Jordan University of Science & Technology

Electrical Engineering Department

Digital Communication Systems Lab

(EE552)

LABORATORY MANUAL

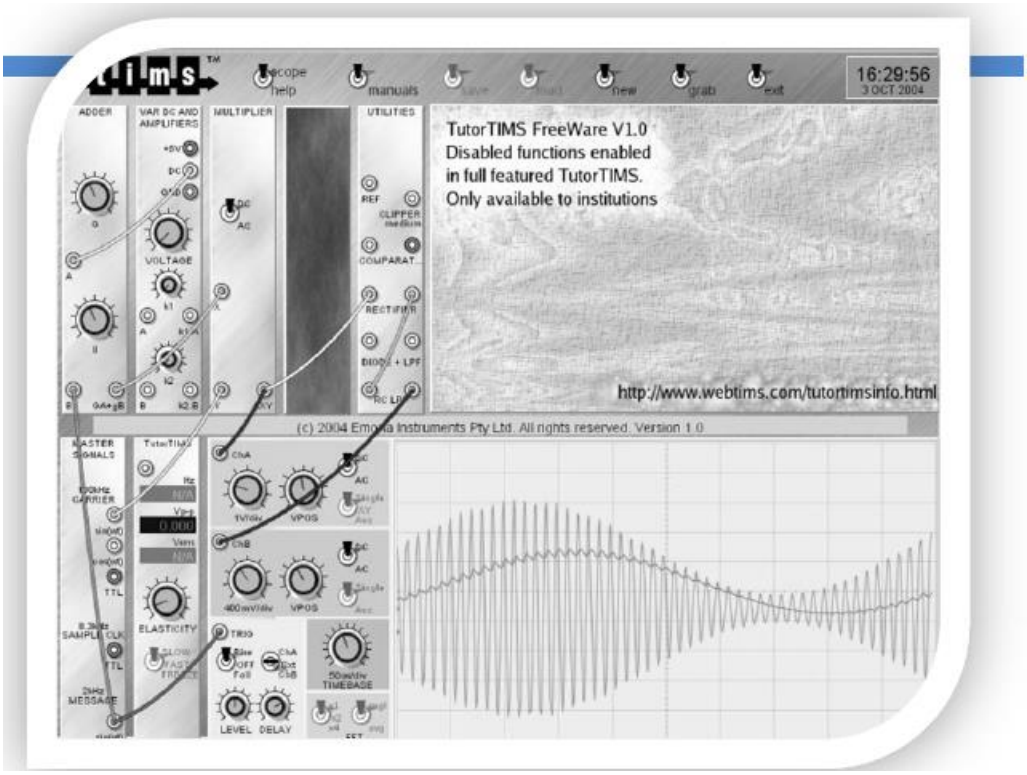


Table of Contents

Report Format and Laboratory Rules

In this laboratory students will work in teams of two. However, the lab reports will be written individually.

Prelab must be presented to your instructor at the beginning of the lab, and must include the following:

- Show the objectives of the experiment.
- Draw the block diagrams of each part, write the procedure in a clear format
- Draw and prepare the tables (if they exist) to be filled in at the lab.

Reports are due at the beginning of the lab period. The reports are intended to be a Complete documentation of the work done in preparation for and during the lab.

Please use the following format for your lab reports (Postlab).

- ***COVER PAGE***:

Laboratory Report Cover Sheet

Jordan University of Science & Technology
Electrical Engineering Department

EE 552
Digital Communication Lab

Experiment #:
Experiment Title:

Student name:
Lab partners:
Lab Section#:

Day & Date of Submission :

- ***OBJECTIVE***: Clearly state, in your own words, the objective of performing the lab.

- ***EQUIPMENT USED***: Indicate the equipment used to perform the experiment

- ***PROCEDURE***: Provide summary of the procedures used in the lab and draw the block diagrams which were constructed during the experiment.

- ***DATA AND DISCUSSIONS***: Provide a record of the data obtained during the experiment (The data should be presented in a clear manner), and provide a discussion about the data

(This is the most important part of the lab report)

- ***CONCLUSIONS***: The student should present conclusions which may be logically deduced from their data and observations.

Please pay attention to the following:

- Copying any prelab/postlab will result in a grade of 0.
- Attendance at your regularly scheduled lab period is required. An unexpected absence will result in loss of credit for your lab.
- Reports Due Dates: Reports are due one week after completion of the corresponding lab. A late lab report will have 10% of the points deducted for being one day late. If a report is2 days late, a grade of 0 will be assigned.
- Quizzes will be held in the first 5 minutes of each lab and it will be related to the new experiment.
- The lab has to be kept clean. All the components, wires, and chairs must be taken back to their places.

EXPERIMENT 1: Delta Modulation & Demodulation

EXPERIMENT 2: Delta-Sigma Modulation

Lab 1

EXPERIMENT 1: Delta Modulation & Demodulation

ACHIEVEMENTS: *an introduction to the basic delta modulator; to observe effects of step size and sampling clock rate change; slope overload and granular noise.*

PREREQUISITES: *some exposure to the principles of delta modulation in course work.*

ADVANCED MODULES: *DELTA MODULATION UTILITIES, WIDEBAND TRUE RMS METER*

PREPARATION

Principle of operation Delta modulation was introduced in the 1940s as a simplified form of pulse code modulation (PCM), which required a difficult-to-implement analog-to-digital (A/D) converter.

The output of a delta modulator is a bit stream of samples, at a relatively high rate (eg, 100 kbit/s or more for a speech bandwidth of 4 kHz) the value of each bit being determined according as to whether the input message sample amplitude has increased or decreased relative to the previous sample. It is an example of differential pulse code modulation (DPCM).

Block diagram

The operation of a delta modulator is to periodically sample the input message, to make a comparison of the current sample with that preceding it, and to output a single bit which indicates the sign of the difference between the two samples. This in principle would require a sample-and-hold type circuit.

De Jager (1952) hit on an idea for dispensing with the need for a sample and hold circuit. He reasoned that if the system *was* producing the desired output then this output could be sent back to the input and the two analog signals compared in a comparator. The output is a delayed version of the input, and so the comparison is in effect that of the current bit with the previous bit, as required by the delta modulation principle.

Figure 1 illustrates the basic system in block diagram form, and this will be the modulator you will be modelling.

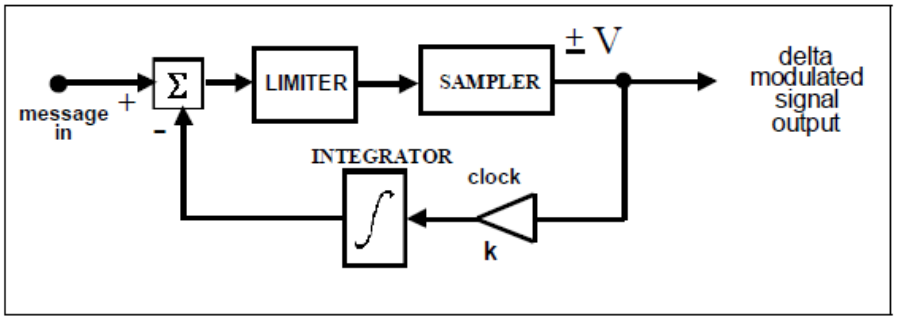


Figure 1: basic delta modulator

The system is in the form of a feedback loop. This means that its operation is not necessarily obvious, and its analysis non-trivial. But you can build it, and confirm that it does behave in the manner a delta modulator should.

The system is a continuous time to discrete time converter. In fact, it is a form of analog to digital converter, and is the starting point from which more sophisticated delta modulators can be developed.

The sampler block is clocked. The output from the sampler is a bipolar signal, in the block diagram being either ±V volts. This is the delta modulated signal, the waveform of which is shown in Figure 2. It is fed back, in a feedback loop, via an integrator, to a summer.

The integrator output is a sawtooth-like waveform, also illustrated in Figure 2. It is shown overlaid upon the message, of which it is an approximation.

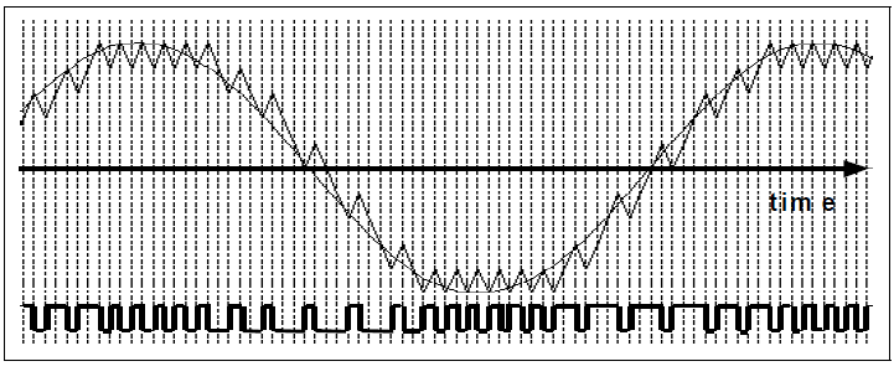


Figure 2: integrator output superimposed on the message with the delta modulated signal below

The sawtooth waveform is subtracted from the message, also connected to the summer, and the difference - an error signal - is the signal appearing at the summer Output. An amplifier is shown in the feedback loop. This controls the loop gain. In Practice it may be a separate amplifier, part of the integrator, or within the summer. It is used to control the size of the 'teeth' of the sawtooth waveform, in conjunction with the integrator time constant.

When analysing the block diagram of Figure 1 it is convenient to think of the summer having unity gain between both inputs and the output. The message comes in at a fixed amplitude. The signal from the integrator, which is a sawtooth approximation to the message, is adjusted with the amplifier to match it as closely as possible. You will be able to see this when you make a model of the system of Figure 1.

Step size calculation

In the delta modulator of Figure 1 the output of the integrator is a sawtooth-like approximation to the input message. The teeth of the saw must be able to rise (or fall) fast enough to follow the message. Thus the integrator time constant is an important parameter.

For a given sampling (clock) rate the step *slope* (volt/s) determines the *size* (volts) of the step within the sampling interval.

Suppose the amplitude of the rectangular wave from the sampler is ±V volt. For a change of input sample to the integrator from (say) negative to positive, the change of integrator output will be, after a clock period T:

$$output = \frac{2kVT}{RC} \ volt \qquad \text{..... 1}$$

Where ***k*** is the gain of the amplifier preceding the integrator (as in Figure 1).

Slope overload and granularity

The binary waveform illustrated in Figure 2 is the signal transmitted. This is the delta modulated signal.

The integral of the binary waveform is the sawtooth approximation to the message.

In the experiment entitled *Delta demodulation* (in this Volume) you will see that this sawtooth wave is the primary output from the demodulator at the receiver.

Lowpass filtering of the sawtooth (from the demodulator) gives a better approximation to the message. But there will be accompanying noise and distortion, products of the approximation process at the modulator.

The unwanted products of the modulation process, observed at the receiver, are of two kinds. These are due to ‘slope overload’, and ‘granularity’.

