King Fahd University of Petroleum & Minerals
Computer Engineering Dept

COE 540 –Computer Networks
Term 102

Dr. Ashraf S. Hasan Mahmoud
Rm 22-148-3
Ext. 1724
Email: ashraf@kfupm.edu.sa

Lecture Contents

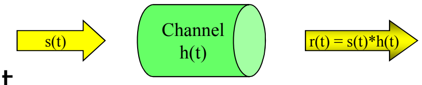
1. Channels and Models
2. Error Detection
3. ARQ: Retransmission Strategies
4. Framing
5. Standard DLCs

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

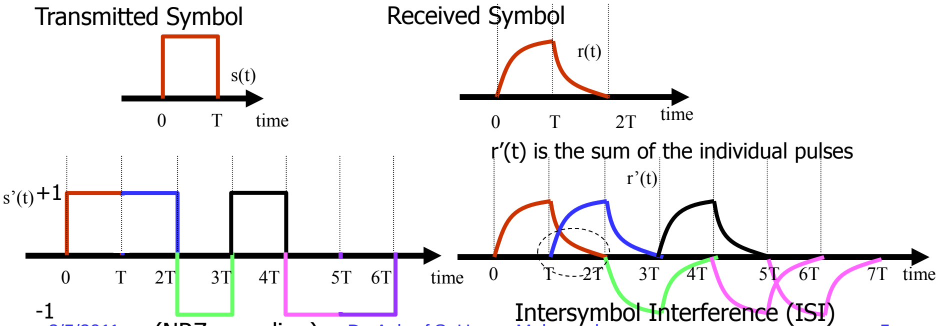
Channels and Models

- 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 <u>Linear-Time Invariant</u> 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

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

(NRZ encoding)

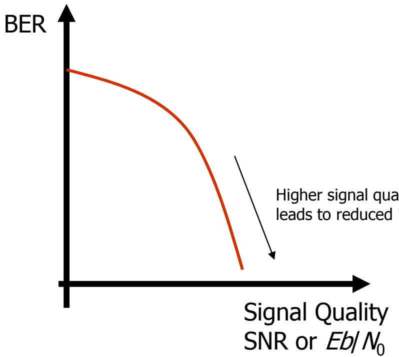
Intersymbol Interference (ISI)

Intersymbol Interference

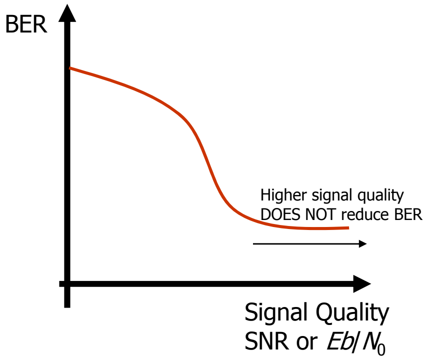
- One symbol is being received while the tail(s) of the preceding symbols are not finished
  - A limit on channel bit rate
  - Irreducible error floor
- A similar phenomena appears if there are multiple delayed copies of the same single transmitted symbol
  - Multipath
  - A real-problem for high speed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

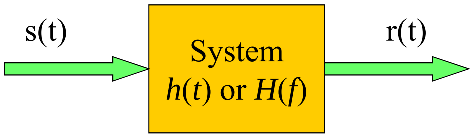
- **For linear Systems:**
  - **h(t) is the system's impulse response – i.e. r(t)= h(t) when s(t) = δ(t)**
  - **s(t) is system input signal**
  - **r(t) is system output signal**

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

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

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

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

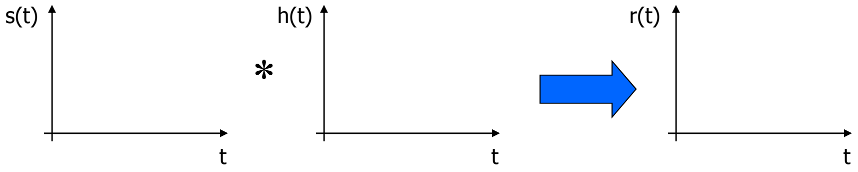
where α = 2/T

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

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

Hint: Π(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:**

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

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

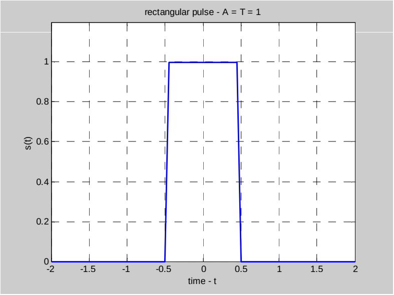
$$A_n = \frac{2}{T} \int_0^T s(t) \cos(2\pi n f_0 t)dt$$

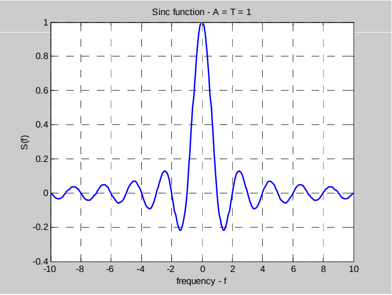
f_0 is the fundamental frequency and is equal to 1/T

