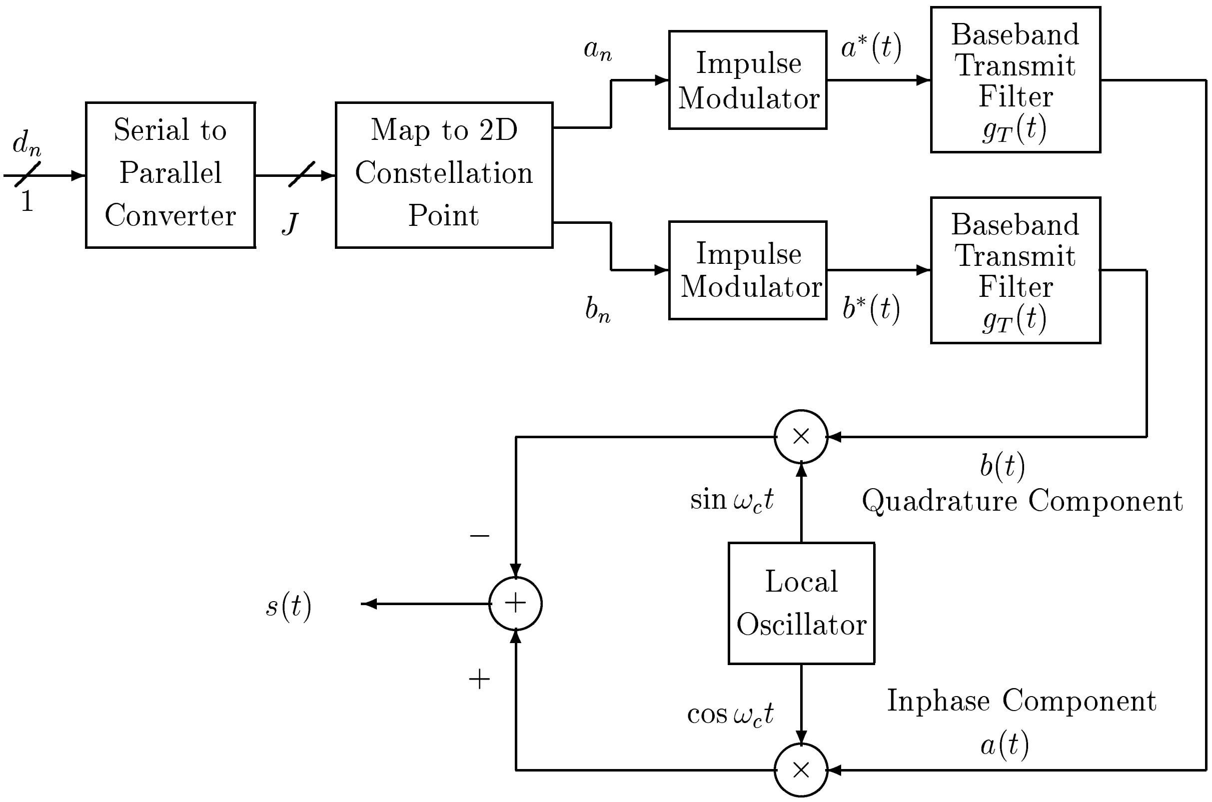
Chapter 13

Fundamentals of Quadrature Amplitude Modulation (QAM)

Contents

Chapter 13
Fundamentals of Quadrature Amplitude Modulation (QAM)

- QAM is a generalization of PAM to bandpass channels.
- It uses bandwidth efficiently and linear channel distortions can be corrected by adaptive equalization at the receiver.
- QAM fits in nicely with a combined coding and modulation scheme called *trellis coded modulation* (TCM) and a method for selecting transmitted symbols known as *shell mapping.*
- ITU-T V series telephone line modems from V.22 through V.34 use QAM. V.90 uses QAM from the client modem to the server modem (upstream) and PAM from the server to the client (downstream). Modern FAX modems also use QAM.
- QAM is also used in high speed cable, microwave, and satellite systems.



A Basic QAM Transmitter

Description of QAM Transmitter Diagram

- The input is a serial binary data stream d_n at R_d bps.
- The Serial to Parallel Converter groups the input bits into J-bit binary words.
- Each J-bit word selects a channel symbol from a 2^J element alphabet resulting in a channel symbol rate of $f_s = R_d/J$ baud.
- The alphabet consists of pairs of real numbers representing points in a 2-dimensional space and is called the *signal constellation.*
- It will be convenient to represent the channel symbol sequence by the sequence of complex numbers $c_n = a_n + jb_n$. It is customary to call the real part, a_n, the *inphase* or I component and the imaginary part, b_n, the *quadrature* or Q component.