Slope overload

This occurs when the sawtooth approximation cannot keep up with the rate-of change of the input signal in the regions of greatest slope.

The step size is reasonable for those sections of the sampled waveform of small slope, but the approximation is poor elsewhere. This is 'slope overload', due to too small a step.

Slope overload is illustrated in Figure 3.

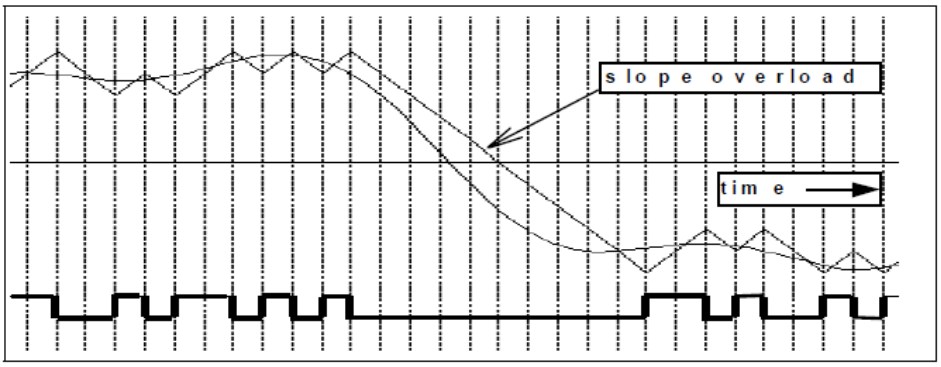


Figure 3: slope overload

To reduce the possibility of slope overload the step size can be increased (for the same sampling rate). This is illustrated in Figure 4. The sawtooth is better able to match the message in the regions of steep slope.

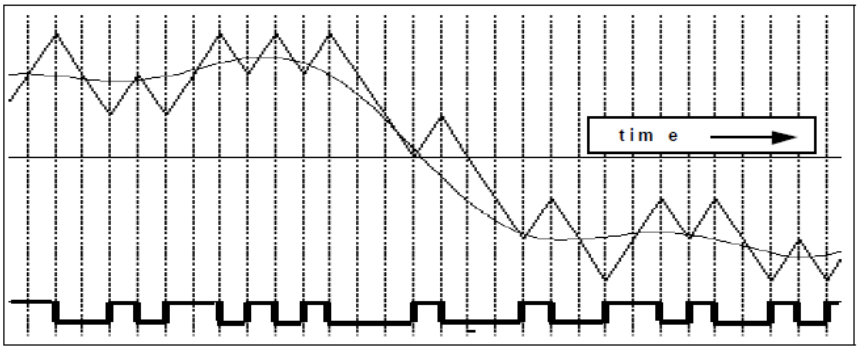


Figure 4: increased step size to reduce slope overload

An alternative method of slope overload reduction is to increase the sampling rate. This is illustrated in Figure 5, where the rate has been increased by a factor of 2.4 times, but the step is the same size as in Figure 3.

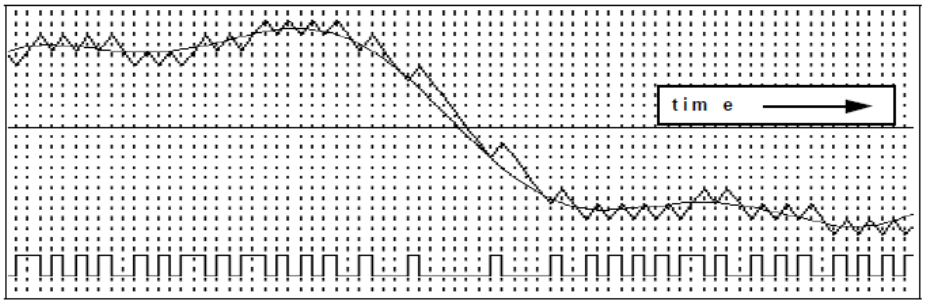


Figure 5: increased sampling rate to reduce slope overload

Granular noise

Refer back to Figure 3. The sawtooth follows the message being sampled quite well in the regions of small slope. To reduce the slope overload the step size is increased, and now (Figure 4) the match over the regions of small slope has been degraded.

The degradation shows up, at the demodulator, as increased quantizing noise, or 'granularity'.

Noise and distortion minimization

There is a conflict between the requirements for minimization of slope overload and the granular noise. The one requires an increased step size, the other a reduced step size. You should refer to your text book for more discussion of ways and means of reaching a compromise. You will meet an example in the experiment entitled *Adaptive delta modulation*

An optimum step can be determined by minimizing the quantizing error at the summer output, or the distortion at the demodulator output.

EXPERIMENT

Setting up

T1 *obtain and examine a DELTA MODULATOR UTILITIES module. Read about it in the TIMS Advanced Modules User Manual. Before plugging it in set the on-board switches to give an intermediate INTEGRATOR time constant (say SW2A to ON, and SW2B to OFF). Start with no division of the 100 kHz sample clock (front panel toggle switch up to 'CLK').*

T2 *plug in the ADDER and DELTA MODULATION UTILITIES module.*

T3 *use a sinewave to set both of the ADDER gains close to unity.* ***Do not change these for the duration of the experiment.***

T4 *use a sinewave to set both of the BUFFER AMPLIFIER gains to about unity (****they are connected in series to make a non-inverting amplifier****). Either one or both of these will be varied to make adjustments to the step size during the course of the experiment.*

T5 *patch up a model of Figure 1. This is shown in Figure 6. Use the 100 kHz TTL signal from the MASTER SIGNALS module as the clock for the SAMPLER, and the 2 kHz MESSAGE for the sinusoidal message to be sampled. The message (2.083 kHz) is a sub-multiple of the 100 kHz sample clock. This helps to obtain text-book like oscilloscope displays.*

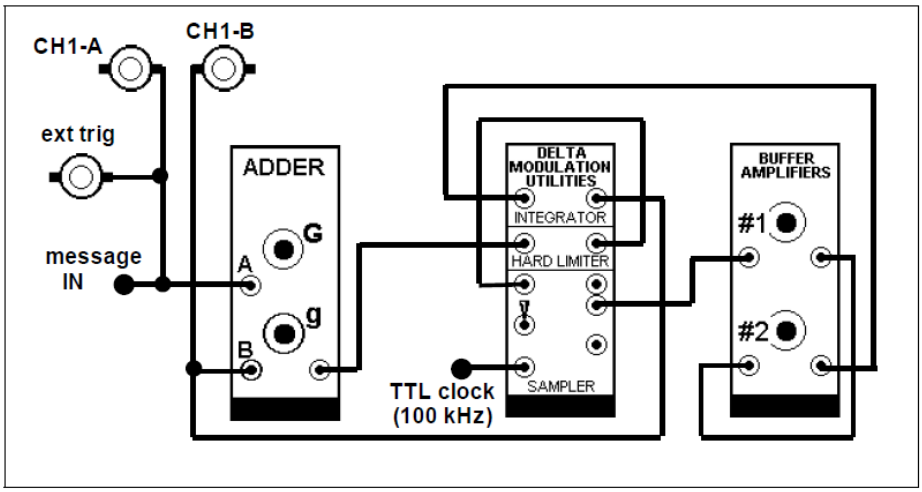


Figure 6: the delta modulator; a model of Figure 1

T6 *use the 2 kHz message as the 'ext. trig' signal to the oscilloscope. The signals of immediate interest are the two inputs to the SUMMER.*

You will now set up the modulator for 'acceptable performance'. This means that the INTEGRATOR output should be a reasonable approximation to the message at the input to the SUMMER (of Figure 1).

The *only adjustments you should make* during the course of the experiment are to:

1. ***The step size***: this can be varied in fixed steps with the INTEGRATOR time constant, or fine steps with the gain ***k*** of the amplifier (two cascaded BUFFER amplifiers) in the feedback loop (Figure 1).

2. ***the sampling clock rate***: with the front panel toggle switch of the DELTA MODULATOR UTILITIES module (100, 50, or 25 kHz*)*

T7 *Display the two inputs to the ADDER on CH1-A and CH1-B. These are the input message, and the INTEGRATOR output respectively.*

Remember that the INTEGRATOR waveform is required to be an approximation to the message. ***Adjust the gain k to achieve what you consider the 'best' match. You should have a display similar to that of Figure 2.***

You will notice that, despite the fact that the message is a sub-multiple of the clock rate, it is also necessary to fine-tune the oscilloscope sweep speed to obtain a totally stable oscilloscope display. This is through no fault of the oscilloscope - think about it!

T8 *Use CH2-A to look at the modulator output - that is, from the SAMPLER. Compare it with the INTEGRATOR output on CH1-B. Confirm the relationship between the two waveforms.*

T9 *vary the gain* ***k****, and watch the INTEGRATOR output (CH1-B) for signs of slope overload.*

T10 *re-adjust for 'moderate' slope overload. Increase and decrease the step size by means of the INTEGRATOR time constant (SW2A and SW2B on the DELTA MODULATION UTILITIES module circuit board). Confirm that the degree of slope overload changes as expected.*

T11 *vary the clock rate by* ***using*** *the front panel switch of the DELTA MODULATION UTILITIES module. Show that slope overload increases when the clock speed is halved, or decreases when the clock rate is doubled. Does the step size change when the clock changes?*

T12 *set an AUDIO OSCILLATOR to about 2 kHz, and use it for the message (And ext. trig signal), instead of the synchronous 2.083 kHz message.*

Leaving all other variables fixed, vary the message frequency and show its effect on the slope overload.

Whilst it is not easy to stabilize the display, it is still possible to see some consequences, including the onset of slope-overload. Record and explain your observations.

Delta Demodulation

The principle of the demodulator is shown in block diagram form in Figure 7 below.

It performs the reverse of the process implemented at the modulator in the vicinity of the sampler and integrator.

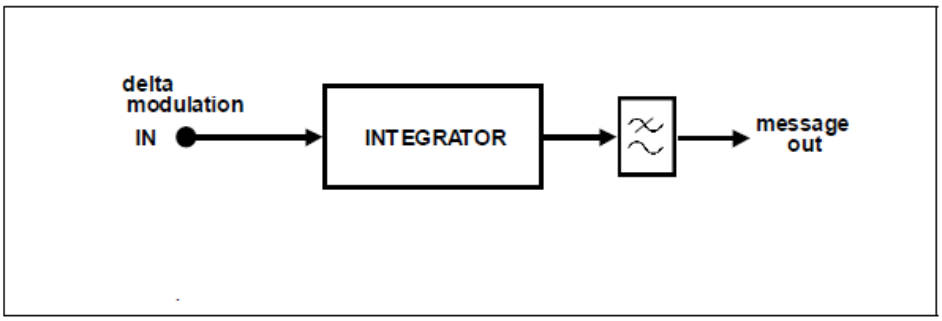


Figure 7: a demodulator for delta modulation

T13 *obtain and examine a DELTA DEMOD UTILITIES module.*

T14 *Model the demodulator of Figure 7. Set the time constant of the INTEGRATOR to the same value as selected in the modulator. Use the RC LPF in the DELTA DEMOD UTILITIES for the output filter, and set the front panel clock switch to match that at the modulator.*

T15 *confirm that the output of the demodulator low pass filter is a reasonable copy of the original message.*

EXPERIMENT 2: Delta-Sigma Modulation

ACHIEVEMENTS: *introduction to an important variation of the basic delta modulator (as used in compact disc players).*

PREREQUISITES: *completion of the experiments entitled* ***Delta modulation*** *and* ***Delta demodulation*** *in this Volume.*

ADVANCED MODULES: *DELTA MODULATION UTILITIES, DELTA DEMOD UTILITIES*

PREPARATION

It is assumed that you have been introduced to the principles of delta-sigma modulation in your course work, and have completed the experiment entitled *Delta modulation*.