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**

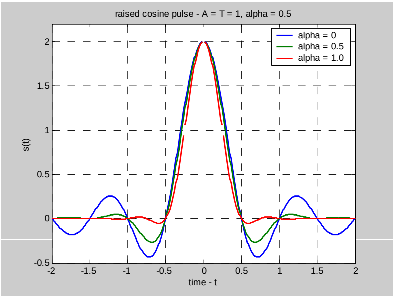
$$s(t) = A\frac{\sin\left(\pi Wt\right)}{\left(\pi Wt\right)}$$

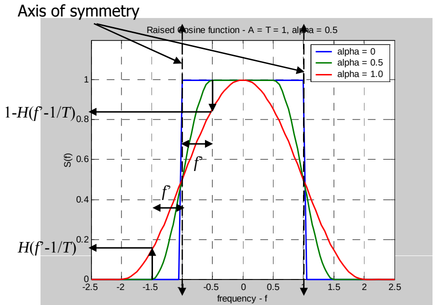
$$S(f) = \frac{A}{W}\Pi\left(f/W\right)$$

$S(f) = A/W$ 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
```

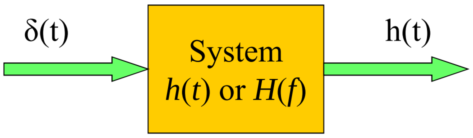
```
figure(1);
plot(t, s_t);           % plot s(t)
title('raised cosine pulse - A = T = 1');
xlabel('time - t');
ylabel('s(t)');
legend('alpha = 0', 'alpha = 0.5', 'alpha = 1.0');
axis([-2 2 -0.5 2.2]);
grid

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) = αe$^{-αt}$ 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

- **Theorem: 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

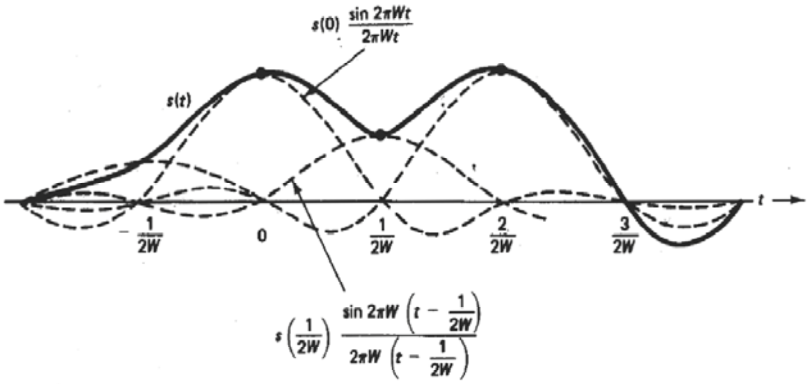


Figure 2.6 Sampling theorem, showing a function $s(t)$ that is low-pass limited to frequencies at most W. The function is represented as a superposition of $(\sin x)/x$ functions. For each sample, there is one such function, centered at the sample and with a scale factor equal to the sample value.

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*"**

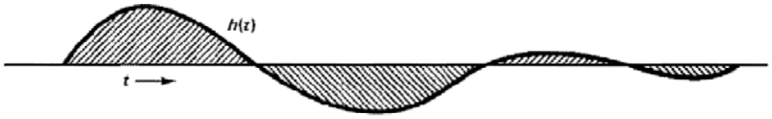


Figure 2.8 Impulse response $h(t)$ for which $H(f) = 0$ for $f = 0$. Note that the area over which $h(t)$ is positive is equal to that over which it is negative.

Bandpass Channels – cont'd

- **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**

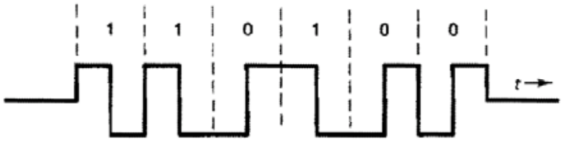
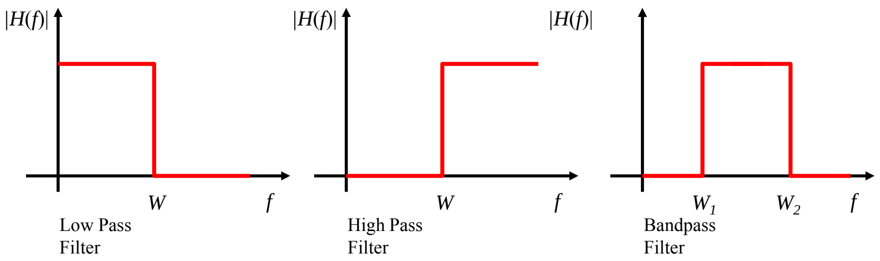


Figure 2.9 Manchester coding. A binary 1 is mapped into a positive pulse followed by a negative pulse, and a binary 0 is mapped into a negative pulse followed by a positive pulse. Note the transition in the middle of each signal interval.