Transmitter Description (cont. 1)

Outputs of the Impulse Modulators

$$a^*(t) = \sum_{k=-\infty}^{\infty} a_k \delta(t - kT)$$

$$b^*(t) = \sum_{k=-\infty}^{\infty} b_k \delta(t - kT)$$

Outputs of the Baseband Shaping Filters

Inphase Component

$$a(t) = \sum_{k=-\infty}^{\infty} a_k g_T(t - kT)$$

Quadrature Component

$$b(t) = \sum_{k=-\infty}^{\infty} b_k g_T(t - kT)$$

Transmitter Description (cont. 2)

Forming the QAM Signal

The baseband components are modulated by quadrature carriers to form the transmitted signal:

$$s(t) = a(t)\cos\omega_c t - b(t)\sin\omega_c t$$

- The carrier frequency ω_c must be greater than the shaping filter cutoff frequency to prevent spectral fold-over.

Example

  - A typical voiceband telephone channel has a passband of about 300 Hz to 3100 Hz.
  - The symbol rate for a V.32bis modem is $f_s = 2400$ Hz.
  - The carrier frequency for a V.32bis modem is $f_c = 1800$ Hz.
  - Notice that the symbol rate is larger than the carrier frequency!

Complex Signal Representation

The Transmitted QAM Signal

$$s(t) = \Re e\{[a(t) + jb(t)]e^{j\omega_c t}\}$$

The Pre-Envelope or Analytic Signal

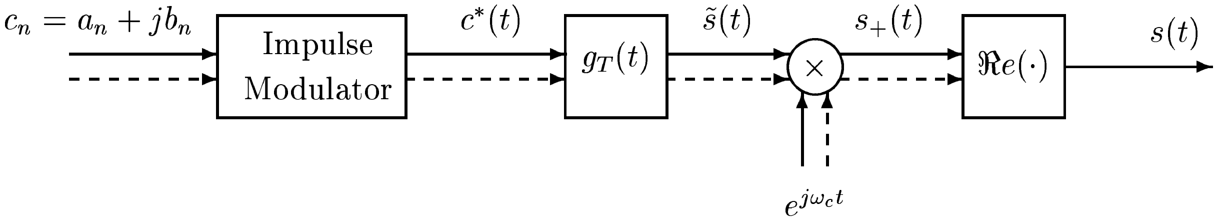
$$\begin{aligned} s_+(t) &= s(t) + j\hat{s}(t) = [a(t) + jb(t)]e^{j\omega_c t} \\ &= \sum_{k=-\infty}^{\infty} (a_k + jb_k)g_T(t - kT) \;\; e^{j\omega_c t} \end{aligned}$$

The Complex Envelope

$$\tilde{s}(t) = s_+(t)e^{-j\omega_c t} = a(t) + jb(t)$$

The QAM modulator can be compactly represented in terms of these complex signals as shown in the figure on the next slide.