Delta-sigma modulation 1 is an apparently simple variation of the basic delta modulation arrangement. Whilst it is easy to describe the variation (by way of the block diagram, for example), the implications of the change are not necessarily transparently obvious. You should refer to your course work, which presumably will have treated the theory at an appropriate level. Suffice to say that the delta-sigma modulator and demodulator combination finds application in the compact disk digital record player, where its properties are exploited to the full.

The nature of the variation can be seen by first reminding yourself of the configuration of the basic delta modulator, shown in block diagram form in Figure 1.

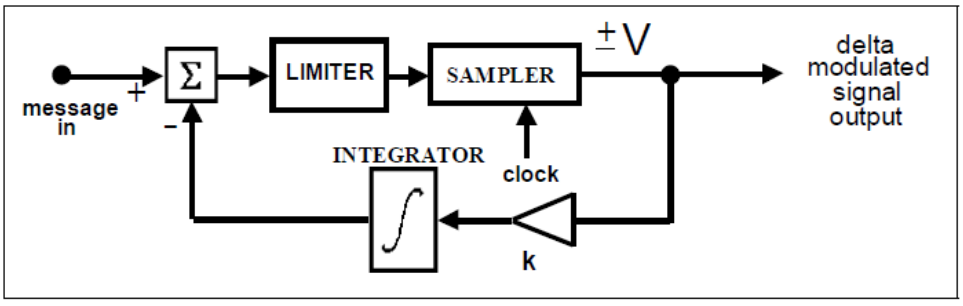


Figure 1: basic delta modulator

The delta-sigma modulator places an integrator between the message source and the

summer of the basic delta modulator.

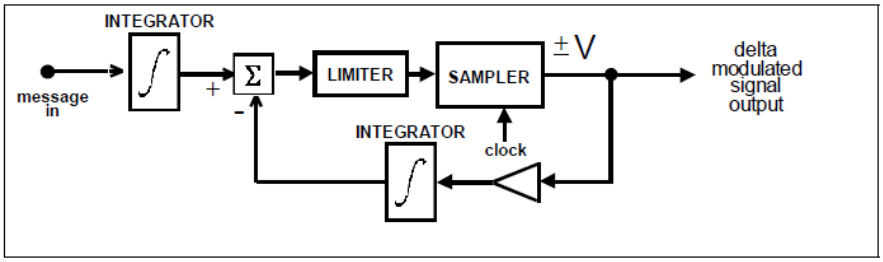


Figure 2: the delta-sigma modulator

The two integrators at each *input* to the linear summer can be replaced by a single integrator at the summer *output*. This simplified arrangement is shown in Figure 3

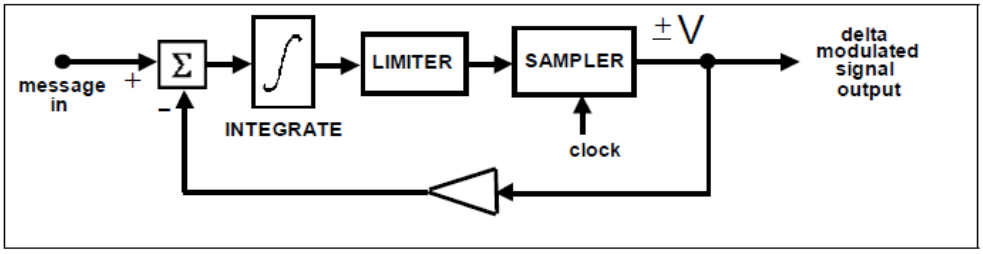


Figure 3: the delta-sigma modulator simplified

The integrator introduced at the input to the summer obviates the need for an integrator in the demodulator. Thus the demodulator can be a simple lowpass filter.

EXPERIMENT

Setting up

A model of the delta-sigma modulator block diagram of Figure3 is shown in Figure 4.

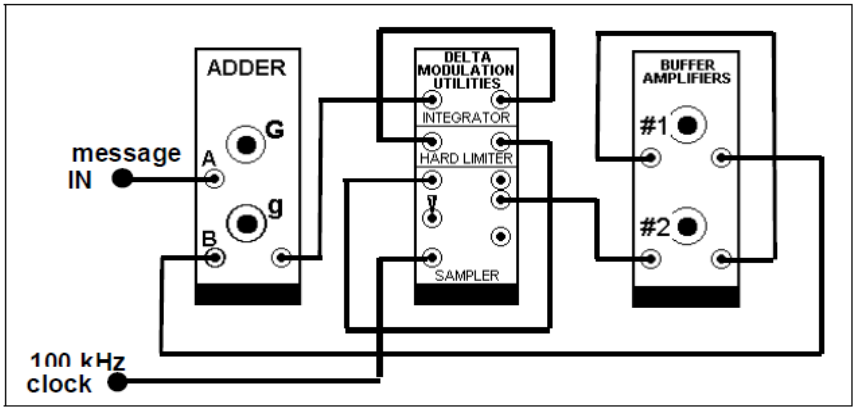


Figure 4: the delta-sigma modulator model

T1 *Patch together the complete delta-sigma modulator according to Figure 4.*

T2 *Adjust both ADDER gains to unity, and both BUFFER AMPLIFIER gains to unity. Throughout the experiment the gain* ***g*** *of the ADDER (acting as the SUMMER) will not be changed*

T3 *Use a low pass filter as a demodulator. Display the output on CH1-B*

You can now examine the behavior of the modulator under various conditions, and with different messages, as was done for the basic delta modulator in an earlier experiment.

T4 *Vary the gain* ***k****, and watch the output (CH1-B) for signs of slope overload, and repeat by varying the clock rate by* ***using*** *the front panel switch of the DELTA MODULATION UTILITIES module.*

T5 *Set an AUDIO OSCILLATOR to about 2 kHz, and use it for the message (And ext. trig signal), instead of the synchronous 2.083 kHz message. Leaving all other variables fixed, vary the message frequency and show its effect on the slope overload. Record your comments*

EXPERIMENT 3: PCM Encoding

EXPERIMENT 4: PCM Decoding

Lab 2

EXPERIMENT 3: PCM Encoding

ACHIEVEMENTS: *introduction to pulse code modulation (PCM) and the PCM ENCODER module. Coding of a message into a train of digital words in binary format.*

PREREQUISITES: *an understanding of sampling, from previous experiments, and of PCM from course work or a suitable text. Completion of the experiment entitled* ***Sampling with SAMPLE & HOLD*** *(in this Volume) would be a distinct advantage.*

ADVANCED MODULES: *PCM ENCODER*

PREPARATION

PCM

This is an introductory experiment to pulse code modulation - PCM. The experiment will acquaint you with the PCM ENCODER, which is one of the TIMS Advanced Modules. This module generates a PCM output signal from an analog input message.

In this experiment the module will be used in isolation; that is, it will not be part of a larger system. The formatting of a PCM signal will be examined in the time domain. A later experiment, entitled *PCM decoding* (in this Volume), will illustrate the recovery of the analog message from the digital signal.

In another experiment, entitled *PCM TDM* (within *Volume D2 - Further & Advanced Digital Experiments*), the module will be part of a system which will generate a two channel pulse code modulated time division multiplexed system (PCM TDM).

PCM encoding

The input to the PCM ENCODER module is an analog message. This must be constrained to a defined bandwidth and amplitude range.

The maximum allowable message bandwidth will depend upon the sampling rate to be used. The Nyquist criterion must be observed.

The amplitude range must be held within the ± 2.0 volts range of the TIMS ANALOG REFERENCE LEVEL. This is in keeping with the input amplitude limits set for all analog modules.

A step-by-step description of the operation of the module follows:

1. The module is driven by an external TTL clock.

2. The input analog message is *sampled* periodically. The *sample rate* is determined by the external clock.

3. The sampling is a *sample-and-hold* operation. It is internal to the module, and cannot be viewed by the user 1. What is held is the *amplitude* of the analog message *at the sampling instant.*

4. Each sample amplitude is compared with a finite set of amplitude levels. These are distributed (uniformly, for *linear* sampling) within the range ± 2.0 volts (the TIMS ANALOG REFERENCE LEVEL). These are the system *quantizing* levels.

5. Each quantizing level is assigned a *number*, starting from zero for the lowest (most negative) level, with the highest number being (L-1), where L is the available number of levels.

6. Each sample is *assigned* a digital (binary) code word representing the number associated with the quantizing level which is closest to the sample amplitude.
The number of bits 'n' in the digital code word will depend upon the number of quantizing levels. In fact, $n = \log_2(L)$.

7. The code word is *assembled into a time frame* together with other bits as may be required (described below). In the TIMS PCM ENCODER (and many commercial systems) a single extra bit is added, in the least significant bit position. This is alternately a *one* or a *zero*. These bits are used by subsequent decoders for frame synchronization.

8. The *frames* are transmitted serially. They are transmitted at the same rate as the samples are taken. The serial bit stream appears at the output of the module.

9. Also available from the module is a synchronizing signal FS ('frame synch').This signals the *end* of each data frame.

The PCM ENCODER module

Front panel features

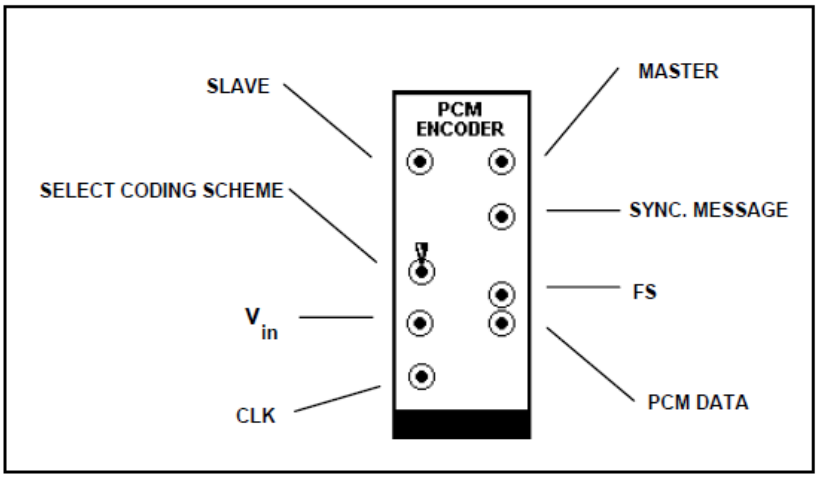


Figure 2: front panel layout of the PCM ENCODER

The front panel layout of the module is shown in Figure 2. Technical details are described in the TIMS *Advanced Modules User Manual.*

Note and understand the purpose of each of the input and output connections, and the three-position toggle switch. Counting from the top, these are:

• ***SLAVE***: not used during this experiment. Do *not* connect anything to this input.

