Hz, f2 = 1270 Hz**
  - **Direction 2: f1 = 2025 Hz, f2 = 2225 Hz**
- **Less susceptible to errors (compared to ASK) – used for rates up to 1200 bps on voice-grade lines**
  - **Also used for high frequency (3 to 30 MHz) radio transmission**
  - **LANs – coaxial cables**



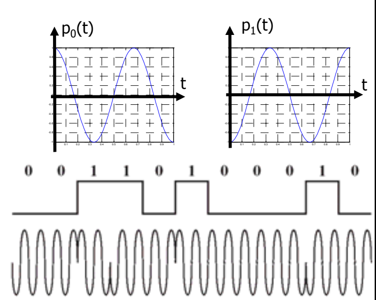
Phase Shift Keying (PSK)

- **Analog pulses (signal elements) used are:**

$$s(t) = \begin{cases} A\cos(2\pi f_c t + \pi) & bit = 1 \\ A\cos(2\pi f_c t) & bit = 0 \end{cases}$$

- **Spectrum of overall signal is centered around f_c**

- **Example of 2-phase (binary) system**



This is called BPSK

Multi-Level ASK

- **ASK is also known as digital PAM – refer to PAM used for PCM encoding**
- **The transmitted symbols:**

$$s_i(t) = A_i\cos(2\pi f_c t), \quad i = 1, 2, \ldots, M \qquad 0 \le t \le T_s$$

where

$A_i = (2i-1-M)d, \quad i = 1, 2, \ldots, M$

2d is distance between adjacent signal amplitudes

M is number of different signal elements (the alphabet size)= 2^L

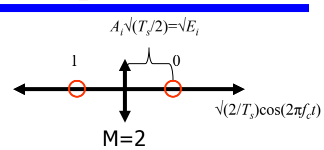
L is number of bits per signal element or symbol

T_s is the symbols duration.

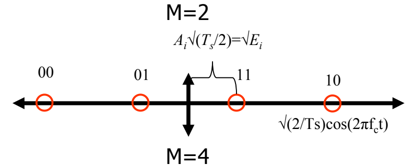
- **The energy for $s_i(t)$, E_i, is given by $A_i^2 T_s/2$**

Multi-Level ASK – Examples

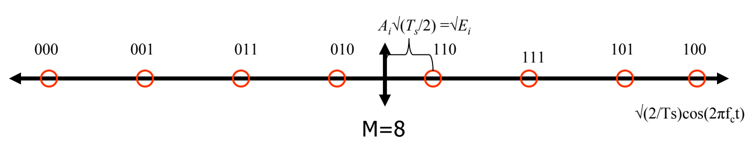
- **Examples:**
- **M = 2 – Binary ASK**
  A1 = -d, A2 = d



- **M = 4 – 4-level ASK**
  A1 = -3d, A2 = -d, A3 = d, A4 = 3d



- **M = 8 – 8 level ASK**
  A1 = -7d, A2 = -5d, A3 = -3d, A4 = -d,
  A5 = d, A6 = 3d, A7 = 5d, A8 = 7d



Note the grey coding!
Adjacent symbols are different by 1 bit only.

Multi-Level PSK

- **The transmitted symbols:**

$$s_i(t) = A\cos(2\pi f_c t + \theta_i),\quad i = 1, 2, \ldots, M \qquad 0 \le t \le T_s$$
$$= A\{\cos(\theta_i)\cos(2\pi f_c t) - \sin(\theta_i)\sin(2\pi f_c t)\}$$

where

$\theta_i = 2\pi(i-1)/M$, $\quad i=1, 2, \ldots, M$.

M is number of different signal elements (the alphabet size)= 2^L

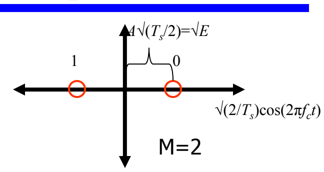
L is number of bits per signal element or symbol

T_s is the symbols duration.

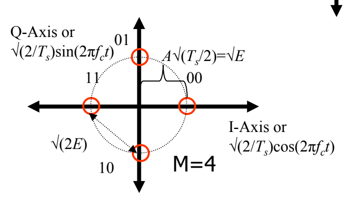
- **The energy for $s_i(t)$, E_i, is given by $A^2T_s/2$**
  - **i.e. all symbols have equal energy ➔ $E = A^2T_s/2$!!**

Multi-Level PSK - Examples

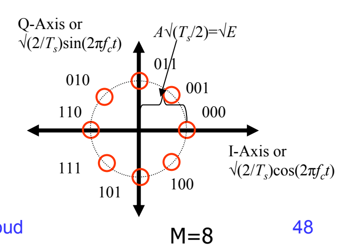
- **M = 2 – BPSK**
  $\theta1 = 0$, $\theta2 = \pi$



- **M = 4 – QPSK**
  $\theta1 = 0$, $\theta2 = \pi/2$,
  $\theta3 = \pi$, $\theta4 = 3\pi/2$,

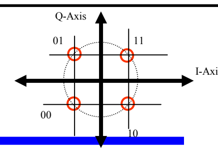


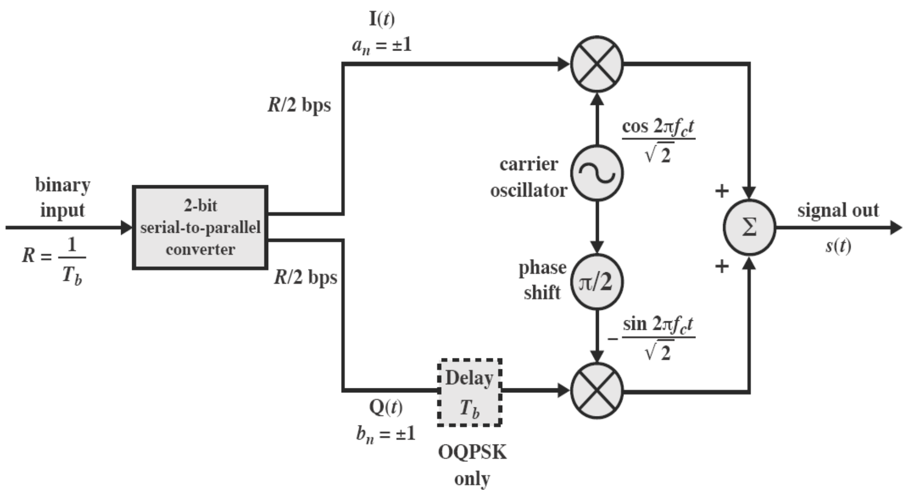
- **M = 8 – 8-PSK**
  $\theta1 = 0$, $\theta2 = \pi/4$, $\theta3 = \pi/2$, $\theta4 = 3\pi/4$,
  $\theta5 = \pi$, $\theta6 = 5\pi/4$, $\theta7 = 3\pi/2$, $\theta8 = 7\pi/4$



Note the grey coding!
Adjacent symbols are different by 1 bit only.