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_0**
- **f_0 is the carrier frequency!**

Bandpass Signal

Baseband Signal

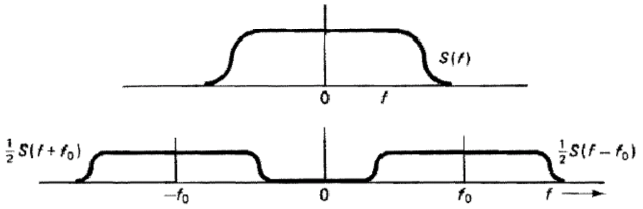


Figure 2.10 Amplitude modulation. The frequency characteristic of the waveform $s(t)$ is shifted up and down by f_0 in frequency.

Modulation

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

Analog Communications

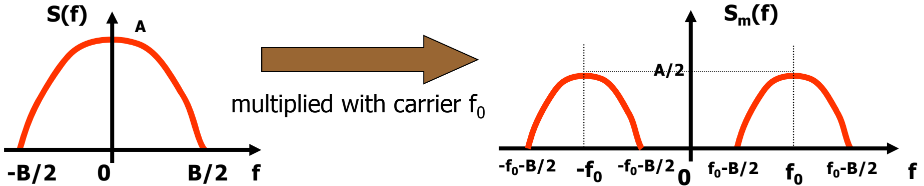
Amplitude Modulation (AM)

- Consider the signal s(t),

$$s_m(t) = s(t) \text{ X } \cos(2\pi f_0 t)$$

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

$$S_m(f) = ½ \text{ X } \{S(f-f_0) + S(f+f_0)\}$$

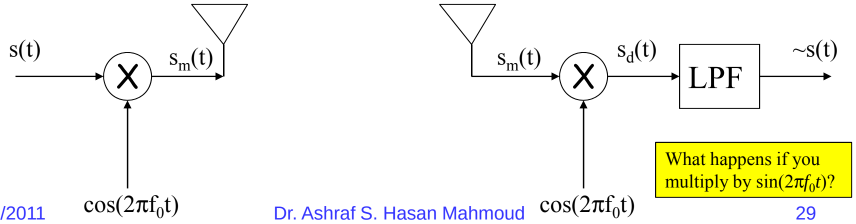


Analog Communications

Modulation – Txer/Rxer

- **At the receiver side:**

$$\begin{aligned} s_d(t) &= s_m(t)\ \text{X}\ \cos(2\pi f_0 t) \\ &= s(t)\ \text{X}\ \cos(2\pi f_0 t)\ \text{X}\ \cos(2\pi f_0 t) \\ &= \underbrace{\tfrac{1}{2}\, s(t)}_{\text{desired term}} + \underbrace{\tfrac{1}{2}\, s(t)\ \text{X}\ \cos(2\pi 2\text{X} f_0 t)}_{\text{undesired term – signal centered around } 2f_c \text{ filtered out using the LPF}} \end{aligned}$$



What happens if you multiply by $\sin(2\pi f_0 t)$?

Quadrature Amplitude Modulation (QAM)

- **Transmits twice as many bits as AM**
- **At receiver – respective multiplication with the carriers is performed – LPF is used to obtain the original signals (scaled by ½)**
- **Adaptive equalizer – compensate for variation in the channel**

- **What is the role of the pulse shape filter (in the modulator)?**

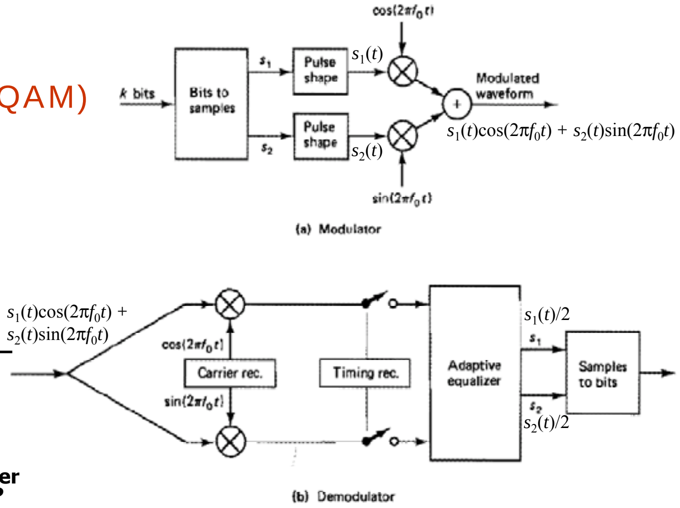
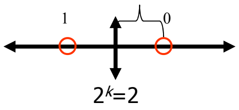