• ***MASTER***: not used during this experiment. Do *not* connect anything to this output.

• ***SYNC. MESSAGE****:* periodic, 'synchronized', message. Either sinusoidal, or sinusoidal-like ('sinuous'), its frequency being a sub-multiple of the MASTER CLOCK (being any one of four frequencies selected by an on-board switch SW2). A message synchronized to the system clock is convenient for obtaining stable oscilloscope displays. Having a recognizable shape (but being more complex than a simple sine wave) gives a qualitative idea of distortion during the decoding process (examined in a later experiment). **See Table A-1 in the Appendix to this experiment for more details.**

• ***SELECT CODING SCHEME****:* a three-position toggle switch which selects the 4-bit or 7-bit encoding scheme of the analog samples; or (together with an onboard jumper connection) the companding scheme.

• ***FS***: frame synchronization, a signal which indicates the end of each data frame.

• ***Vin:****:* the analog signal to be encoded.

• ***PCM DATA***: the output data stream, the examination of which forms the major part of this experiment.

• ***CLK***: this is a TTL (red) input, and serves as the MASTER CLOCK for the module. Clock rate must be 10 kHz or less. For this experiment you will use the 8.333 kHz TTL signal from the MASTER SIGNALS module.

The TIMS PCM time frame

Each binary word is located in a *time frame*. The time frame contains eight *slots* of equal length, and is eight clock periods long. The slots, from first to last, are numbered 7 through 0. These slots contain the bits of a binary word. The least significant bit (LSB) is contained in slot 0.

The LSB consists of alternating *ones* and *zeros*. These are placed ('embedded') in the frame by the encoder itself, and cannot be modified by the user. They are used by subsequent decoders to determine the location of each frame in the data stream, and its length. See the experiment entitled *PCM decoding* (in this Volume).

The remaining seven slots are available for the bits of the binary code word. Thus the system is capable of a resolution of seven-bit maximum. This resolution, for purposes of experiment, can be reduced to four bits (by front panel switch). The 4- bit mode uses only five of the available eight slots - one for the embedded frame synchronization bits, and the remaining four for the binary code word (in slots 4, 3, 2, and 1).

EXPERIMENT

The only module required for this experiment is a TIMS PCM ENCODER.
It is not necessary, for this experiment, to become involved with *how* the PCM ENCODER module achieves its purpose. What will be discovered is *what* it does under various conditions of operation.

The module is capable of being used in two modes: as a stand-alone PCM encoder, for one channel, or, with modifications to the data stream, as part of a two-channel time division multiplexed (TDM) PCM system.

Operation as a single channel PCM encoder is examined in this experiment.

Before plugging the module in:

T1 *locate the on-board switch SW2. Put the LEFT HAND toggle DOWN and the RIGHT HAND toggle UP. This sets the frequency of a message from the module at SYNC. MESSAGE. This message is synchronized to a submultiple of the MASTER CLOCK frequency.*

Patching up

T2 *insert the module into the TIMS frame. Switch the front panel toggle switch to 4-BIT LINEAR.*

T3 *patch the 8.333 kHz TTL SAMPLE CLOCK from the MASTER SIGNALS module to the CLK input of the PCM ENCODER module.*

T4 *connect the* V_{in} *input socket to ground of the variable DC module.*

T5 *connect the frame synchronization signal FS to the oscilloscope ext. synch. Input.*

T6 *on CH1-A display the frame synchronization signal FS. Adjust the sweep speed to show three frame markers. These mark the* ***end*** *of each frame.*

T7 *display the CLK signal on CH2-A.*

T8 *record the number of clock periods per frame.*

T9 *on CH2-B display the PCM DATA from the PCM DATA output socket.* Make a sketch of this signal.

Your display should be similar to that of Figure 3 below, except that this shows five frames (too many frames on the oscilloscope display makes bit identification more difficult).

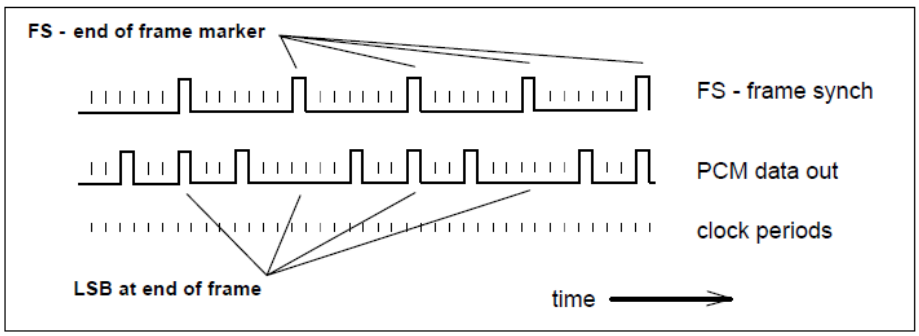


Figure 3: 5 frames of 4-bit PCM output for zero amplitude input

Knowing:
1. The number of slots per frame is 8
2. The location of the least significant bit is coincident with the end of the frame
3. The binary word length is four bits
4. The first three slots are 'empty' (in fact filled with zeros, but these remain unchanged under all conditions of the 4-bit coding scheme)

Then:
T10 *identify the binary word in slots 4, 3, 2, and 1.*

Quantizing levels for 4-bit linear encoding

You will now proceed to determine the quantizing/encoding scheme for the 4-bit linear case.

T11 *remove the ground connection, and connect the output of the VARIABLE DC module to Vin. Sweep the DC voltage slowly backwards and forwards over its complete range, and note how the data pattern changes in discrete jumps.*

T12 *slowly increase the amplitude of the DC input signal until there is a sudden change to the PCM output signal format. Record the format of the new digital word, and the input amplitude at which the change occurred.* ***Tabulate your results***

> ***If you have a WIDEBAND TRUE RMS METER module use this to monitor the DC amplitude at Vin - otherwise use the oscilloscope (CH1-B).***
> ***Adjust Vin to its maximum negative value. Record the DC voltage and the pattern of the 4-bit binary number.***

T13 *continue this process over the full range of the DC supply.*

4-bit data format

From measurements made so far you should be able to answer the questions:

- What is the sampling rate?
- What is the frame width?
- What is the width of a data bit?
- What is the width of a data word?
- How many quantizing levels are there?
- are the quantizing levels uniformly (linearly) spaced?

7-bit linear encoding

T14 *change to 7-bit linear encoding by use of the front panel toggle switch.*
It would take a long time to repeat all of the above Tasks for the 7-bit encoding scheme. Instead:

T15 *make sufficient measurements so that you can answer all of the above questions in the section titled* ***4-bit data format*** *above. Making one or two assumptions (such as ?) you should be able to deduce the coding scheme used.*

Periodic messages

Although the experiment is substantially complete, you may have wondered why a periodic message was not chosen at any time. Try it.

T16 *Take a periodic message from the SYNC. MESSAGE socket. This was set as the second Task.*

T17 *Adjust the oscilloscope to display the message. Record its frequency and shape. Check if these are compatible with the Nyquist criterion; adjust the amplitude if necessary with one of the BUFFER AMPLIFIERS.*

T18 *Now look at the PCM DATA output. Synchronize the oscilloscope (as previously) to the frame (FS) signal. Display two or three frames on CH1-A, and the PCM DATA output on CH2-A.*

You will see that the data signal reveals very little. It consists of many overlaid digital words, all different.

One would need more sophisticated equipment than is assumed here (a digital analyzer, a storage oscilloscope, ability to capture a single frame, and so on) to deduce the coding and quantizing scheme from such an input signal.

APPENDIX

For a MASTER CLOCK of 8.333 kHz, Table A-1 below gives the frequencies of the synchronized message at the SYNC. MESSAGE output for the setting of the on-board switch SW2.

For other clock frequencies the message frequency can be calculated by using the 'divide by' entry in the Table.

These messages are periodic, but not necessarily sinusoidal in shape. The term 'sinuous' means sine-like.

LII toggle	*RII toggle*	*divide clock by*	*freq with 8.333kHz clock*	*approx. ampl. and waveform*
UP	UP	32	260.4 Hz	0.2 V_{pp} sine
DOWN	UP	64	130.2 Hz	2.0 V_{pp} sine
UP	DOWN	128	65.1 Hz	4.0 V_{pp} sinuous
DOWN	DOWN	256	32.6 Hz	4.0 V_{pp} sinuous

Table A-1

EXPERIMENT 4: PCM Decoding

ACHIEVEMENTS: *decoding of a PCM signal. Determination of the quantizing scheme used at the encoder. Message reconstruction. Introduction to companding; comparison of 7-bit linear with 4-bit companded PCM.*

PREREQUISITES: *completion of the experiment entitled* ***PCM encoding*** *in this Volume. An appreciation of the principles of companding.*

ADVANCED MODULES: *A PCM ENCODER and a PCM DECODER (version 2 preferable).*

PREPARATION

Signal source

The signal to be decoded 1 in this experiment will be provided by you, using the PCM ENCODER module as set up in the experiment entitled *PCM encoding*. The format of the PCM signal is described there. You should have already completed that experiment.

Clock synchronization

A clock synchronization signal will be stolen from the encoder.

Frame synchronization

Automatic

In the PCM DECODER module there is circuitry which automatically identifies the location of each frame in the serial data stream. To do this it collects groups of eight data bits and looks for the repeating pattern of alternate ones and zeros placed there (embedded) by the PCM ENCODER in the LSB position.

It can be shown that such a pattern cannot occur elsewhere in the data stream provided that the original bandlimited analog signal is sampled at or below the Nyquist rate.

When the embedded pattern is found an 'end of frame' synchronization signal FS is generated, and made available at the front panel.

The search for the frame is continuously updated. Why?

Under noisy conditions (not relevant for this particular experiment) the reliability of the process will depend upon the size of the group of frames to be examined. This can be

set by the on-board switch SW3 of the PCM DECODER module. See *Table A-1* in the Appendix to this experiment for details.

Stolen

Frame synchronization can also be achieved, of course, by 'stealing' the synchronization signal, FS, from the PCM ENCODER module. Use of this signal would assume that the clock signal to the PCM DECODER is of the correct phase.
This is assured in this experiment, but would need adjustment if the PCM signal is transmitted via a bandlimited channel (see Tutorial Question **0**). Hence the embedded frame synchronization information.

PCM decoding

The PCM DECODER module is driven by an external clock. This clock signal is synchronized to that of the transmitter. For this experiment a 'stolen' clock will be used. The source of frame timing information has been discussed above.

Upon reception, the PCM DECODER:

1. Extracts a frame synchronization signal FS from the data itself (from the embedded alternate ones and zeros in the LSB position), or uses an FS signal stolen from the transmitter (see above).