QPSK/OQPSK Modulator





Multi-Level FSK (MFSK)

- **Analog pulses (signal elements) used are:**

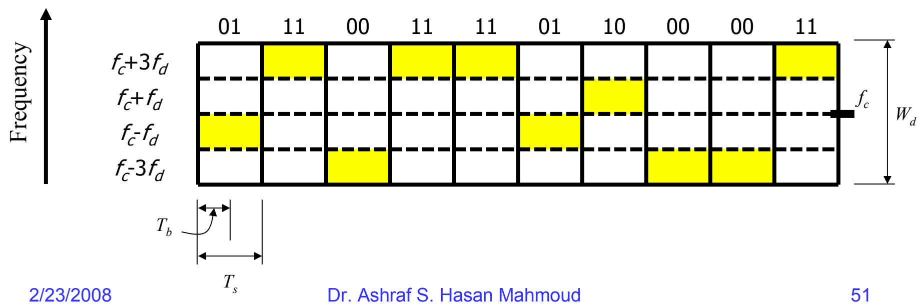
$$s_i(t) = A\cos(2\pi f_i t) \qquad 1 \le i \le M$$

- **Where**
  - **$f_i = f_c + (2i-1-M)f_d$**
  - **f_c: carrier frequency**
  - **f_d: the difference frequency**
  - **M: number of different signal elements (the alphabet size)= 2^L**
  - **L: number of bits per signal element or symbol**

MFSK Example – M = 4

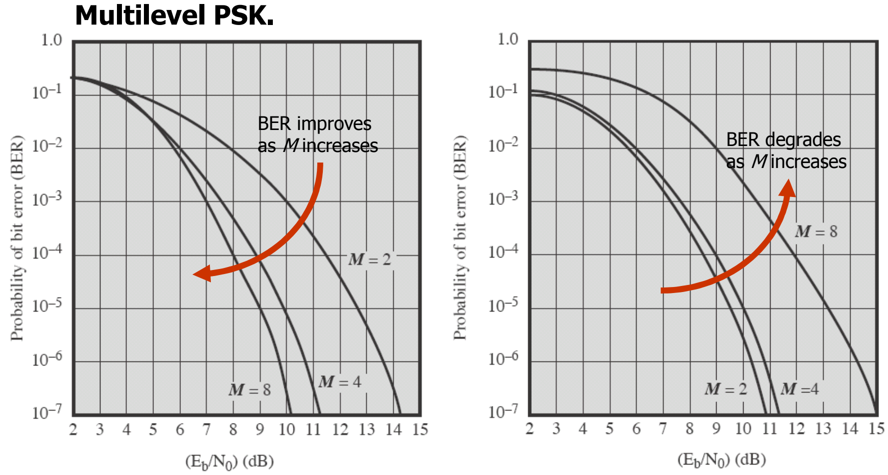
- **Example – M = 4**
  - **f1 = fc – 3fd → 00**
  - **f2 = fc – fd → 01**
  - **f3 = fc + fd → 10**
  - **f4 = fc + 3fd → 11**

Note this scheme does not utilize grey coding!!



Performance – cont'd

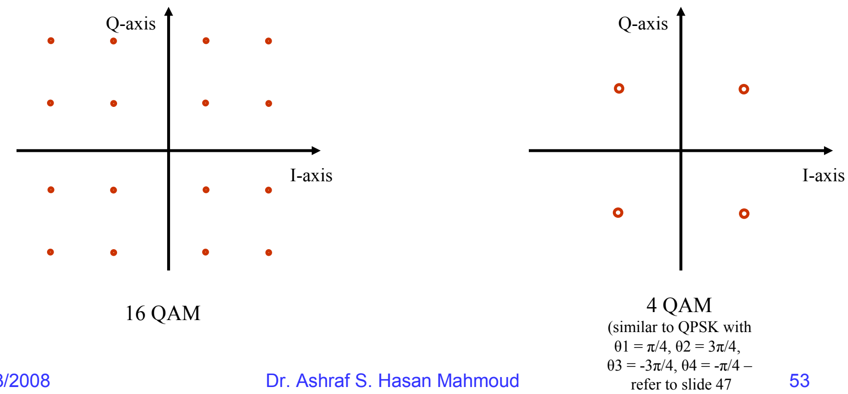
- **Theoretical bit error rate for (a) Multilevel FSK and (b) Multilevel PSK.**



(a) **(b)**

Quadrature Amplitude Modulation (QAM)

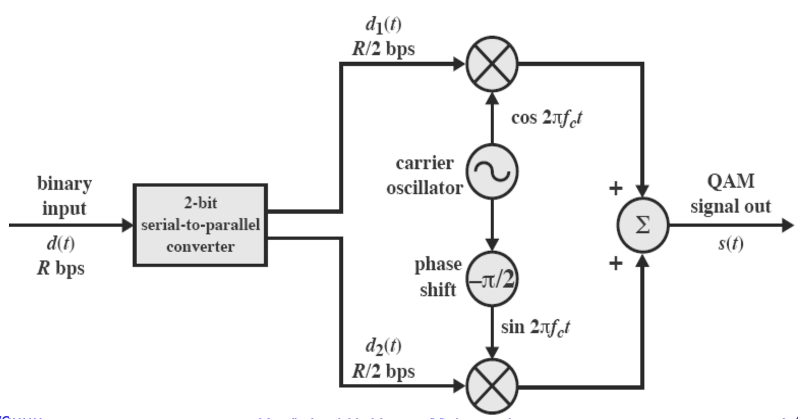
- Popular analog signaling technique – used in ADSL
- A combination of ASK and PSK
- Example signal constellations:



Quadrature Amplitude Modulation (QAM)

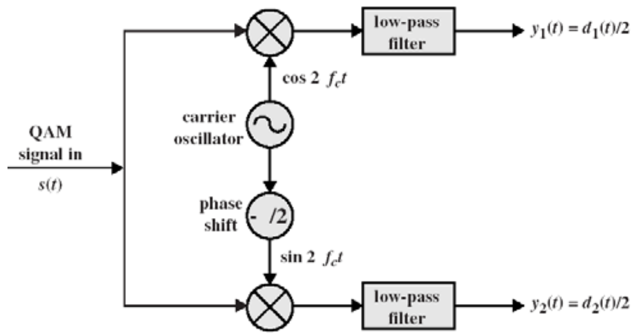
- Signal given by:

$$s(t) = d_1(t)\cos(2\pi f_c t) + d_2(t)\sin(2\pi f_c t)$$



Example 4: QAM

Problem: The figure below shows the QAM demodulator corresponding to the to the QAM modulator shown in previous slide. Show that this arrangement DOES recover the two signals d1(t) and d2(t), which can be combined to recover the original signal.



Example: QAM - Solution

<u>Solution</u>:

$$s(t) = d1(t)\cos(\omega_c t) + d2(t)\sin(\omega_c t)$$

Use the following identities:

$$\cos(2\alpha) = 2\cos^2(\alpha) – 1; \sin^2(\alpha) = 2\sin(\alpha)\cos(\alpha)$$

For upper branch:

$$s(t) \text{ X } \cos(\omega_c t) = d1(t)\cos(2\omega_c t) + d2(t)\sin(\omega_c t)\cos(\omega_c t)$$
$$= (1/2)d1(t) + (1/2)d1(t)\cos(2\omega_c t) + (1/2)d2(t)\sin(2\omega_c t)$$

Use the following identities:

$$\cos(2\alpha) = 1 – 2\sin^2(\alpha); \sin^2(\alpha) = 2\sin(\alpha)\cos(\alpha)$$

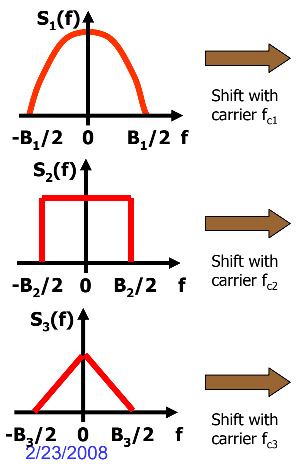
For lower branch:

$$s(t) \text{ X } \sin(\omega_c t) = d1(t)\cos(\omega_c t)\sin(\omega_c t) + d2(t)\sin(2\omega_c t)$$
$$= (1/2)d1(t)\sin(2\omega_c t) + (1/2)d2(t) - (1/2)d2(t)\cos(2\omega_c t)$$

All terms at $2\omega_c$ are filtered out by the low-pass filter, yielding:

$$y1(t) = (1/2)d1(t); \quad y2(t) = (1/2)d2(t)$$

Frequency Division Multiplexing (FDM)

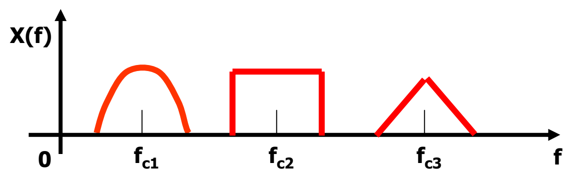


$x(t) = s_1(t) \times \cos(2\pi f_{c1}t) + s_2(t) \times \cos(2\pi f_{c2}t) + s_3(t) \times \cos(2\pi f_{c3}t)$

- x(t) is transmitted on the media

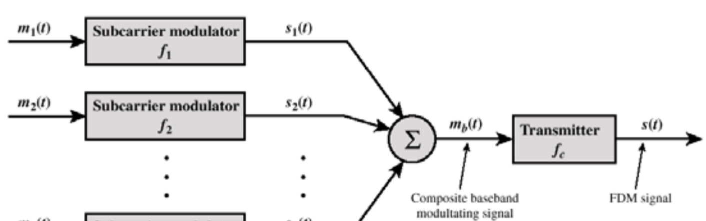
-The three spectra are not overlapping if f_{c1}, f_{c2}, and f_{c3} are chosen appropriately

-Original composite signals $s_1(t)$, s2(t), and s3(t) can be recovered using bandpass filters with appropriate bandwidths centered at f_{c1}, f_{c2}, and f_{c3}, respectively.

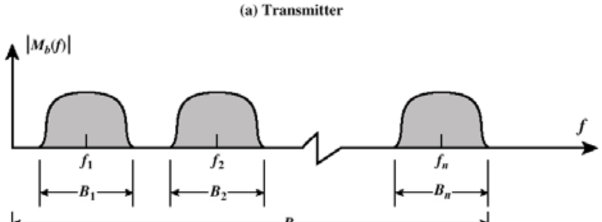


Frequency-Division Multiplexing - Transmitter

- $m_i(t)$: analog or digital information
- Modulated with subcarrier f_i ➔ $s_i(t)$
- $m_b(t)$ composite baseband modulating signal
- $m_b(t)$ modulated by f_c ➔The overall FDM signal s(t)



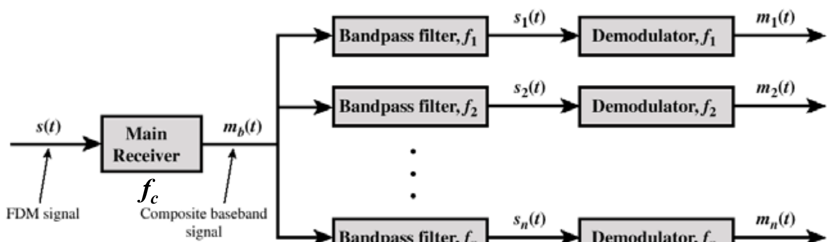
(a) Transmitter



Spectrum function of composite baseband modulating signal $m_b(t)$

Frequency-Division Multiplexing - Receiver

- $m_b(t)$ is retrieved by demodulating the FDM signal s(t) using carrier f_c
- $m_b(t)$ is passed through a parallel bank of bandpass filters – centered around f_i
- The output of the i^{th} filter is the i^{th} signal $s_i(t)$
- $m_i(t)$ is retrieved by demodulating $s_i(t)$ using subcarrier f_i