Representation of the QAM Modulator in Terms of Complex Signals



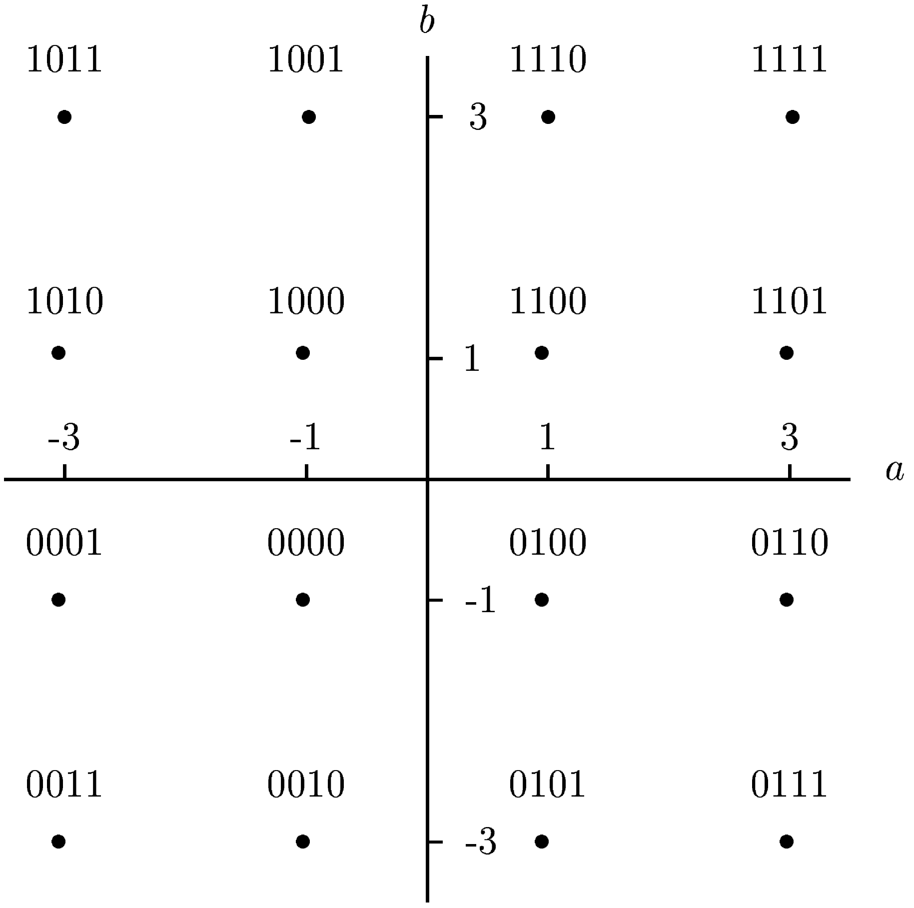
Two Constellation Examples

Examples of two constellations are described on the following slides:

(1) A 4×4 rectangular 16-point constellation.
Used in V.22bis for 2400 bps at 600 baud and in V.32 uncoded for 9600 bps at 2400 baud.

(2) A 4-point rectangular subset of the first.
Used in V.22 and V.22bis for 1200 bps at 600 baud and in V.32 for 4800 bps at 2400 baud.

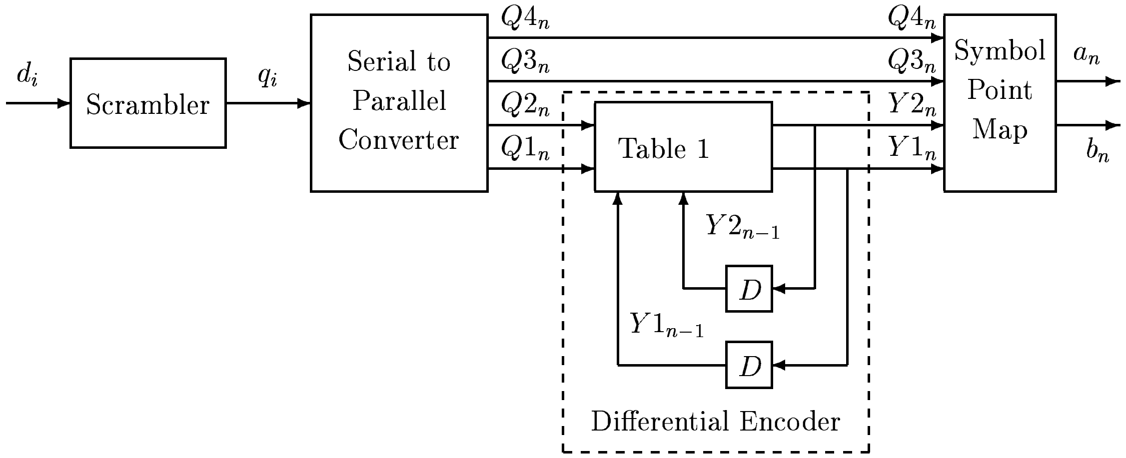
Data bits are assigned to points so that the system is transparent to 90° rotations.