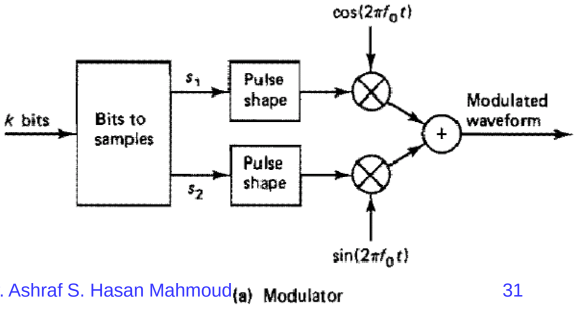
Figure 2.11 Quadrature amplitude modulation. (a) Each sample period, *k* bits enter the modulator, are converted into quadrature amplitudes, are then modulated by sine and cosine functions, respectively, and are then added. (b) The reverse operations take place at the demodulator.

Quadrature Amplitude Modulation (QAM) – cont’d

- **How to map the k bits into samples s1 and s2?**

- **k = 1, i.e. one branch of modulator is working**
- **bit 1 is mapped to +1 and bit 0 is mapped to -1 ➔ Binary ASK**

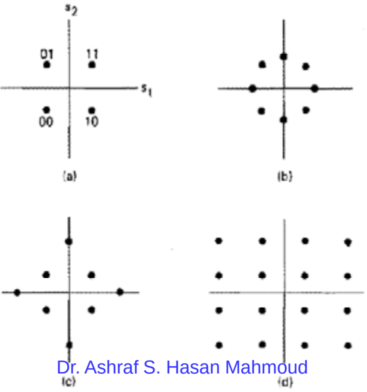




Signal Constellation

- **k = 2, bits b1b2 ➔ s1 is either +1 or -1 depending on b1, also s2 is either +1 or -1 depending on b2 ➔ QAM**

- **k bits ➔ 2^k combinations – mapped into different amplitude pairs – signal constellation**



One amplitude:
a) PSK – $2^k = 4$
b) PSK – $2^k = 8$
Two amplitudes:
c) ASK/PSK – $2^k = 8$
d) QAM – $2^k = 16$

Figure 2.12 Signal constellations for QAM. Part (a) maps two binary digits into a quadrature amplitude sample. Parts (b) and (c) each map three binary digits, and part (d) maps four binary digits. Parts (a) and (b) also can be regarded as phase-shift keying.

Why (c) is a better modulation scheme compared to (b)?

Bandwidth and Capacity

- For voice circuits (telephone lines)
  - W = 2400 Hz
- Capacity:
  - k = 2 → R = 2x2400 = 4800 b/s
  - k = 4 → R = 4x2400 = 9600 b/s
- Very interesting – why not use k as large as possible (i.e. higher order of (modulation) levels)?
- The answer: BER

Digital Communications

Shannon Capacity

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

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

where SNR = S / (N0 W) is the ratio of signal power to noise power – a measure of the signal quality

Example 3: Shannon Capacity

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

SNR = 15 dB = $10^{\wedge(15/10)}$ = 31.6

C = $200 X 10^3$ X $\log_2(1+31.6)$

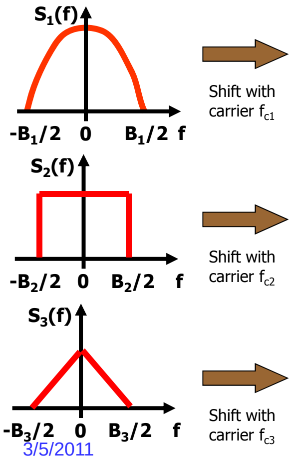
= 1005.6 kb/s

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

Shannon Limit

- **Shannon asserts that with the use of error-correction coding, ANY rate less than C CAN BE ACHIEVED with ARBITRARY small error probability**
- **Very powerful statement**
- **Shannon does not specify how to achieve this capacity reliably – communications research!!**

Frequency Division Multiplexing (FDM)

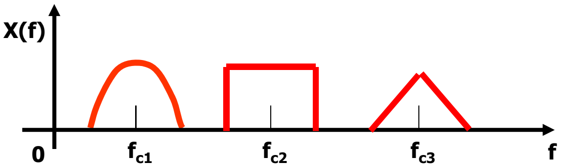


$x(t) = s_1(t) \text{ X } \cos(2\pi f_{c1}t) + s_2(t) \text{ X } \cos(2\pi f_{c2}t) + s_3(t) \text{ X } \cos(2\pi f_{c3}t)$

- x(t) is transmitted on the media

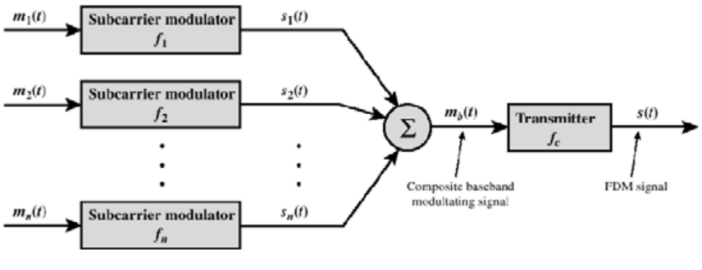
-The three spectra are not overlapping if f_{c1}, f_{c2}, and f_{c3} are chosen appropriately

-Original composite signals $s_1(t)$, $s_2(t)$, and $s_3(t)$ can be recovered using bandpass filters with appropriate bandwidths centered at f_{c1}, f_{c2}, and f_{c3}, respectively.