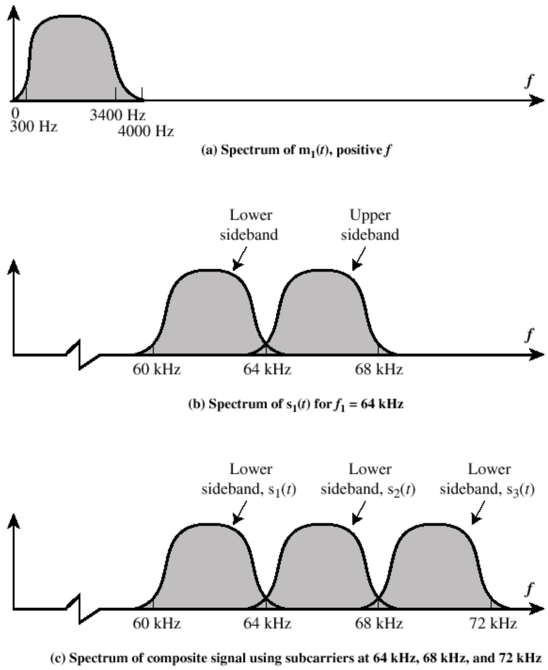
Frequency-Division Multiplexing – Example 5: Cable TV – cont'd

- Cable has BW ~ 500 MHz ➔ 10s of TV channels can be carried *simultaneously* using FDM
- Table: Cable Television Channel Frequency Allocation (partial): 61 channels occupying bandwidth up to 450 MHz

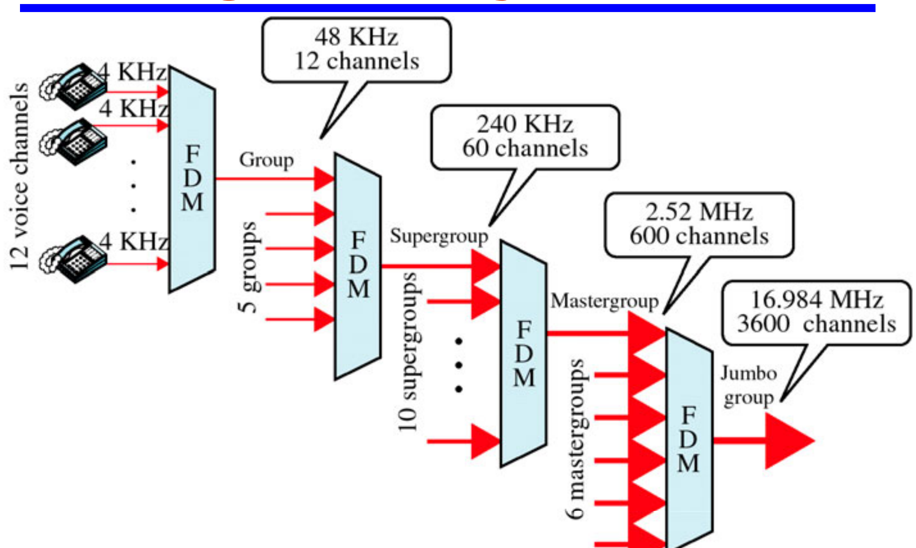
Channel No	Band (MHz)	Channel No	Band (MHz)	Channel No	Band (MHz)
2	54-60	22	168-174	42	330-336
3	60-66	23	216-222	43	336-342
4	66-72	24	222-234	44	342-348
5	76-82	…	…	…	…
6	82-88				
7	174-180				
8	180-186				
9	186-192				
10	192-198				
11	198-204				
12	204-210				
13	210-216				
FM	88-108				
14	120-126				
15	126-132				
16	…				
…	…				

Frequency-Division Multiplexing – Example 6: Voiceband Signals

- $m_1(t)$: voiceband signal – bandwidth = 4000 Hz
- When modulated by a carrier f_1 = 64 KHz ➔ two identical sidebands; overall bandwidth = 2X4KHz = 8 KHz
- Information of $m_1(t)$ is preserved if one of the sidebands is eliminated (filtered out) ➔ bandwidth of modulated signal = 4 KHz
- (c) shows spectrum for composite signal using three subcarriers

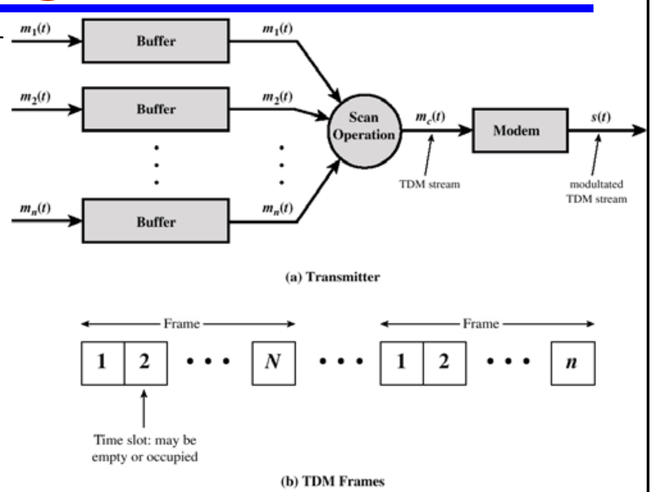


Frequency-Division Multiplexing – Analog Carrier Systems



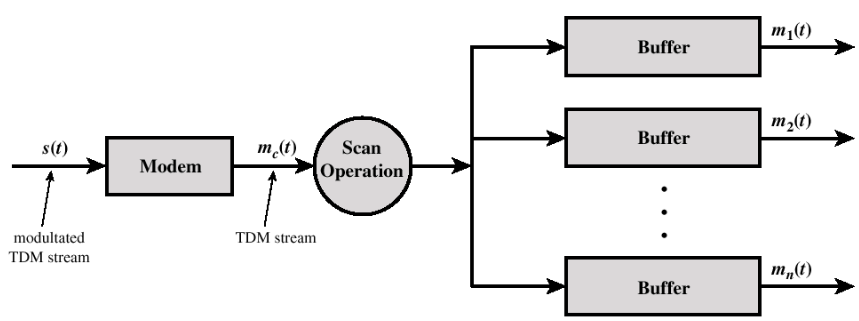
Synchronous Time-Division Multiplexing - Transmitter