Labels = $(Y1_n, Y2_n, Q3_n, Q4_n)$

The 16-Point Rectangular QAM Constellation

Mapping Input Bits to Constellation Points



Inputs		Previous Outputs		Quadrant Phase Change	Outputs	
$Q1_n$	$Q2_n$	$Y1_{n-1}$	$Y2_{n-1}$		$Y1_n$	$Y2_n$
0	0	0	0	+90°	0	1
0	0	0	1	+90°	1	1
0	0	1	0	+90°	0	0
0	0	1	1	+90°	1	0
0	1	0	0	0°	0	0
0	1	0	1	0°	0	1
0	1	1	0	0°	1	0
0	1	1	1	0°	1	1
1	0	0	0	+180°	1	1
1	0	0	1	+180°	1	0
1	0	1	0	+180°	0	1
1	0	1	1	+180°	0	0
1	1	0	0	+270°	1	0
1	1	0	1	+270°	0	0
1	1	1	0	+270°	1	1
1	1	1	1	+270°	0	1

Table 1. Differential Quadrant Coding for V.22bis and V.32 Uncoded Options

90° Rotational Invariance

The 4×4 constellation is invariant to 90° rotations. A 90° rotation gives the same diagram. This can be a problem because:

- a 90° carrier phase offset has the effect of multiplying the point $a_n + jb_n$ by j which rotates it by 90°. With this symmetry, carrier tracking loops in the receivers can only determine the correct phase to the nearest multiple of 90°.

The system can be made transparent to 90° phase offsets by a combination of

(1) differentially encoding two of the input bits to specify the quadrant ($Q1_n$ and $Q2_n$)

(2) assigning the remaining two input bits to points within a quadrant so that a 90° rotation leaves them unchanged. Notice that $Q3_n$ and $Q4_n$ are the same for the 90° rotations of each point.

A 4-Point Four Phase Constellation

A 4-point constellation can be formed from the 4×4 constellation by selecting the subset of points with labels $\{(1101), (1001), (0001), (0101)\}$.

- These points lie on a circle and are separated by 90° and are called a QPSK constellation (*Quadrature Phase Shift Keying*).
- It is used in V.22bis modem for transmission at 1200 bps with a symbol rate of 600 baud and in V.32 at 4800 bps with 2400 baud.
- It can be made transparent to 90° rotations by using the 4×4 mapper with inputs $(Y1_n, Y2_n, Q3_n, Q4_n) = (Y1_n, Y2_n, 0, 1)$
- These points were selected to make the 4-point constellation a subset of the 4×4 constellation and have about the same average power.

Scramblers

Scramblers are used to break up long strings of 1's or 0's in the input data sequence and cause the constellation points to be chosen pseudo-randomly. Systems in the receiver like an adaptive equalizer, and carrier and symbol clock tracking loops require this symbol variation to operate properly.

The V.22bis Scrambler

This uses the difference equation

$$q_i = d_i \oplus q_{i-14} \oplus q_{i-17}$$

where $\oplus$ represents modulo 2 addition or the exclusive-or logical function. The corresponding connection polynomial is

$$h(D) = 1 + D^{14} + D^{17}$$

Actually, the V.22bis scrambler is slightly more complicated in that it contains a means for

Scramblers (cont.)

detecting a string of 64 1's at its output and complementing the next output bit. This prevents the all 1's scrambler lock-up condition when the input is all 1's.

The V.32 Scramblers

Calling Modem

$$q_i = d_i \oplus q_{i-18} \oplus q_{i-23}$$

$$h_c(D) = 1 + D^{18} + D^{23}$$

Answer Modem

$$q_i = d_i \oplus q_{i-5} \oplus q_{i-23}$$

$$h_a(D) = 1 + D^{5} + D^{23}$$

A Modulator Structure Using Passband Shaping Filters

$$s_+(t) = \sum_{k=-\infty}^{\infty} \left(c_k e^{j\omega_c kT}\right) g_T(t-kT) e^{j\omega_c(t-kT)}$$

Let $h(t) = g_T(t) e^{j\omega_c t} = h_I(t) + jh_Q(t)$

where

$$h_I(t) = g_T(t)\cos\omega_c t \quad \text{and} \quad h_Q(t) = g_T(t)\sin\omega_c t$$

This filter is a bandpass filter with the frequency response $H(\omega) = G_T(\omega - \omega_c)$.