2. Extracts the binary number, which is the coded (and quantized) amplitude of the sample from which it was derived, from the frame.

3. Identifies the quantization level which this number represents.

4. Generates a voltage proportional to this amplitude level.

5. Presents this voltage to the output Vout. The voltage appears at Vout for the duration of the frame under examination.

6. Message reconstruction can be achieved, albeit with some distortion, by lowpass filtering. A built-in reconstruction filter is provided in the module.

Encoding

At the encoder the sample-and-hold operation (before encoding) is executed periodically. It produces a rectangular pulse form 2. Each pulse in the waveform is of *exactly* the same amplitude as the message *at the sampling instant*.

But *it is not possible* to recover a *distortion less* message from these samples. They are *flat top*, rather than *natural* samples.

Call this the sampling distortion.

At the encoder the amplitude of this waveform was then *quantized*. It is still a rectangular pulsed waveform, but the amplitude of each pulse will, in general, be in error by a small amount. Call this waveform s(t).

This was examined in the experiment entitled *Sampling with SAMPLE & HOLD* (in this Volume), to which you should refer.

Decoding

The voltage at Vout of the decoder is *identical with* s(t) above. The decoder itself has introduced no distortion of the received signal.
But s(t) is already an inexact version of the sample-and-hold operation at the encoder. This will give rise to *quantization distortion* as well as the *sampling*

The TIMS PCM DECODER module

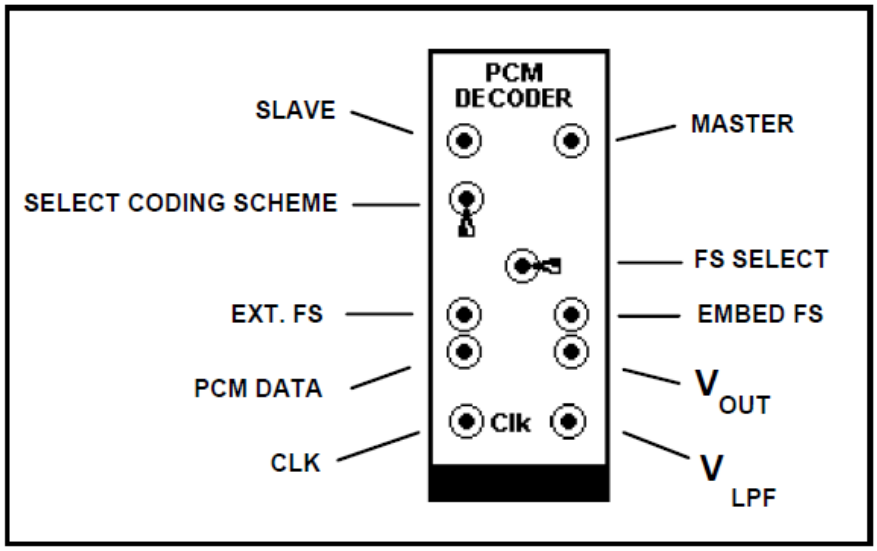


Figure 3: front panel layout of the PCM DECODER

A TIMS PCM DECODER module will be used for decoding.

The front panel of this module is shown in Figure 3.

Note and understand the purpose of the input and output connections, and the toggle switches. Counting from the top, these are:

• ***SLAVE***: not used during this experiment. Do *not* connect anything to this input.
• ***MASTER***: not used during this experiment. Do *not* connect anything to this output.

• ***SELECT CODING SCHEME***: a three position toggle which selects the coding scheme used by the signal to be decoded

• ***FS SELECT:*** a two-position toggle switch which selects the method of obtaining the frame synchronization signal (FS) either external at (EXT.FS), or derived internally from the embedded information in the received PCM itself (EMBED FS).

• ***EXT. FS:*** connect an external frame sync. Signal here if this method of frame synchronization is to be used.

• ***EMBED FS:*** if the frame synch. Signal is derived internally from the embedded information, it is available for inspection at this output.

• ***PCM DATA:*** the PCM signal to be decoded is connected here.

• ***VOUT:*** the decoded PCM signal.

• ***CLK:*** this is a TTL (red) input, and serves as the MASTER CLOCK for the module. Clock rate must be 10 kHz or less. For this experiment you will use the 8.333 kHz TTL signal from the MASTER SIGNALS module.

EXPERIMENT

The transmitter (encoder)

A suitable source of PCM signal will be generated using a PCM ENCODER module. This module was examined in the experiment entitled *PCM encoding*.

You should set it up before patching up the demodulator.

T1 *before plugging in PCM ENCODER module, set the toggles of the on-board SYNC MESSAGE switch SW2. Set the left hand toggle DOWN, and the right hand toggle UP. This selects a 130 Hz sinusoidal message, which will be used later. Now insert the module into the TIMS system.*

T2 *use the 8.333 kHz TTL signal from the MASTER SIGNALS module for the CLK.*

T3 *select, with the front panel toggle switch, the 4-bit LINEAR coding scheme.*

T4 *synchronize the oscilloscope 'externally' to the frame synchronization signal at FS. Set the sweep speed to 0.5 ms/cm (say). This should show a few frames on the screen.*

T5 *connect CH1-A of the SCOPE SELECTOR to the PCM OUTPUT of the PCM ENCODER.*

T6 *we would like to recognize the PCM DATA out signal. So choose a 'large' negative DC for the message (from the VARIABLE DC module).*

From previous work we know the corresponding code word is '0000', so only the embedded alternating '0' and '1' bits (for remote FS) in the LSB position should be seen.

T7 *vary the DC output and show the appearance of new patterns on CH1-A. Draw up a table relating input voltages to the appeared patterns*
When finished, return the DC to its maximum negative value (control fully anti-clockwise).

The receiver (decoder)

T8 *connect the PCM DATA output signal from the transmitter to the PCM DATA input of the receiver.*

T9 *use the front panel toggle switch to select the 4-bit LINEAR decoding scheme (to match that of the transmitter).* *Steal' an 8.333 kHz TTL clock signal from the transmitter and connect it to the CLK input.*

T10 *in the first instance 'steal' the frame synchronization signal FS from the transmitter by connecting it to the frame synchronization input FS of the receiver. At the same time ensure that the FS SELECT toggle switch on the receiver is set to EXT. FS.*

T11 *connect CH2-A to the sample-and-hold output of the PCM DECODER.*

T12 *slowly vary the DC output from the VARIABLE DC module back and forth over its complete range. Observe the behavior of the two traces. The input to the encoder moves continuously. The output from the decoder moves in discrete steps. These are the 16 amplitude quantizing steps of the PCM ENCODER.*

You are observing the source of quantizing noise. The output can take up only one of 16 predetermined values.

T13 *draw up a table relating input to output voltages.*

T14 *compare the quantizing levels just measured with those determined in the experiment entitled*

A periodic message

It was not possible, when examining the PCM ENCODER in the experiment entitled *PCM encoding*, to see the sample-and-hold waveform within the *encoder*. But you Have just been looking at it (assuming perfect decoding) at the output of the *decoder*. With a periodic message its appearance may be more familiar to you.

T15** change to a periodic message 3 by connecting the SYNC MESSAGE of the PCM ENCODER, **via** a BUFFER AMPLIFIER, to its input V_{in}. An amplitude of 2 $V_{p\text{-}p}$ is suitable.* ***Observe and record the signal at CH2-A.

***T16** change the coding scheme from 4-bit to 7-bit. That is, change the front panel toggle switch of **both** the PCM ENCODER **and** the PCM DECODER from 4-bit to 7-bit. Observe, record, and explain the change to the waveform on CH2-A.*

EXPERIMENT 5: PRBS Generation

EXPERIMENT 6: Eye Patterns

Lab 3

EXPERIMENT 5: PRBS GENERATION

ACHIEVEMENTS: *Introduction to the pseudo random binary sequence (PRBS) generator*; time *domain viewing: snap shot and eye patterns; two generator synchronization*

PREREQUISITES*: None*

EXTRA MODULES*: A second SEQUENCE GENERATOR,*

PREPARATION

Digital messages

In analog work the standard test message is the sine wave, followed by the two-tone signal for more rigorous tests. The property being optimized is generally signal-to-noise ratio (SNR). Speech is interesting, but does not lend itself easily to mathematical analysis, or measurement.

In digital work a binary sequence, with a known pattern of '1' and '0', is common. It is more common to measure bit error rates (BER) than SNR, and this is simplified by the fact that known binary sequences are easy to generate and reproduce.

A common sequence is the pseudo random binary sequence:

Random binary sequences

The output from a pseudo random binary sequence generator is a bit stream of binary pulses; i.e.., a sequence of 1's (HI) or 0' s (LO), of a *known* and *reproducible* pattern.

The bit *rate,* or number of bits per second. Is determined by the frequency of an external *clock,* which is used to drive the generator. For each clock period a single bit is emitted from the generator; either at the '1' or '0' level, and of a width equal to the clock period. For this reason the external clock is referred to as a *bit clock.*

For a long sequence the 1's and O's are distributed in a (pseudo) random manner.

The sequence pattern *repeats* after a defined number of clock periods. In a typical generator the *length* of the sequence may be set to 2^n clock periods, where n is an integer. In the TIMS SEQUENCE GENERATOR (which provides two, independent sequences, X and Y) the value of n may be switched to one of three values, namely 2,

5, or 11. There are *two* switch positions for the case n = 5, giving different patterns. The SYNCH output provides a reference pulse generated once per sequence repetition period.

This is *the start-of-sequence* pulse. It is *invaluable as a trigger source* for an oscilloscope.

Viewing

There are two important methods of viewing a sequence in the time domain.

The snapshot

A short section, about 16 clock periods of a TTL sequence, is illustrated in Figure 1 below.

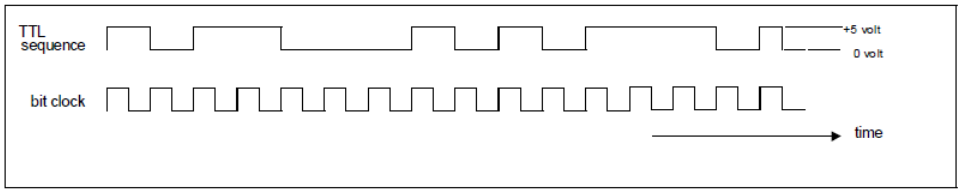


Figure 1: a sequence of length 16 bits

Suppose the output of the generator which produced the TTL sequence, of which this is a part, was viewed with an oscilloscope, with the horizontal sweep triggered by the display itself.

The display will not be that of Figure I above! Of course not, for how would the oscilloscope know which section of the display was wanted?

Consider just what the oscilloscope might show!

Specific sections of a sequence *can* be displayed on a general purpose oscilloscope, but the sequence generator needs to provide some help to do this.

As stated above, it gives a *start-of-sequence* pulse at the *beginning* of the sequence. This can be used to *start* (trigger) the oscilloscope sweep. At the end of the sweep the oscilloscope will wait until the next start-of-sequence is received before being triggered to give the next sweep.