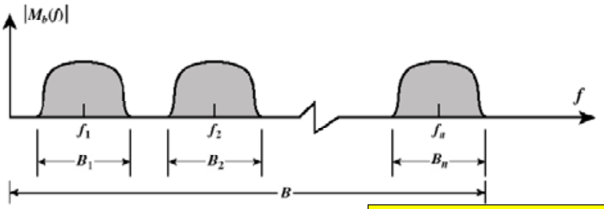


Frequency-Division Multiplexing - Transmitter

- $m_i(t)$: 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

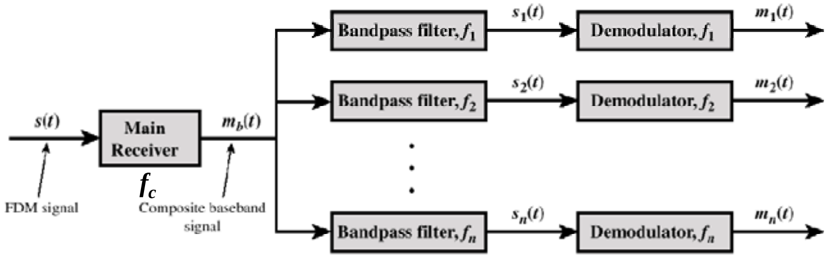


- Think of the bandwidth *B* divided amongst the *n* users – each gets *B*/*n* Hz
- All users can be active at the same time!
- Requires *n* modems!!

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	…				
…	…				

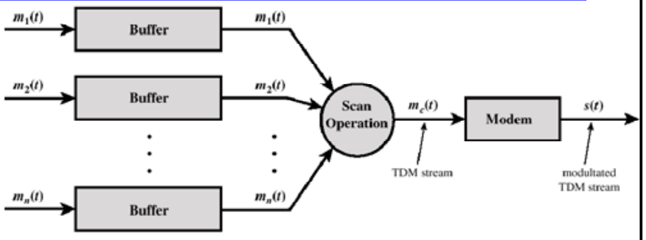
Other examples of FDM:
- AM/FM Radio Stations
- TV broadcasting

Other Examples of FDM

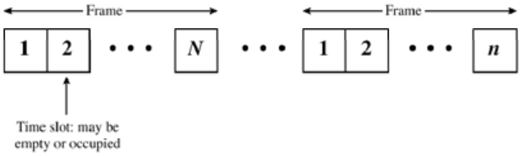
- AM/FM Radio Stations
- TV broadcasting

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)



(a) Transmitter

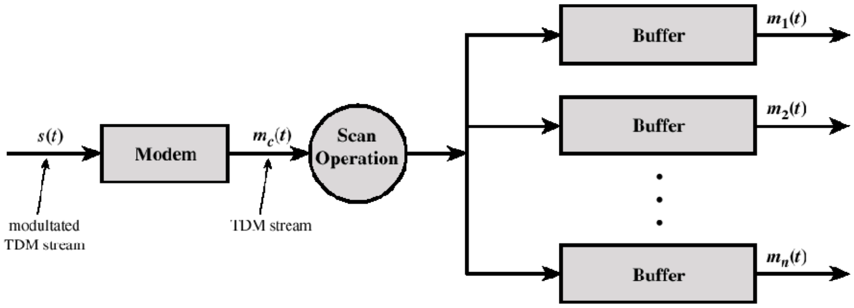


(b) TDM Frames

Requires ONE modem that uses the entire bandwidth B Hz!!

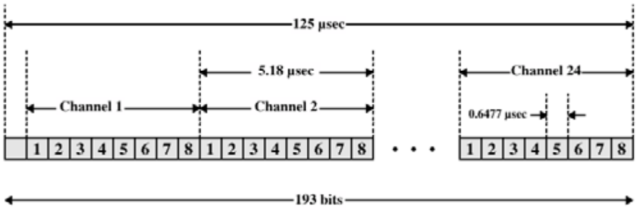
Synchronous Time-Division Multiplexing - Receiver

- 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



TDM – Example: 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.