Let

$$c_k' = c_k e^{j\omega_c kT} = a_k' + jb_k'$$

where

$$a_k' = \Re e\{c_k'\} = a_k \cos\omega_c kT - b_k \sin\omega_c kT$$

and

$$b_k' = \Im m\{c_k'\} = a_k \sin\omega_c kT + b_k \cos\omega_c kT$$

Passband Shaping (cont. 1)

Using these definitions gives

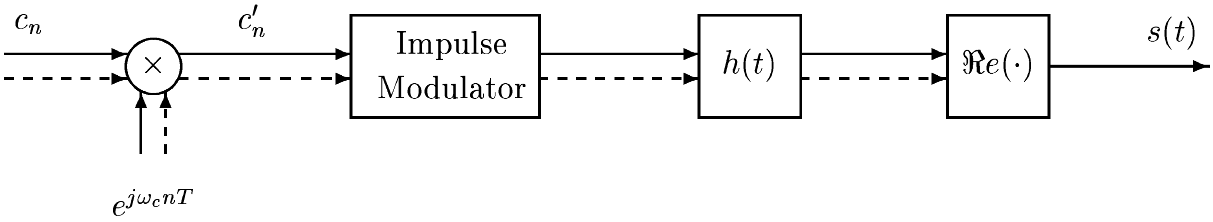
$$s_+(t) = \sum_{k=-\infty}^{\infty} c'_k h(t - kT)$$

and

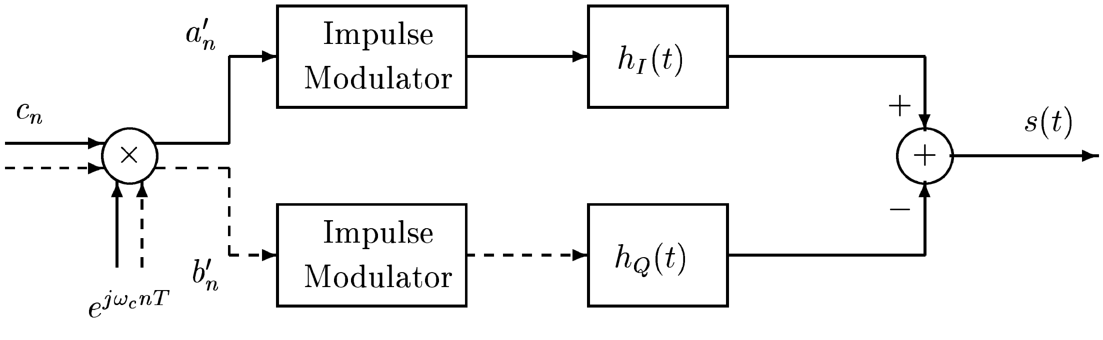
$$\begin{aligned} s(t) &= \Re e\{s_+(t)\} \\ &= \sum_{k=-\infty}^{\infty} a'_k h_I(t - kT) - b'_k h_Q(t - kT) \end{aligned}$$

Block diagrams for the modulator using passband shaping are shown on the next slide.

Block Diagrams of Modulators Using Passband Shaping Filters



QAM Modulator Using a Passband Shaping Filter



Expanded Modulator Block Diagram

Ideal QAM Demodulation

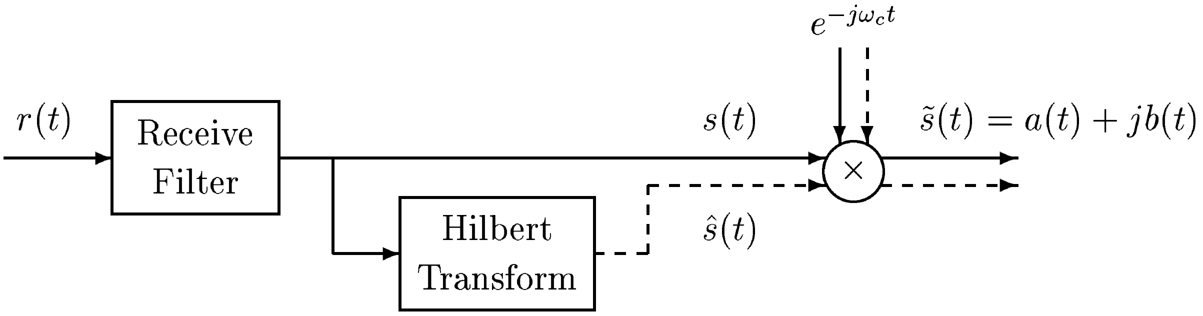
QAM Demodulator Using the Complex Envelope

The complex envelope is

$$\tilde{s}(t) = s_+(t)e^{-j\omega_c t} = \sum_{k=-\infty}^{\infty} (a_k + jb_k)g_T(t - kT)$$

So, if $g_T(t)$ has no ISI

$$\tilde{s}(nT) = a_n + jb_n$$



See Section 5.2.2 for a discussion of the Hilbert transform, pre-envelope, and complex envelope.