Thus the beginning 'n' bits of the sequence are displayed, where 'n' is determined by the sweep speed.

For a sequence length of many-times-n bits, there would be a long delay between sweeps. The persistence of the screen of a general purpose oscilloscope would be too

short to show a steady display, so it will blink. You wi1l see the effect during the experiment.

The eye pattern

A long sequence is useful for examining 'eye patterns'. These are defined and Applications

One important application of the PRBS is for supplying a known binary sequence. This is used as a test signal (message) when making bit error rate (BER) measurements.

For this purpose a perfect *copy* of the *transmitted* sequence is required at the receiver, for direct comparison with the *received* sequence. This perfect copy is obtained from a second, identical, PRBS generator.

The second generator requires:

1. Bit clock information, so that it runs at the same rate as the first
2. A method of aligning its output sequence with the received sequence. Due to transmission through a bandlimited channel, it will be delayed in time with respect to the sequence at the transmitter.
3.

EXPERIMENT

The 'snapshot' display

Examine a SEQUENCE GENERATOR module, and read about it in the *TIMS User Manual.*

A suitable arrangement for the examination of a SEQUENCE GENERATOR is illustrated in Figure 2.

Notice that the length of the sequence is controlled by the settings of a DIP switch. SW2, *located on the circuit board.* See the Appendix to this experiment for details.

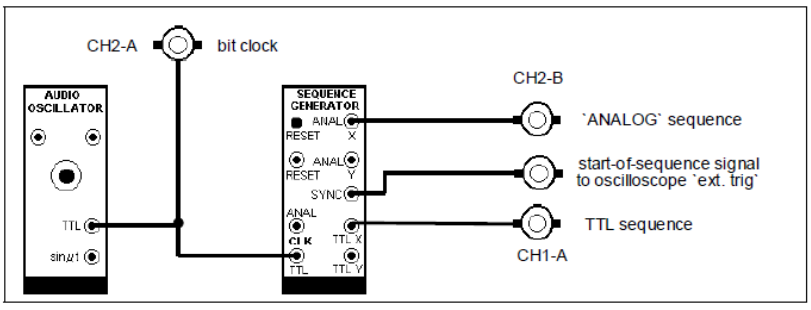


Figure 2: examination of a SEQUENCE GENERATOR

T1 *Before inserting the SEQUENCE GENERATOR set the on-board DIP switch SW2 to generate a short sequence. Then patch up the model of Figure 2 above. Set the AUDIO OSCILLATOR, acting as the bit clock, to about 2 kHz. Set the oscilloscope sweep speed to suit; say about 1 ms/cm.*

T2 *Observe the TTL sequence on CH1-A.*

T3 *Display the start-of-sequence SYNC signal from the SEQUENCE GENERATOR on CH1-B.*

T4 *Increase the sequence length by re-setting the on-board switch SW2. Re- establish synchronization using the start-of-sequence SYNC signal connected to the 'ext. trig' of the oscilloscope. Notice the effect upon the display.*

T5 *Have a look with your oscilloscope at a yellow analog output from the SEQUENCE GENERATOR.*

Notice: The DC offset has been removed, and the amplitude is now suitable for processing by analog modules (eg. by a filter representing an analog channel -see the experiment entitled The noisy channel model in this Volume). Observe also that the polarity has been reversed with respect to the TTL version. This is just a consequence of the internal circuitry.

Band limiting

The displays you have seen on the oscilloscope are probably as you would have expected them to be! That is, either 'HI' or 'LO' with sharp, almost invisible, transitions between them. This implies that there was no band limiting between the signal and the viewing instrument.

If transmitted via a lowpass filter, which could represent a bandlimited (baseband) channel, then there will be some modification of the shape, as viewed in the time domain.

For this part of the experiment you will use a TUNABLE LPF to limit, and vary, the bandwidth. Because the sequence will be going to an analog module it will be necessary to select an 'analog' output from the SEQUENCE GENERATOR

T6 *Select a short sequence from the SEQUENCE GENERATOR*

T7 *Connect an analog version of the sequence (YELLOW) to the input of a TUNABLE LP F.*

T8 *On the front panel of the TUNABLE LPF set the toggle switch to the WIDE position. Obtain the widest bandwidth by rotating the TUNE control fully clockwise.*

T9 *With the oscilloscope still triggered by the 'start-of-sequence' SYNC signal, observe both the filter input and output on separate oscilloscope channels. Adjust the gain control on the TUNABLE LPF so the amplitudes are approximately equal.*

T10 *Monitor the filter corner frequency, by measuring the CLK signal from the TUNABLE LPF with the FREQUENCY COUNTER**(divide by 880(normal) or 360 (wide) for details see TIMS User Manual). *Slowly reduce the bandwidth, and compare the difference between the two displays.*

Notice that, with reducing bandwidth:

a) Identification of individual bits becomes more difficult

b) There is an increasing delay between input and output

Remember that the characteristics of the filter will influence the results of the last Task.

Two generator alignment

In an experiment entitled *BER measurement in the noisy channel* (within *Volume D2 - Further & Advanced Digital Experiments)* you will find out *why* it is important to be able to align two sequences. In this experiment you will find out *how* to do it.

Two SEQUENCE GENERATOR modules may be coupled so that they deliver two identical, *aligned*, sequences.

- That they should deliver the *same sequence* it is sufficient that the generator circuitry be identical
- That they be at the *same rate* it is necessary that they share a common bit clock
- That they be *aligned* requires that they start at the same time.

TIMS SEQUENCE GENERATOR modules (and those available commercially) have in built facilities to simplify the alignment operation. One method will be examined with the scheme illustrated in block diagram form in Figure 3 below.

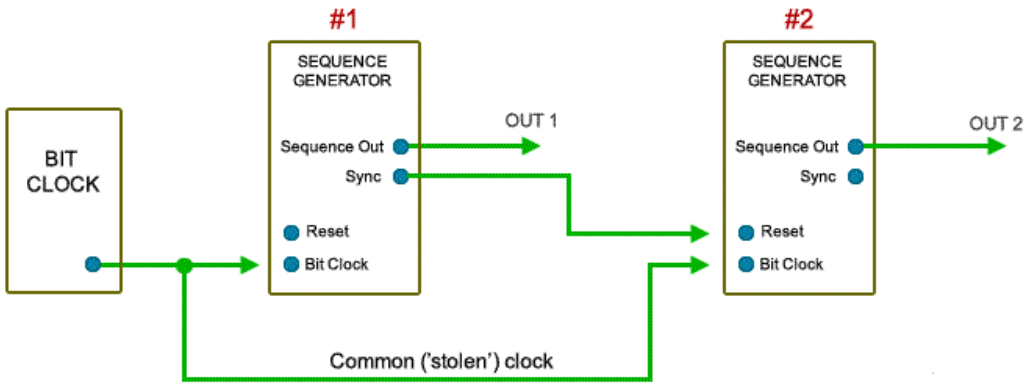


Figure 3: aligning two identical generators

The scheme of Figure 3 is shown modeled with TIMS in Figure 4 below.

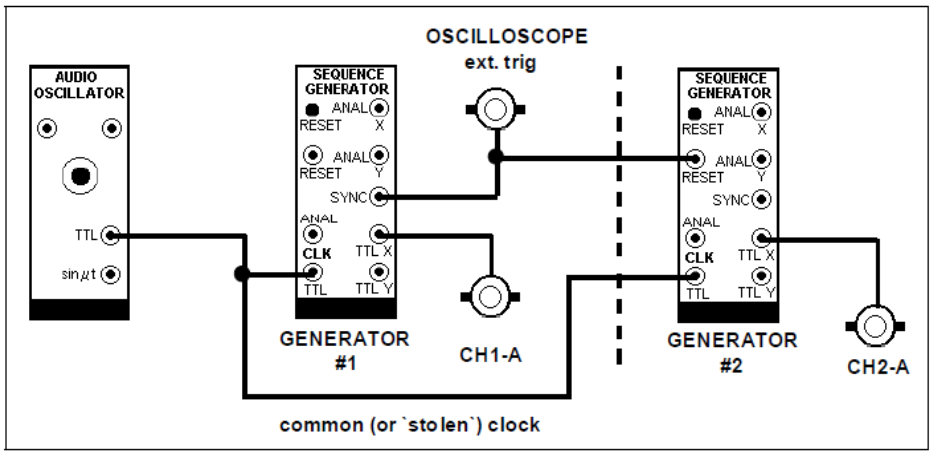


Figure 4: TIMS model of the block diagram of Figure 3.

You will now investigate the scheme. Selecting short sequences will greatly assist during the setting-up procedures, by making the viewing of sequences on the oscilloscope much easier.

T11 *Before plugging in the SEQUENCE GENERATOR modules, set them both to the same short sequence.*

T12 *Patch together as above, but omit the link from the 'GENERATOR #1 SYNC to 'GENERATOR #2' RESET. Do not forget to connect the 'start-of- sequence' SYNC signal of the GENERATOR #1 to the 'ext. trig' of the oscilloscope.*

T13 *Press the 'GENERATOR #2' RESET push button several times. Observe on the oscilloscope that the two output sequences are synchronized in time but the data bits*

do not line-up correctly. Try to synchronize the sequences manually by repeating this exercise many times. It is a hit- and-miss operation, and is likely to be successful only irregularly.

T14 *Connect the SYNC of the 'GENERATOR #1' to the RESET of the 'GENERATOR #2'. Observe on the oscilloscope that the two output sequences are now synchronized in time and their data are aligned.*

T15 *Break the synchronizing path between the two generators. What happens to the alignment?*

Once the two generator are aligned, **they will remain aligned**, even after the alignment link between them is broken. The bit clock will keep them in step

The above scheme has demonstrated a method of aligning two generators, and was seen to perform satisfactorily. But it was in a somewhat over simplified environment.

What if the two generators had been separated some distance, with the result that there was a delay between sending the SYNC pulse from GENERATOR #1 and its reception at GENERATOR #2?

The sequences would be offset by the time delay

In other words, the sequences would *not* be aligned.

APPENDIX

PRBS Generator - sequence length

The length of the sequences from the SEQUENCE GENERATOR can be set with the DIP switch SW2 located *on the circuit board.*

See Table A-1 below.

LH toggle	RH toggle	n	Sequence length
UP	UP	5	32
UP	DOWN	8	256
DOWN	UP	8	256
DOWN	DOWN	11	2048

Table A-1: on-board switch SW2 settings

There are two sequences of length 256 bits. These sequences are different.

EXPERIMENT 6: EYE PATTERNS

ACHIEVEMENTS: *Understanding the Nyquist I criterion, transmission rates via band limited channels; comparison of the 'snap shot' display with the 'eye patterns '.*

PREREQUISITES: *Some acquaintance with basic notions of digital transmission*

ADVANCED MODULES: *BASEBAND CHANNEL FILTERS*

PREPARATION

Pulse Transmission

It is well known that, when a signal passes via a bandlimited channel it will suffer waveform distortion. As an example, refer to Figure I. As the data rate increases the waveform distortion increases, until transmission becomes impossible.