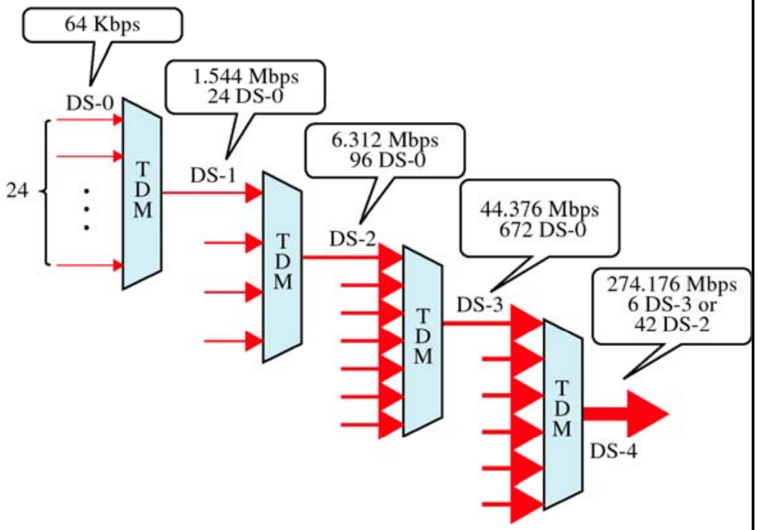
Notes:

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

TDM – Example 8: Digital Carrier Systems (2)

- TDM



Other Channel Impairments

- Main impairment – Thermal noise aka additive white Gaussian noise (AWGN)
- Other impairments
  - Phase jitter and frequency offsets
  - Nonlinear amplification (delay distortion)
  - Impulse noise (e.g. lightening)
  - Crosstalk/interference
- Errors causes by AWGN tend to be *random* and disbursed
- Errors caused by the other types of noises tend to occur in BURSTS of arbitrary length
  - Error detection and retransmission is performed by the data link layer)

Digital Channels

- T1/SONET hierarchy
- ISND and Broadband ISDN

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

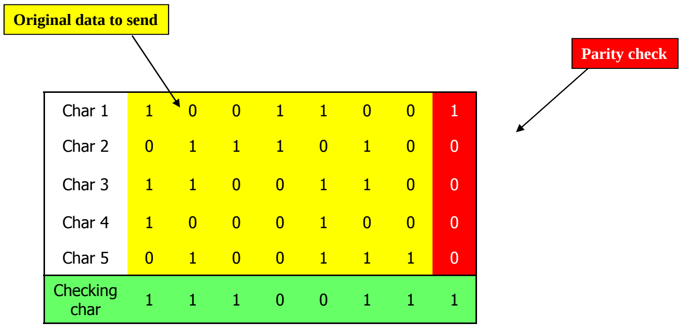
- Why type of errors does this scheme detect?
  - All odd number of errors – Does that depend on the length of the "data" string?
  - All even number of errors are not detected

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)



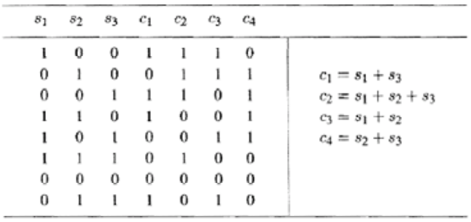
Char 1	1	0	0	1	1	0	0	1
Char 2	0	1	1	1	0	1	0	0
Char 3	1	1	0	0	1	1	0	0
Char 4	1	0	0	0	1	0	0	0
Char 5	0	1	0	0	1	1	1	0
Checking char	1	1	1	0	0	1	1	1

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) = 27%
- There could be undetected errors – How?

Linear Codes

- Code: the mathematical transformation to generate the code word (data + parity check)
- K data bits + L parity bits = Frame
- 2^K possible data strings ➔ 2K possible code words (each of length K + L bit)



s_1	s_2	s_3	c_1	c_2	c_3	c_4
1	0	0	1	1	1	0
0	1	0	0	1	1	1
0	0	1	1	1	0	1
1	1	0	1	0	0	1
1	0	1	0	0	1	1
1	1	1	0	1	0	0
0	0	0	0	0	0	0
0	1	1	1	0	1	0

$c_1 = s_1 + s_3$

$c_2 = s_1 + s_2 + s_3$

$c_3 = s_1 + s_2$

$c_4 = s_2 + s_3$

Figure 2.15 Example of a parity check code. Code words are listed on the left, and the rule for generating the parity checks is given on the right.

Linear Codes (2)

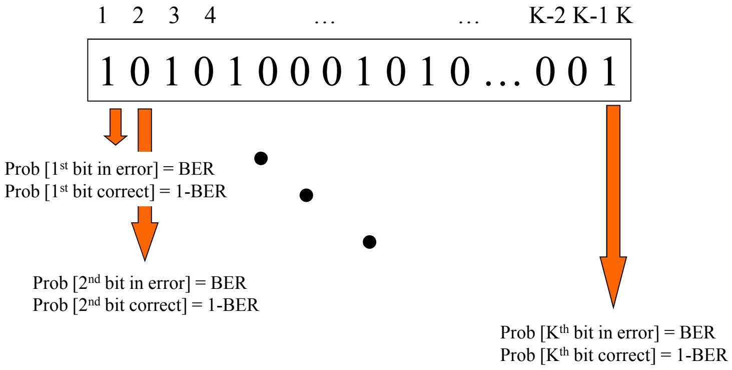
- Effectiveness of the code:
  - Minimum distance of the code – def = smallest number of errors that can convert one code word to another
    - E.g. for single bit parity checks – min distance = 2, for horizontal and vertical parity checks - min distance = 4
  - The burst detecting capability – def = smallest integer B such that a code can detect all burst of length B or less
    - E.g. for single bit parity checks – burst length = 1, for horizontal and vertical parity checks – burst length = 1 + length of row (assuming rows are transmitted one after the other)
  - Probability of an undetected error ~ 2^{-L} (How? See textbook page 61)
    - Typically, longer parity checks ➔ lower undetected error probability