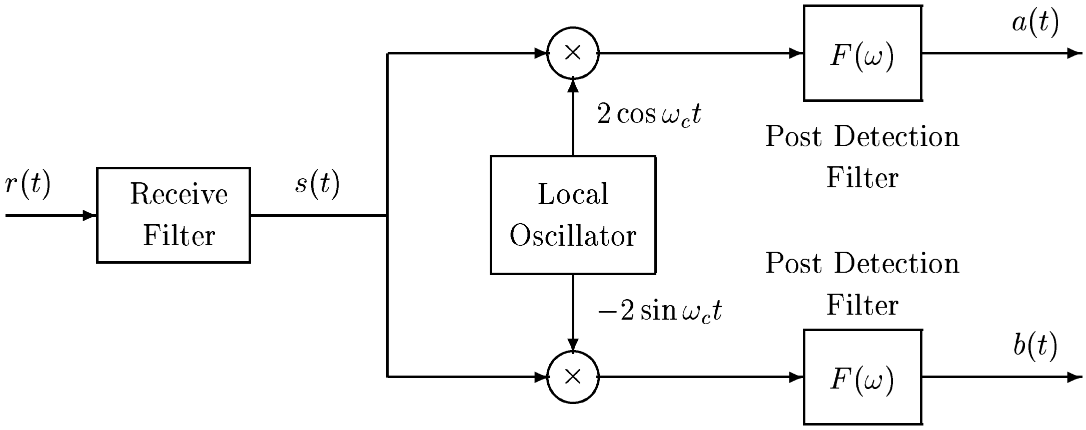
A Second Type of QAM Demodulator

Output of the Upper Product Modulator

$$s(t)2\cos\omega_c t = a(t) + a(t)\cos 2\omega_c t - b(t)\sin 2\omega_c t$$

Output of the Lower Product Modulator

$$-s(t)2\sin\omega_c t = b(t) - b(t)\cos 2\omega_c t - a(t)\sin 2\omega_c t$$



The components around $2\omega_c$ are eliminated by the post detection filters.

QAM Modulator Experiments

You will make a substantial part of a V.22bis calling modem transmitter.

- The V.22bis modem was designed for full duplex data transmission at 1200 or 2400 bits per second over ordinary dial-up 2-wire telephone lines.
- V.22bis uses a symbol rate of $f_s = 600$ symbols/sec (baud).
- *Full duplex* means that one modem transmits to a second and the second transmits to the first simultaneously, and the transmissions in both directions are independent.
- The modem that initiates the transmission by placing a call to the other modem is referred to as the *calling* or *originate* modem and the called modem is referred to as the *answer* modem.

Modulator Experiments (cont.)

Frequency division multiplexing is used to achieve the full duplex transmission.

- The calling modem transmits a QAM signal using a carrier frequency of 1200 Hz
- The answer modem transmits using a 2400 Hz carrier.
- The baseband shaping filters are specified to have a square-root of raised cosine frequency response with an excess bandwidth factor of $\alpha = 0.75$. Therefore, the spectrum of the calling modem is nominally confined to the band $675 \leq f \leq 1725$ Hz and the answer modem spectrum to $1875 \leq f \leq 2925$ Hz.
- Hybrids and bandpass filters are used to separate the transmit and receive signals at the receivers.

Making a QAM Transmitter (1)

Use your main C program to control the overall program flow. Consider doing all the numerically intensive computations like convolutions using assembly routines called from C. This is not necessary, but in any case, use -o3 optimization. Do the following:

1. Use a 9600 Hz sampling rate. With this sampling rate and a symbol rate of 600 baud, you will have to generate $9600/600 = 16$ output samples per symbol. You will be asked below to write the output samples to the McBSP0 data transmit register (DXR) with an interrupt service routine activated by the transmitter XRDY flag. You can generate the 600 Hz symbol rate timing by counting interrupts in the interrupt service routine.

Making a QAM Transmitter (2)

2. Implement the V.22bis scrambler defined on Slide 13-12. Set the initial shift register state to all 0's and use the input sequence $d_i = 1$ for all i. Check that your scrambler is working by computing an initial segment of the output sequence by hand and comparing it with your scrambler output.