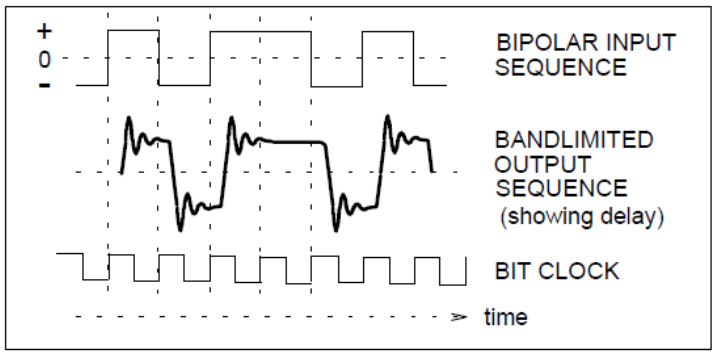


Figure 1: waveforms before and after moderate bandlimiting

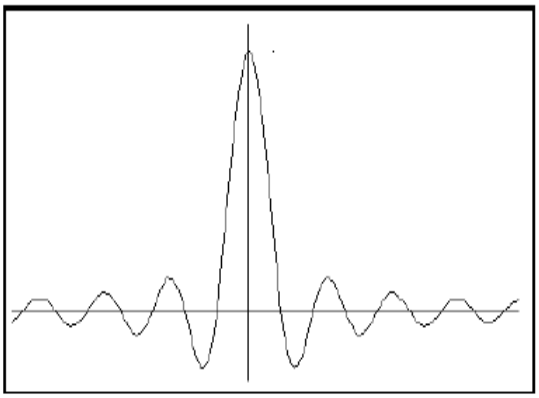
In this experiment you will be introduced to some important aspects of pulse transmission which are relevant to digital and data communication applications.

Issues of interest include:

- In the 1920s Harry Nyquist proposed a clever method now known as Nyquist's first criterion, that makes possible the transmission of telegraphic signals over channels with limited bandwidth without degrading signal quality .This idea has withstood the test of time. It is very useful for digital and data communications.

The method relies on the exploitation of pulses that look like sin(x)/x - see the Figure opposite.

The trick is that zero crossings always fall at equally spaced points. Pulses of this type are known as 'Nyquist I' (there is also Nyquist II and III).



- In practical communication channels distortion causes the dislocation of the zero crossings of Nyquist pulses, and results in *intersymbol interference* (ISI) I. Eye patterns provide a practical and very convenient method of assessing the extent of ISI degradation. A major advantage of eye patterns is that they can be used 'on-line' in real-time. There is no need to interrupt normal system operation.
- The effect of ISI becomes apparent at the receiver when the incoming signal has to be 'read' and decoded; i.e.., a detector decides whether the value at a certain time instant is, say, 'HI' or 'LO' (in a binary decision situation). A decision error may occur as a result of noise. Even though ISI may not itself cause an error in the absence of noise, it is nevertheless undesirable because it decreases the margin relative to the decision threshold, i.e.., a given level of noise that may be harmless in the absence if ISI, may lead to a high error rate when ISI is present.
- Another issue of importance in the decision process is *timing jitter.* Even if there is no ISI at the nominal decision instant, timing jitter in the reconstituted bit clock results in decisions being made too early or too late relative to the ideal point. As you will discover in this experiment. Channels that are highly bandwidth efficient are more sensitive to timing jitter.

Maximum transmission rate assessment

This is what is going to be done:

1. First, set up a pseudorandom sequence. To start you will use the shortest available sequence, so that you can easily observe it with an oscilloscope. Very long sequences are not easy to observe because the time elapsed between trigger pulses is too long. The oscilloscope will be triggered to the start of sequence signal. The display has been defined as a 'snap shot'.
2. Next you will pass this sequence through a selection of filters. Three are available in the BASEBAND CHANNEL FILTERS module, and a fourth will be the TUNABLE LPF module. You will observe the effect of the filters on the shape of the sequence, at various pulse rates.
3. Then the above observations will be repeated, but this time the oscilloscope will be triggered by the bit clock. Giving what is defined as an *eye pattern.*

4. Finally you will compare the performance of the various cases in terms of achievable transmission rate and 'eye opening'.

EXPERIMENT

T1 *Set up the model of Figure 2. The AUDIO OSCILLATOR serves as the bit clock for the SEQUENCE GENERATOR. A convenient rate to start with is 2 kHz. Select CHANNEL # I. Select a short sequence (both toggles of the on-board switch SW2 Up).*

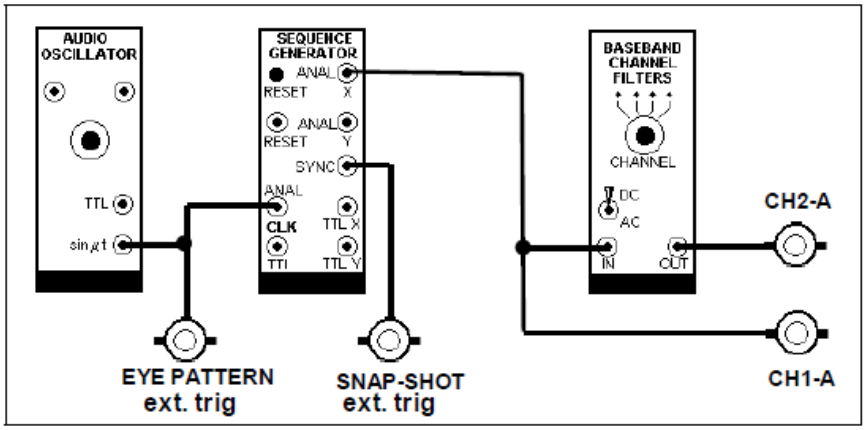


Figure 2: viewing snap shots and eye patterns

T2 *Synchronize the oscilloscope to the 'start-of-sequence ' synchronizing signal from the SEQUENCE GENERATOR. Set the sweep speed to display between 10 and 20 sequence pulses (say 1 ms/cm). This is the 'snap shot' mode. Both traces should be displaying the same-picture, since CHANNEL #1 is a 'straight through' connection.*

The remaining three channels (#2, #3, and #4) in the BASEBAND CHANNEL FILTERS module represent channels having the *same* slot bandwidth (40 db stopband attenuation at 4 kHz), but otherwise different transmission characteristics, and, in particular, different 3 dB frequencies. Graphs of these characteristics are shown in Appendix A.
You should also prepare a TUNABLE LPF to use as a fourth channel, giving it a 40 dB attenuation at 4 kHz. To do this:

T3 *Using a sinusoidal output from an AUDIO OSCILLATOR as a test input:*

a. *Set the TUNE and GAIN controls of the TUNABLE LPF fully clockwise. Select the NORM bandwidth mode.*
b. *Set the AUDIO OSCILLATOR to a frequency of, say, 1 kHz. This is well within the current filter passband.*
c. *Note the output amplitude on the oscilloscope.*
d. *Increase the frequency of the AUDIO OSCILLATOR to 4 kHz.*

Snap-shot assessment

Now it is your task to make an assessment of the maximum rate, controlled by the frequency of the AUDIO OSCILLATOR, at which a sequence of pulses can be transmitted through each filter before they suffer unacceptable distortion. The criterion for judging the maximum possible pulse rate will be your opinion that you can recognize the output sequence as being similar to that at the input.

It is important to remember that the four filters have the same *slot bandwidth* (i.e.., 4 kHz, where the attenuation is 40 dB) but different 3 *dB bandwidths.*

T4 *Record your assessment of the maximum practical data rate through each of the four channels.*

At the very least your report will be a record of the four maximum transmission rates. But it is also interesting to compare these rates with the characteristics of the filters. Perhaps you might expect the filter with the widest passband to provide the highest acceptable transmission rate?

Eye pattern assessment

Now you will repeat the previous exercise, but, instead of observing the sequence as a single trace, you will use *eye patterns,* the set-up will remain the same except for the oscilloscope usage and sequence length.

T5 *Change the oscilloscope synchronizing signal from the start-of-sequence SYNC output of the SEQUENCE GENERATOR to the sequence bit clock. Increase the sequence length (both toggles of the on-board switch SW2 DOWN).*

T6 *Select CHANNEL* #3. *Use a data rate of about* 2 *kHz. You should have a display on CH2-A similar to that of Figure* 3 *below.*

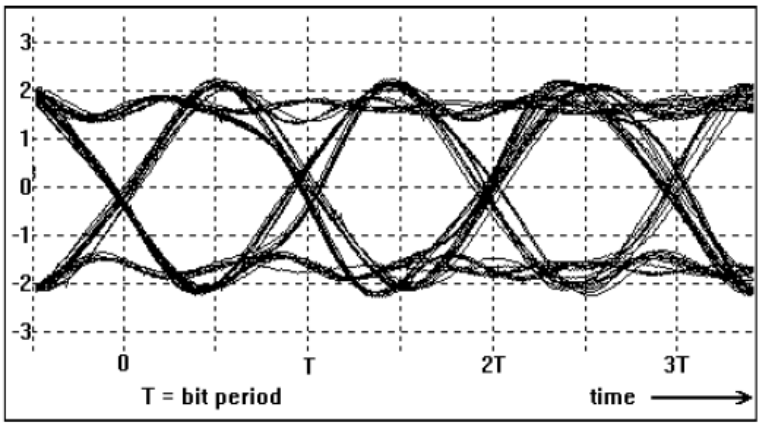


Figure 3: a 'good' eye pattern

T7 *Increase the data rate until the eye starts to close. Figure* 4 *shows an eye not nearly as clearly defined as that of Figure* 3.

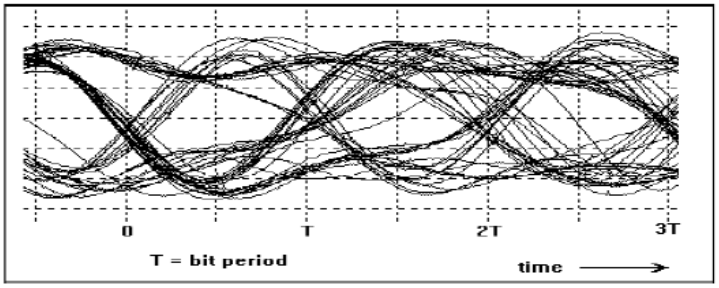


Figure 4: compare with Figure 3; a faster data rate

T8 *Take some time to examine the display, and consider what it is you are looking at! There is one 'eye' per bit period. Those shown* in *Figure* 3 *are considered to be 'wide open'. But as the data rate increases the eye begins to close.*

T9 *Determine the highest data rate for which you consider you would always be able to make the correct decision (HI or LO). Note that the actual moment to make the decision will be the same for all bits, and relatively easy to distinguish. Record this rate for each of the four filters.*

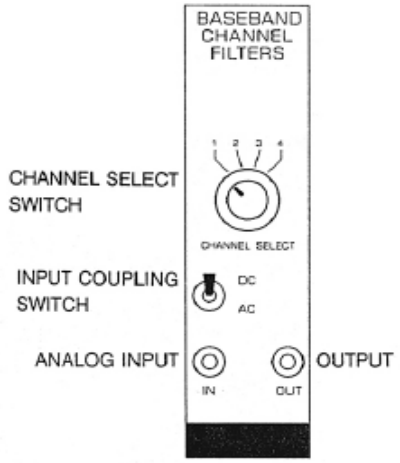
You have now seen two different displays, the snapshot and the eye pattern.