- Digital sources $m_i(t)$ – usually buffered
- A scanner samples sources in a cyclic manner to form a frame
- $m_c(t)$ is the TDM stream or frame ➔ frame structure is fixed
- Frame $m_c(t)$ is then transmitted using a modem ➔ resulting analog signal is s(t)



Synchronous Time-Division Multiplexing - Receiver

- TDM signal s(t) is demodulated ➔ result is TDM digital frame $m_c(t)$
- $m_c(t)$ is then scanned into n parallel buffers;
- The i^{th} buffer correspond to the original $m_i(t)$ digital information



Synchronous Time-Division Multiplexing – Bit/Character Interleaving

- TDM frame: sequence of slots – fixed structure – NOTE: no header/error control for this frame
  - One or more slots per digital source
  - The order of the slots dictated by the scanner control
  - The slot length equals the transmitter buffer length:
    - Bit: bit interleaving
      - Used for synchronous sources – but can be used for asynchronous sources
    - Character: character-interleaving
      - Used for asynchronous sources
      - Start/stop bits removed at tx-er and re-inserted at rx-er
- Synchronous TDM: time slots are pre-assigned to sources and FIXED
  - If there is data, the slot is occupied
  - If there is no data, the slot is left unoccupied

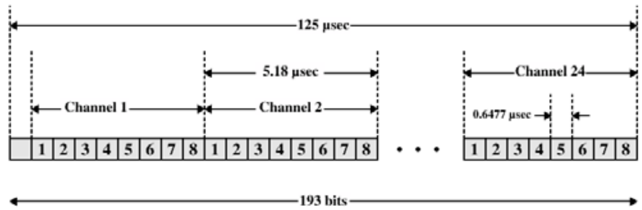
This is a cause of inefficiency!

TDM Link Control

- TDM frame:
  - No header and no error detection/control – these are per connection procedures
  - Frame synchronization is required – to identify beginning and end of frame
    - Added-digit framing: One control bit is added to each start of frame – all these bits from consecutive frame form an identifiable pattern (e.g. 1010101…)
    - These added bits for framing are inserted by system ➔ control channel
    - Frame search mode: Rx-er parses incoming stream until it recognizes the pattern ➔ then TDM frame is known
- Pulse stuffing:
  - Different sources may have separate/different clocks
  - Source rates may not be related by a simple rational number
  - <u>Solution</u>: inflate lower source rates by inserting extra dummy bits or pulses to mach the locally generated clock speed

TDM – Example 7: Digital Carrier Systems

- Voice call is PCM coded ➔ 8 b/sample
- DS-0: PCM digitized voice call – R = 64 Kb/s
- Group 24 digitized voice calls into one frame as shown in figure ➔DS-1: 24 DS-0s
- Note channel 1 has a digitized sample from 1st call; channel 2 has a digitized sample from 2nd calls; etc.



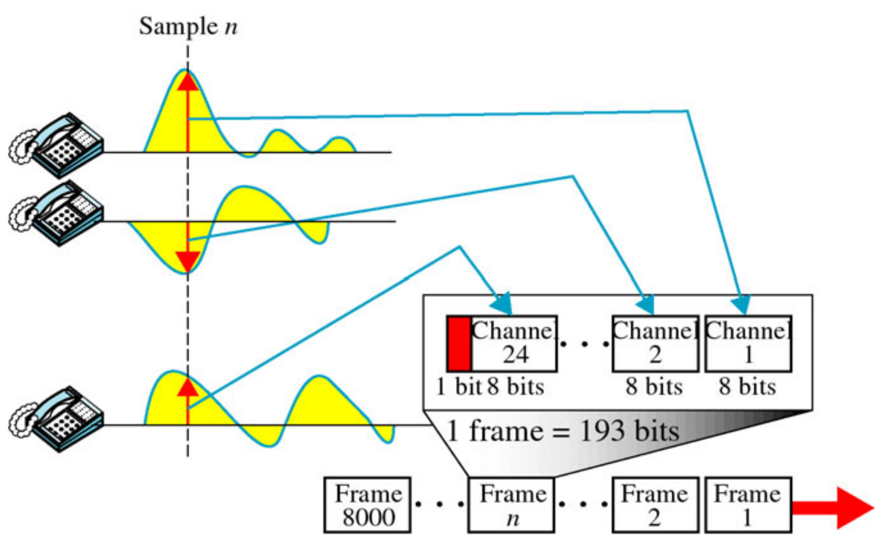
Notes:

1. The first bit is a framing bit, used for synchronization.
2. Voice channels:
   - •8-bit PCM used on five of six frames.
   - •7-bit PCM used on every sixth frame; bit 8 of each channel is a signaling bit.
3. Data channels:
   - •Channel 24 is used for signaling only in some schemes.
   - •Bits 1-7 used for 56 kbps service
   - •Bits 2-7 used for 9.6, 4.8, and 2.4 kbps service.

Figure 8.9 DS-1 Transmission Format

Figure 8-28

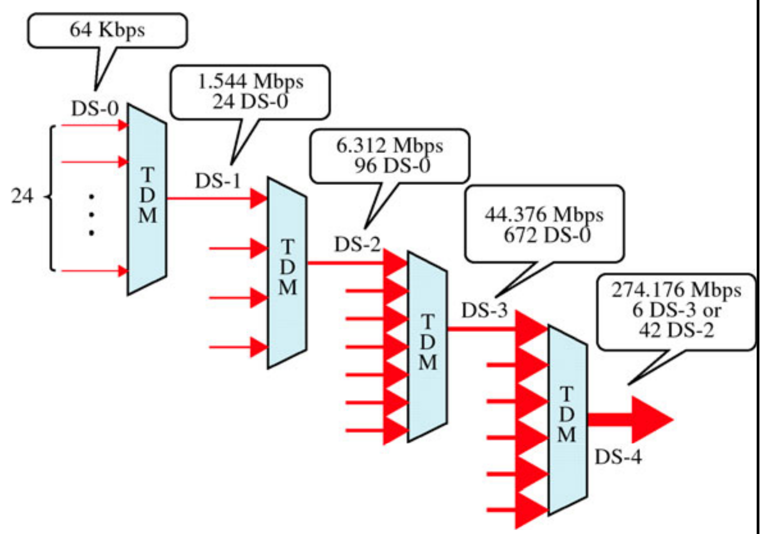
T-1 Frame



T-1 = 8000 frames/s = 8000 x 193 bps = 1.544 Mbps

TDM – Example 8: Digital Carrier Systems (2)