---

Linear Codes – Error Correction

- 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).

Error Detection



1 2 3 4 … … K-2 K-1 K

1 0 1 0 1 0 0 0 1 0 1 0 … 0 0 1

Prob [1^{st} bit in error] = BER
Prob [1^{st} bit correct] = 1-BER

Prob [2^{nd} bit in error] = BER
Prob [2^{nd} bit correct] = 1-BER

Prob [K^{th} bit in error] = BER
Prob [K^{th} bit correct] = 1-BER

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

Error Detection – cont'd

- Hence, for a frame of K bits,

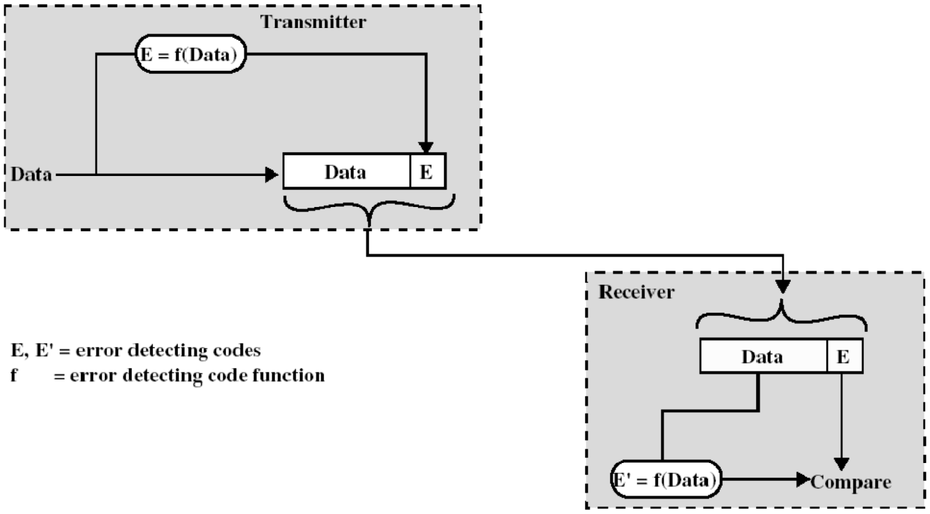
Prob [frame is correct] = Prob [0 bits in error]
= $(1\text{-BER})^K$

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)

K-bit block of data (*S*) → L-bit file check sequence (*C*)

Processing: compute FCS (for some given an L+1 bit polynomial *g*)

K-bit block of data | L-bit file check sequence

K+L bit frame to be transmitted = x

- 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

- Example1: 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?

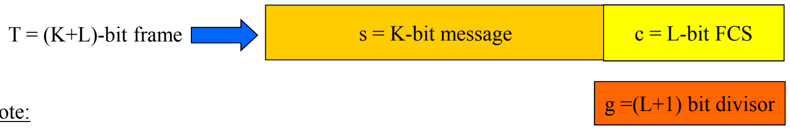
$D^4M(D) = D^4(D^7+D^6+D^4+D^3+1) = D^{11}+D^{10}+D^8+D^7+D^4$, the equivalent bit pattern is 110110010000 (i.e. four zeros added to the left of the original M pattern)

- 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)
  $g(D) = D^L + g_{L-1}D^{L-1} + \ldots + g_1D + 1$



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:

Polynomial of degree K+L
--------------------------------- = polynomial of degree K + remainder polynomial of degree L-1
Polynomial of degree L

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**
- **$s(D) = D^9 + D^7 + D^3 + D^2 + 1$ ➔ $D^5 s(D) = D^{14} + D^{12} + D^8 + D^7 + D^5$**