   Your program should contain the options of generating two scrambled output bits per symbol for 1200 bps transmission or four scrambled output bits per symbol for 2400 bps transmission.

3. Implement the differential encoder shown in Slide 13-9. Your program should contain options for both the 1200 and 2400 bps modes. At 1200 bps the output of your function should be $(Y1_n, Y2_n, 0, 1)$ and at 2400 bps it should be $(Y1_n, Y2_n, Q3_n, Q4_n)$.

Making a QAM Transmitter (3)

4. Map the 4-bit differential encoder output to a constellation point by looking up the values for a_n and b_n in a table corresponding to Slide 13-8.

5. Now implement the modulator using passband shaping filters as shown in Slide 13-16. Generate 16 output samples per symbol resulting in a 9600 Hz output sampling rate.

   Use **C:\DIGFIL\SQRTRACO.EXE** along with the formulas for $h_I(t)$ and $h_Q(t)$ on Slide 13-14 to generate the impulse response samples of your inphase and quadrature passband shaping filters. The filter impulse responses should be limited to the time interval $[-3T, 3T]$ where T is the symbol period. Since you are making a calling modem transmitter, use a 1200 Hz carrier frequency.

Making a QAM Transmitter (4)

Notice that the symbol rotation shown at the input to the modulator of Slide 13-16 is not required in this case since

$$e^{j\omega_c nT} = e^{j2\pi n f_c/f_s} = e^{j2\pi n \frac{1200}{600}} = e^{j4\pi n} = 1$$

For testing purposes, also generate the baseband shaping filter coefficients for a raised cosine response with the α and duration of the passband filters. You can use the program **C:\DIGFIL\RASCOS.EXE** to generate the impulse response.

Use the interpolation filter bank method presented in Section 11.3 to generate the $L = 16$ output samples from the inphase and quadrature passband shaping filters each symbol period. Combine the filter outputs to form 16 samples of the modulated signal $s(t)$ for the next symbol period and write the samples to a “mailbox”.

Making a QAM Transmitter (5)

Using a Mailbox

- The samples put in the mailbox should be integers suitable for sending to the **left channel** of the codec and they should be scaled so that the codec output is limited to ±0.5 v. A symbol clock signal may be put in the right channel.
- The mailbox should be a 32-word array. One half of the array should contain the 16 output samples for the symbol currently being transmitted and the other half should contain the 16 new samples for the next symbol period.
- The halves of the array should be swapped after each symbol period.
- The mailbox can be implemented as a circular buffer.

Making a QAM Transmitter (6)

Using Interrupts to Output Samples

6. Write an interrupt service routine to load output samples into the McBSP0 DXR.

- The interrupt should be triggered by the XRDY flag of McBSP0.
- The routine should contain a pointer to the next output sample in the mailbox. It should write the sample to the DXR and then increment the pointer modulo 32.
- The interrupt service routine should also maintain a count of the number of interrupts that have occurred modulo 16 to provide the 600 Hz symbol timing.
- The input pointer should be set to (output + 16) mod 32 at the end of each symbol.

Testing Your Transmitter (1)

Test your transmitter to verify that it is operating properly by performing the following steps:

1. First select the 1200 bps option.
    - Clear the scrambler shift register and make its input $d_i = 1$ for all i.
    - Set the coefficients of the inphase passband shaping filter equal to those of the baseband raised cosine shaping filter designed for this experiment. Make the quadrature component zero by setting the coefficients of the quadrature passband shaping filter to zero.
    - Observe the eye diagram on the oscilloscope for this signal. Eye diagrams are discussed in Chapter 11. Explain the number of levels you observe in the eye diagram.

Testing Your Transmitter (2)

- **Generating a Sync Signal**
  Generate a sync signal for the eye diagram by sending a 600 Hz clock to the right channel of the codec. You can do this by sending a constant like 16000 to the right channel for the first eight samples in a baud and -16000 for the last eight samples.

2. Next make the inphase passband shaping filter coefficients zero and make the quadrature passband shaping filter equal to the baseband raised cosine filter. Observe the resulting eye pattern on the oscilloscope.