- TDM



Propagation Media

- Wired Media:
  - Twisted pair
  - Cable
  - Optical fiber
- Wireless Media – microwave links, satellite, etc.
- Signal attenuation – loss of power due to media resistance
  - Attenuation (dB) inversely proportional to distance
  - Trade-off: repeater (to extend distance) and Bit rate
- Refer to textbook for characteristics of TP, coaxial, optical, radio frequency communications

Error Detection

- Error control over links involves:
  - Error detection
  - Error correction
    - ARQ
    - FEC
- Remember – DLC responsibility is to provide an error-free reliable packet stream to the next layer up.
- Error detection depends on PARITY CHECK

Single Parity Checks

- One bit added to the "data" string → c bit
  - 1 if the number of 1's in the data string is odd
  - 0 if the number of 1's in the data string is even
- c is the sum, modulo 2, of the data string bits
- Example:
  - ASCII characters: 7 bits (code) + 1 parity bit

s_1	s_2	s_3	s_4	s_5	s_6	s_7	c
1	0	1	1	0	0	0	1

- Why type of errors does this scheme detect?
  - All odd number of errors – Does that depend on the length of the "data" string?
  - All even number of errors are not detected

How Appropriate Single Parity Checks?

- What "type" of errors are expected in communication generally?

VRC/LRC Parity Check

- Extension of simple parity: Vertical Redundancy Check (VRC) and Longitudinal Redundancy Check (LRC)

Original data to send

Parity check

Char 1	1	0	0	1	1	0	0	1
Char 2	0	1	1	1	0	1	0	0
Char 3	1	1	0	0	1	1	0	0
Char 4	1	0	0	0	1	0	0	0
Char 5	0	1	0	0	1	1	1	0
Checking char	1	1	1	0	0	1	1	1

VRC/LRC Parity Check (2)

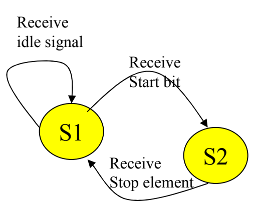
- Can detect all odd errors – same as the simple parity check
- Can detect any combination of even error in characters that DO NOT result in even number of errors in a column
- Excess Redundancy: 13/(35+13) =
- There could be undetected errors – How?

Linear Codes

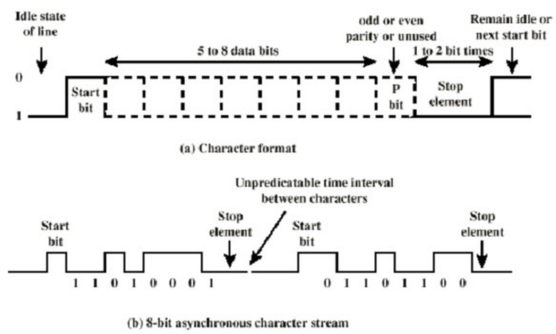
- Code: the mathematical transformation to generate the code word (data + parity check)
- Effectiveness of the code:
  - Minimum distance of the code – def = smallest number of errors that can convert one code word to another
  - The burst detecting capability – def = smallest integer B such that a code can detect all burst of length B or less
  - Probability of an undetected error ~ 2^{-L} (How? See textbook page 61)
- If a code a minimum distance of d → then the code can be used to correct any combination of fewer than d/2 error (textbook problem 2.10).

Asynchronous Transmission

- Simple / Cheap
- Efficiency: transmit 1 start bit + 8 bit of data +2 stop bits ➔ Efficiency = 8/11 = 72% (or overhead = 3/11 = 28%)
- Good for data with large gaps (e.g. keyboard, etc)



S1: receiver in idle state
S2: receive is receiving character

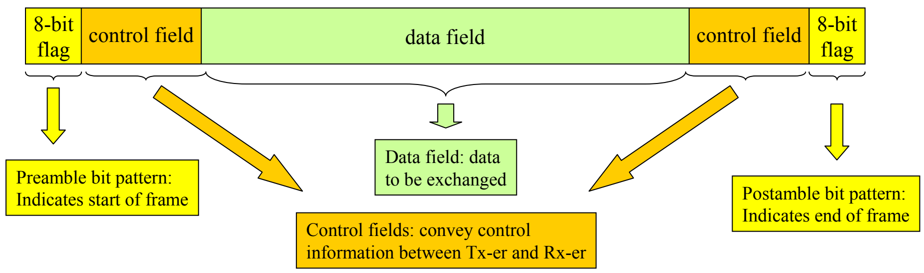


Synchronous Transmission

- What if there is a STEADY STREAM of bits between Tx-er and Rx-er
  - Still use the start/stop bits ➔ low efficiency
  - Use synchronous transmission
- Synchronous Techniques:
  - Provide SEPARATE clock signal
    - Expensive and only good for short distances
  - Depend on data encoding to extract clock info
    - E.g. Manchester encoding

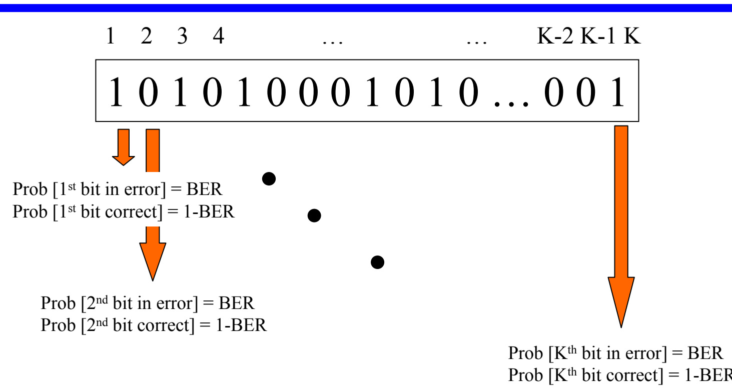
Synchronous Frame Format

- Typical Frame Structure



- For large data blocks, synchronous transmission is far more efficient than asynchronous:
  - E.g. HDLC frame 48 bits are used for control, preamble, and postamble – if 1000 bits are used for data ➔ efficiency = 99.4% (or overhead = 0.6%)