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:**

$$\begin{array}{r|cccccccccccccc}
 & D^9 & +D^8 & & +D^6 & & +D^4 & & +D^2 & +D & & & & & \\
\cline{2-15}
D^5 \; +D^4 \; +D^2 \; +1 & D^{14} & & +D^{12} & & & & +D^8 & +D^7 & & +D^5 & & & & \\
\cline{1-1}
 & D^{14} & +D^{13} & & +D^{11} & & +D^9 & & & & & & & & \\
\cline{2-7}
 & & D^{13} & +D^{12} & +D^{11} & & +D^9 & +D^8 & +D^7 & & +D^5 & & & & \\
 & & D^{13} & +D^{12} & & +D^{10} & & +D^8 & & & & & & & \\
\cline{3-8}
 & & & & D^{11} & +D^{10} & +X^9 & & +D^7 & & +D^5 & & & & \\
 & & & & D^{11} & +D^{10} & & +X^8 & & +D^6 & & & & & \\
\cline{5-10}
 & & & & & & D^9 & +D^8 & +D^7 & +D^6 & +D^5 & & & & \\
 & & & & & & X^9 & +D^8 & & +D^6 & & +D^4 & & & \\
\cline{7-12}
 & & & & & & & & +D^7 & & +D^5 & +D^4 & & & \\
 & & & & & & & & +D^7 & +D^6 & & +D^4 & & +D^2 & \\
\cline{9-14}
 & & & & & & & & & D^6 & +D^5 & & & +D^2 & \\
 & & & & & & & & & D^6 & +D^5 & & +D^3 & & +D \\
\cline{10-15}
 & & & & & & & & & & & & +D^3 & +D^2 & +D
\end{array}$$

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

➔ c is equal to 01110

- **Frame x = 1010001101*01110***
- **As an exercise, verify that x(D) divided by g(D) has no remainder**

CRC Calculation – cont’d

- Message s = 1010001101 (10 bits)

➔ s(D) = $D^9 + D^7 + D^3 + D^2 + 1$

➔ ➔ $D^5 s(D) = D^{14} + D^{12} + D^8 + D^7 + D^5$

- Pattern g = 110101

➔ $g(D) = D^5 + D^4 + D^2 + 1$

- $c(D) = D^3 + D^2 + D$
- $z(D) = D^9+D^8+D^6+D^4+D^2+D$
- $x(X) = D^5 s(D) + c(D)$

  $= D^{14} + D^{12} + D^8 + D^7 + D^5 + D^3 + D^2 + D,$

  or

  T = 1010001101**01110**

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

For $g(D)$, g_0 must be 1 and g_L must be 1

CRC Calculation – Shift Register Circuit

- The long division can be implemented in hardware by the feedback shift register circuit.
- Operations:
  - Put switch on position (1)
  - Initially first L bits of s(D) are loaded (sK-1 the MSB is at the right)
  - K shifts – all data is pushed in
  - Move switch to position (2)
  - Read the CRC – requires L shifts

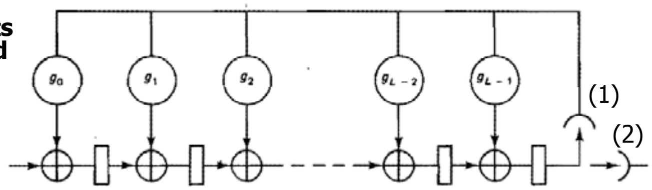


Figure 2.16 Shift register circuit for dividing polynomials and finding the remainder. Each rectangle indicates a storage cell for a single bit and the preceding circles denote modulo 2 adders. The large circles at the top indicate multiplication by the value of g_i. Initially, the register is loaded with the first L bits of $s(D)$ with s_{K-1} at the right. On each clock pulse, a new bit of $s(D)$ comes in at the left and the register reads in the corresponding modulo 2 sum of feedback plus the contents of the previous stage. After K shifts, the switch at the right moves to the horizontal position and the CRC is read out.

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

- x is a code word iff divisible by g(D)
  - E.g. $x(D) = g(D)z(D)$ – where $z(D) = s(D)D^L/g(D)$
- Assume the received frame is y(D), then $y(D) = x(D) + e(D)$
  - e(D) – is the error sequence or each error in the frame corresponds to a nonzero coefficient in e(D)
- Remainder [y(D)/g(D)] = Remainder [e(D)/g(D)]
  - Prove this?
- If $e(D) = 0$ ➔ frame is error free, Remainder $[e(D)/g(D)] = 0$
- If $e(D) \neq 0$ ➔ There is error(s)
  - If Remainder $[e(D)/g(D)] = 0$ ➔ UNDETECTABLE ERROR
  - If Remainder $[e(D)/g(D)] \neq 0$ ➔ DETECTABLE ERROR

Some Properties (2)

Key Result

- $e(D) \neq 0$ is UNDETECTABLE iff $e(D) = g(D)z(D)$ for some nonzero polynomial $z(D)$

- All single-bit errors are detected
  - Proof in textbook page 63 (problem 2.3)

Some Properties (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

Design of Generator Polynomial

- g(D) is chosen as the product of a primitive polynomial of degree L-1 times the polynomial D+1
  - All odd errors are detected
  - All double bit errors are detected (for block lengths less than 2^{L-1})
  - ➔ minimum distance = 4
  - ➔ burst length = (at least L)
  - ➔ probability of failing to detect errors in completely random strings = 2^{-L}

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