3. Select the 2400 bps option and repeat the previous step.

Testing Your Transmitter (3)

4. **Displaying the Signal Constellation** Display the baseband signal constellation by sending the inphase constellation point a_n, scaled appropriately, to the left codec channel for each of the 16 samples in a symbol and send the quadrature component b_n to the right channel for each of the 16 samples in a symbol. Attach the left and right outputs to two oscilloscope channels and set it to the x-y display option. First set your transmitter to the 1200 bps mode and you should observe the 4-point constellation. Then set your transmitter to the 2400 bps option and you should observe the 16-point constellation.

5. When your basic program flow is correct, put in the correct inphase and passband shaping filter coefficients. Observe the nature of the transmitted signal on the oscilloscope for the 1200 and 2400 bps options.

Testing Your Transmitter (4)

6. If a spectrum analyzer is available, measure the spectrum of the signals from Step 5 and sketch the results. Check that it has square-root of raised cosine shaping.
7. **Unscrambled 1's Input**
   Select the 1200 bps option and make the differential encoder input $(Q1_n, Q2_n) = (1, 1)$ for all n. The ITU-T standard calls this the *unscrambled binary 1's* sequence and it is used by the answer modem in one segment of the handshaking sequence.
   - The resulting sequence of transmitted constellation points continuously rotates by $-90°$. You should observe a periodic signal on the oscilloscope. Measure its fundamental frequency.
   - Also measure the spectrum of the transmitted signal with the spectrum analyzer.
   - Explain your results mathematically.

Testing Your Transmitter (5)

8. **Unscrambled 0's Input**
   Repeat the previous step with the input to the differential encoder set to all 0's. This causes continuous +90° phase shifts.

9. **The S1 or Dotting Sequence**
   Another pattern called the S1 sequence is also used during handshaking. This pattern uses the 1200 bps constellation and alternates between two points separated by 90°.
   - The two points are generated by making the differential encoder dibit inputs $(Q1_n, Q2_n)$ alternate between $(0, 0)$ which causes a +90° phase change and $(1, 1)$ which causes a −90° phase change.
   - The exact pair of points used depends on the initial value $(Y1_0, Y2_0)$ of the absolute quadrant and is not specified in the V.22bis standard.

- For example, the S1 sequence could alternate between the points (0001) and (0101) shown in Slide 13-8.
- Make your transmitter send the S1 sequence continuosly and observe the signal on the oscilloscope. Measure the spectrum with the spectrum analyzer. Determine the spectrum theoretically and compare your measured and theoretical results.

Testing the Transmitter with a Commercial Modem (Optional)

In this optional exercise, you will connect your transmitter to a Penril Alliance V.32 modem and test the V.22bis 2400 bps option using the 4-wire leased line configuration. Perform the following:

1. Before two modems begin transmitting customer data to each other, they must go through a handshake sequence
    - to agree on the transmission speed,

Testing with a Commercial Modem (2)

- to adjust automatic gain controls (AGC),
- to train their symbol clock,
- to lock carrier tracking loops,
- and to train their adaptive equalizers.

Program your transmitter to generate the following three segment startup sequence:

(a) First, send the S1 sequence described on Slide 13-31 for 100 ms.

(b) Second, send scrambled binary 1 using the 1200 bps constellation for 700 ms. That is, clear the scrambler shift register and make its input identically 1.

(c) Third, send scrambled binary 1 using the 2400 bps constellation for 200 ms.

After the startup sequence, continue to send scrambled binary 1 using the 2400 bps constellation.

Testing with a Commercial Modem (3)

2. Attach the Alliance diagnostic port on the back panel to the oscilloscope and set the scope for an X-Y display as described in the text. Place the Alliance in analog loop-back and observe the signal constellations for various data rates following the instructions in the text.
3. Now use the SPEED menu to set the MAX and MIN speeds to 2400 which will force the Alliance to be a V.22bis modem communicating at 2400 bps.
4. Attach the EVM D/A output to the LEASED LINE connector on the Alliance.
5. Make sure DTR (data terminal ready) is turned on as described in the text. Start your transmitter and observe the constellation display.

Testing with a Commercial Modem (4)

6. (Optional Exercise) Connect a bit-error rate test set to MCBSB1 as described in Chapter 10. Use the bit stream from the test set as the input to the scrambler of your transmitter. Attach another test set to the Alliance modem. Configure both test sets to generate the same pseudo-random sequence at 2400 bps and start bit error rate tests. There should be no errors if your transmitter is working correctly.

   You could also add noise to the transmitted samples in the DSP and observe the effect on the received constellation and plot a curve of the bit-error rate vs. signal-to-noise ratio.