Error Detection



Prob [n bits in error in frame] = $\binom{K}{n}(BER)^n(1-BER)^{K-n}$

Error Detection – cont'd

- Hence, for a frame of K bits,

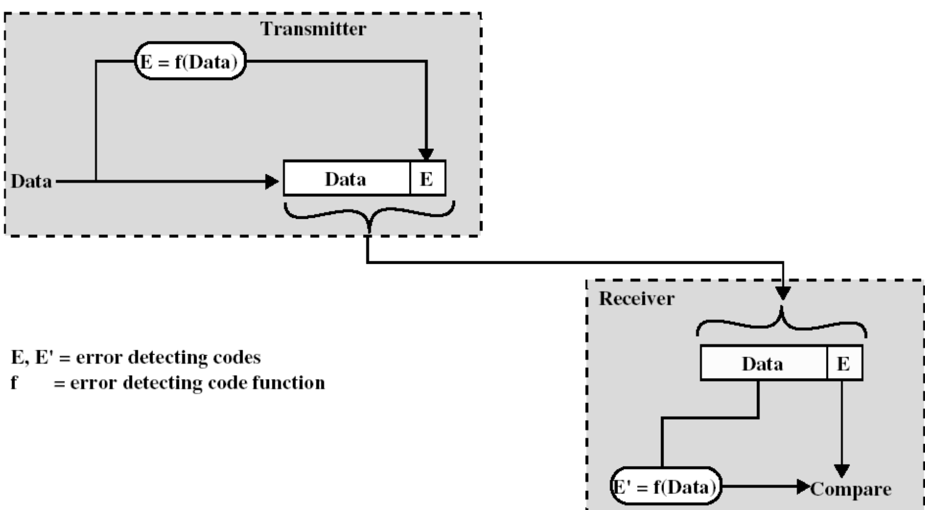
  Prob [frame is correct] = Prob [0 bits in error]
  = $(1\text{-BER})^K$

Prob [frame is erroneous] = Prob[1 OR MORE bits in error]
= 1 – Prob[0 bits in error]
= 1 - $(1\text{-BER})^K$

Or

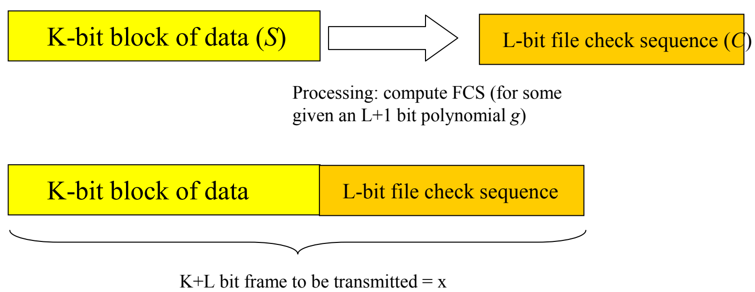
Prob [frame is erroneous] = Prob [1 bit in error] +
Prob[2 bits in error] + … +
Prob[K bits in error]
= 1 – Prob[0 bits in error]
= 1 - $(1\text{-BER})^K$

Error Detection (2)



E, E' = error detecting codes
f = error detecting code function

Cyclic Redundancy Check (CRC)



- Modulo 2 arithmetic (like XOR) is used to generate the FCS:
  - 0 ± 0 = 0; 1 ± 0 = 1; 0 ± 1 = 1; 1 ± 1 = 0
  - 1 X 0 = 0; 0 X 1 = 0; 1 X 1 = 1

CRC – Mapping Binary Bits into Polynomials

- Consider the following K-bit word or frame and its polynomial equivalent:

$$s_{K-1}\ s_{K-2}\ \ldots\ s_2\ s_1\ s_0 \rightarrow s_{K-1}D^{K-1} + s_{K-2}D^{K-2} + \ldots + s_1D^1 + s_0$$

where s_i ($K-1 \leq i \leq 0$) is either 1 or 0

- <u>Example1</u>: an 8 bit word s = 11011001 is represented as $s(D) = D^7+D^6+D^4+D^3+1$

CRC – Mapping Binary Bits into Polynomials - cont'd

- Example2: What is $D^4M(D)$ equal to?

  $D^4M(D) = D^4(D^7+D^6+D^4+D^3+1) = D^{11}+D^{10}+D^8+D^7+D^4$, the equivalent bit pattern is 110110010000 (i.e. four zeros added to the left of the original M pattern)

- Example3: What is $D^4M(D) + (D^3+D+1)$?

  $D^4M(D) + (D^3+D+1) = D^{11}+D^{10}+D^8+D^7+D^4+ D^3+D+1$, the equivalent bit pattern is 110110011011 (i.e. pattern 1011 = D^3+D+1 added to the left of the original M pattern)

CRC Calculation

- x = (K+L)-bit frame to be tx-ed, L < K
- s = K-bit message, the first K bits of frame T
- c = L-bit FCS, the last L bits of frame T
- g = pattern of L+1 bits (a predetermined divisor)

T = (K+L)-bit frame → | s = K-bit message | c = L-bit FCS |

g =(L+1) bit divisor

Note:
-x(D) is the polynomial (of K+L-1st degree or less) representation of frame x
-s(D) is the polynomial (of K-1st degree or less) representation of message s
- c(D) is the polynomial (of L-1st degree or less) representation of FCS
- g(D) is the polynomial (of Lth degree) representation of the divisor P
-x(D) = D^L s(D) + c(D) – refer to previous example

CRC Calculation (2)

- Design: frame x such that it divides the pattern g with no remainder?
- Solution: Since the first component of x, s, is the data part, it is required to find c (or the FCS) such that x divides g with no remainder

Using the polynomial equivalent:
$x(D) = D^L s(D) + c(D)$
One can show that $c(x)$ = remainder of $[D^L s(D)]/ g(D)$
i.e if $D^L s(D) / g(x)$ is equal to $z(D) + r(D)/g(D)$, then $c(D)$ is set to be equal to $r(X)$.

Note that:

$$\frac{\text{Polynomial of degree } K+L}{\text{Polynomial of degree } L} = \text{polynomial of degree } K + \text{remainder polynomial of degree } L-1$$

CRC Calculation - Procedure

1. **Shift pattern s by L bits to the lift**
2. **Divide the new pattern $D^L s(D)$ by the pattern g**
3. **The remainder of the division R (L bits) is set to be the FCS or c(D)**
4. **The desired frame x is $D^L s(D)$ plus the c(D)**

CRC Calculation Example

- Message s = 1010001101 (10 bits) ➔ k = 10
- $s(D) = D^9 + D^7 + D^3 + D^2 + 1$ ➔ $D^5s(D) = D^{14} + D^{12} + D^8 + D^7 + D^5$

Pattern P = 110101 (6 bits – note 0th and Lth bits are 1s) ➔ L + 1 = 6 ➔ L = 5

- $g(D) = D^5 + D^4 + D^2 + 1$

Find the frame T to be transmitted?

- Solution:

$$
\begin{array}{l}
\qquad\qquad\qquad\qquad D^9 + D^8 + D^6 + D^4 + D^2 + D \\
D^5 + D^4 + D^2 + 1 \;\big)\; D^{14} + D^{12} + D^8 + D^7 + D^5 \\
D^{14} + D^{13} + D^{11} + D^9 \\
D^{13} + D^{12} + D^{11} + D^9 + D^8 + D^7 + D^5 \\
D^{13} + D^{12} + D^{10} + D^8 \\
D^{11} + D^{10} + X^9 + D^7 + D^5 \\
D^{11} + D^{10} + X^8 + D^6 \\
D^9 + D^8 + D^7 + D^6 + D^5 \\
X^9 + D^8 + D^6 + D^4 \\
+D^7 + D^5 + D^4 \\
+D^7 + D^6 + D^4 + D^2 \\
D^6 + D^5 + D^2 \\
D^6 + D^5 + D^3 + D \\
+D^3 + D^2 + D
\end{array}
$$

- **FCS = $R(D) = D^3 + D^2 + D$**
  (or $0D^4+D^3+D^2+D$)

➔ c is equal to 01110

- **Frame x = 101000110101110**
- **As an exercise, verify that x(D) divided by g(D) has no remainder**

CRC Calculation – The previous example BUT using Polynomials – cont'd

- **Message s = 1010001101 (10 bits)**
- ➔ **$s(D) = D^9 + D^7 + D^3 + D^2 + 1$**
- ➔ **➔ $D^5s(D) = D^{14} + D^{12} + D^8 + D^7 + D^5$**
- **Pattern g = 110101**

➔ $g(D) = D^5 + D^4 + D^2 + 1$

- **$c(D) = D^3 + D^2 + D$**
- **$z(D) = D^9+D^8+D^6+D^4+D^2+D$**
- **$x(X) = D^5s(D) + c(D)$**
  $= D^{14} + D^{12} + D^8 + D^7 + D^5 + D^3 + D^2 + D$,
  or
  T = 101000110101110

- **<u>Exercise</u>: Verify that $z(D)\, g(D) + c(D) = D^5\, s(D)$**

CRC – Receiver Procedure

- **Tx-er transmits frame x**
- **Channel introduces error pattern E**
- **Rx-er receives frame y = x⊕E (note that if E = 000..000, then y is equal to x, i.e. error free transmission)**
- **y is divided by g, Remainder of division is R**
- **if R is ZERO, Rx-er assumes no errors in frame; else Rx-er assumes erroneous frame**
- **If an error occurs and y is still divisible by P → UNDETECTABLE error (this means the E is also divisible by g)**

Some Properties

- All single-bit errors are detected
  - Proof in textbook page 63 (problem 2.3)
- All double-bit errors are detected, if g(D) is chosen to be primitive polynomial and the string s is of length less or equal to 2^L-1
  - Proof in the textbook page 63/64
- Any odd number of errors, as long as P(x) contains a factor (D+1)
  - See problem 2.14

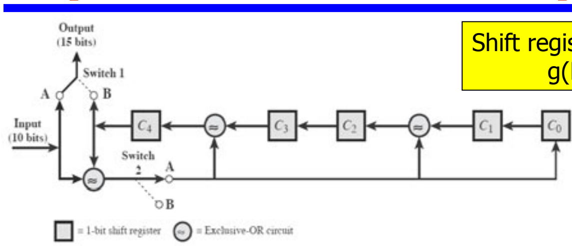
Some Popular CRC Polynomials

- CRC-12: D12+D11+D3+D2+D+1
- CRC-16: D16+D15+D2+1
- CRC-CCITT: D16+D12+D5+1
- CRC-32: D32+D26+D23+D22+D16+D12+D11+D10+D8+D7+D5+D4+D2+D+1

- CRC-12 – used for transmission of streams of 6-bit characters and generates a 12-bit FCS
- CEC-16 and CRC-CCITT – used for transmission of 8-bit characters in USA and Europe – result in 16-bit FCS
- CRC-32 – used in IEEE802 LAN standards

CRC – Shift Register Implementation – Example

Shift register circuit for dividing by $g(D) = D^5+D^4+D^2+1$



	C_4	C_3	C_2	C_1	C_0	$C_4 \oplus C_3 \oplus I$	$C_4 \oplus C_1 \oplus I$	$C_4 \oplus I$	I = input
Initial	0	0	0	0	0	1	1	1	1
Step 1	1	0	1	0	1	1	1	1	0
Step 2	1	1	1	1	1	1	1	0	1
Step 3	1	1	1	1	0	0	0	1	0
Step 4	0	1	0	0	1	1	0	0	0
Step 5	1	0	0	1	0	1	0	1	0
Step 6	1	0	0	0	1	0	0	0	1
Step 7	0	0	0	1	0	1	0	1	1
Step 8	1	0	0	0	1	1	1	1	0
Step 9	1	0	1	1	1	0	1	0	1
Step 10	0	1	1	1	0				

MSB

Message to be sent

Refer to previous example:
s = 1010001101 (K=10)
g = 110101 (L=5)
c = 01110

What are the effects of the switch positions A and B?