It is generally accepted that the eye pattern gives a better indication of the appropriate instant the HI or LO decision should be made, and its probable success, than does the snapshot display. Do you agree?

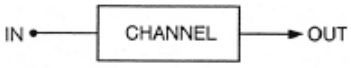
APPENDIX

Baseband Channel Filters

Four switch selectable, baseband channels are provided, comprising three different filters and one straight-through connection. Each of the three filters has a stop-band frequency of near 4kHz.



FRONT PANEL



BLOCK DIAGRAM

USE

Only one channel may be selected and used at a time. Note that each of the four channels may be AC or DC coupled by front panel toggle switch.

CHANNEL CHARACTERISTICS

Before using any of these four channels in experiments, each channel should be characterised by actual measurement of amplitude and phase responses. As a minimum, the cut-off and stop-band frequencies should be measured, using the AUDIO OSCILLATOR and TRUE RMS METER modules or an oscilloscope.

COMPARISONS

AMPLITUDE AND PHASE VERSUS FREQUENCY

It is useful to compare the amplitude and phase response of each channel with the 7th order elliptic TUNEABLE LOWPASS FILTER module (a standard module from of the BASIC MODULE SET). Compare against the same cut-off frequency by adjusting the TUNEABLE LOWPASS FILTER's cut-off frequency to match each channel's cut-off frequency.

EYE DIAGRAMS

Observing the EYE DIAGRAMS of digital data signals passing through the above selection of filters will illustrate each filter's (hence, channel's) performance.

BASIC SPECIFICATIONS

Input coupling AC or DC, channels 1 to 4

Channel responses

Channel 1 straight-through

Channel 2 Butterworth, 7th order

Channel 3 Bessel, 7th order

Channel 4 OpFil Linear Phase *, 7th order

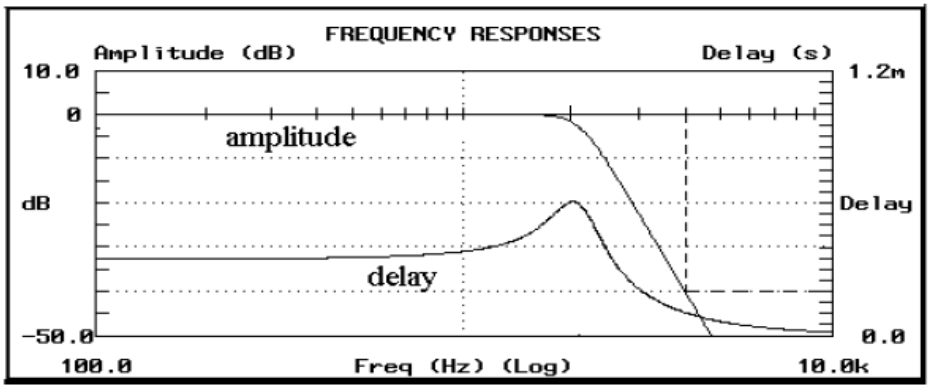
Stop-band attenuation approx 40dB, 4kHz

Passband ripple 0.5dB

* OpFil Linear Phase filter is a proprietary filter design having a sharp roll-off characteristic with a linear phase response in the passband. This filter was designed by Optimum Filters Pty Ltd, Sydney, Australia.

SWITCH POS #2

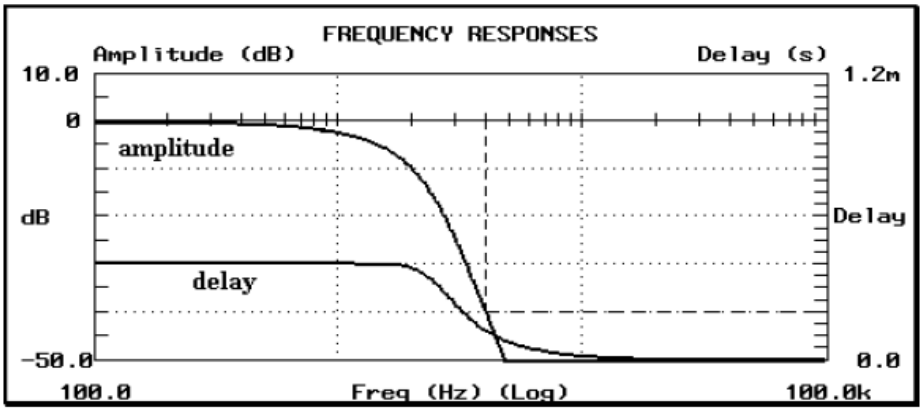
This filter is selected with the front panel switch in position 2



response	monotonic falling
passband	-1 dB at 1.88 kHz
stopband	-40 dB at 4.0 kHz

SWITCH POS #3

This filter is selected with the front panel switch in position 3

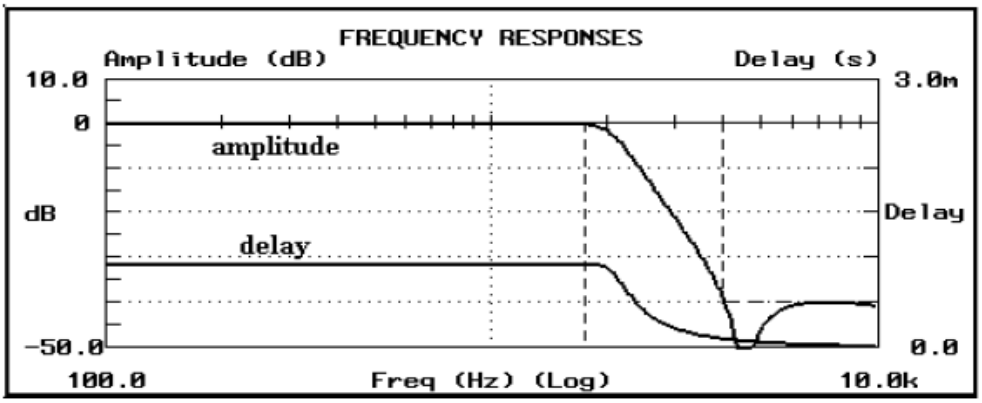


response	monotonic falling
passband edge	-1 dB at 620 Hz
stopband	-40 dB at 4.0 kHz

SWITCH POS #4

This filter is selected with the front panel switch in position 4

It exhibits an equiripple ('flat') group delay response over the complete passband and into the transition band.



passband ripple	0.1 dB
passband edge	1.75 kHz
stopband attenuation	40 dB
slotband	DC to 4 kHz
delay ripple	10 µs peak-to-peak
delay bandwidth	DC to 1.92 kHz

EXPERIMENT 7: Line Coding and Decoding.

EXPERIMENT 8: Bit Clock Regeneration

Lab 4

EXPERIMENT 7: Line Coding

ACHIEVEMENTS: *Familiarity with the properties of the LINE-CODE ENCODER and LINE-CODE DECODER modules, and the codes they generate.*

PREREQUISITES*: An appreciation of the purpose behind line coding.*

EXTRA MODULES: *LINE-CODE ENCODER and LINE-CODE DECODER.*

PREPARATION

This 'experiment' is tutorial in nature, and serves to introduce two new modules.

In your course work you should have covered the topic of line coding at whatever level is appropriate for you. TIMS has a pair of modules, one of which can perform a number of line code transformations on a binary TTL sequence. The other performs decoding.

Why line coding?

There are many reasons for using line coding. Each of the line codes you will be examining offers one or more of the following advantages:

Spectrum shaping and relocation without modulation or filtering. This is important in telephone line applications, for example, where the transfer characteristic has heavy attenuation below 300 Hz.

Bit clock recovery can be simplified.

DC component can be eliminated; this allows AC (capacitor or transformer) coupling between stages (as in telephone lines). Can control baseline wander (baseline wander shifts the position of the signal waveform relative to the detector threshold and leads to severe erosion of noise margin).

Error detection capabilities.

Bandwidth usage; the possibility of transmitting at a higher rate than other schemes over the same bandwidth.
At the very least the LINE-CODE ENCODER serves as an interface between the TTL level signals of the transmitter and those of the analog channel. Likewise, the LINE-CODE DECODER serves as an interface between the analog signals of the channel and the TTL level signals required by the digital receiver.

The modules

The two new modules to be introduced are the LINE-CODE ENCODER and the LINE-CODE DECODER.

You will not be concerned with how the coding and decoding is performed. You should examine the waveforms, using the original TTL sequence as a reference.

In a digital transmission system line encoding is the final digital processing performed on the signal before it is connected to the analog channel, although there may be simultaneous band limiting and wave shaping.

Thus in TIMS the LINE-CODE ENCODER accepts a TTL input, and the output is suitable for transmission via an analog channel.

At the channel output is a signal at the TIMS ANALOG REFERENCE LEVEL, or less. It could be corrupted by noise. Here it is re-generated by a *detector*. Finally the TIMS LINE-CODE DECODER module accepts the output from the DETECTOR and decodes it back to the binary TTL format.

Preceding the line code encoder may be a source encoder with a matching decoder at the receiver. These are included in the block diagram of Figure 1, which is of a typical baseband digital transmission system. It shows the disposition of the LINE- CODE ENCODER and LINE-CODE DECODER. All band limiting is shown concentrated in the channel itself, but could be distributed between the transmitter, channel, and receiver.

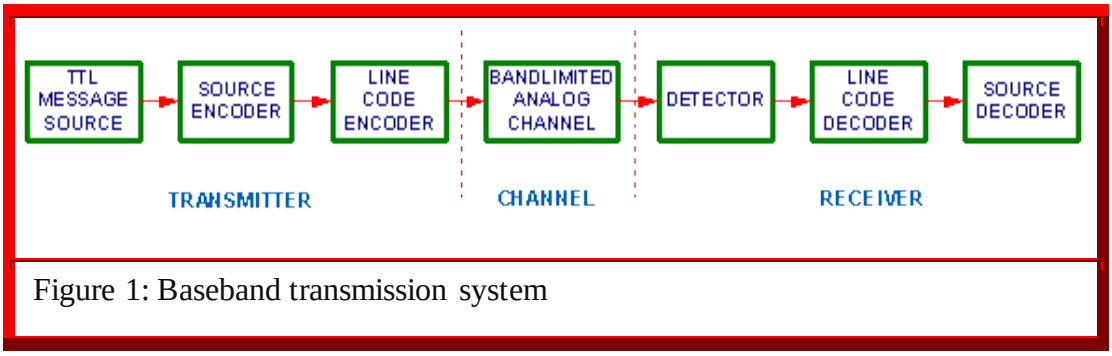


Figure 1: Baseband transmission system

The LINE-CODE ENCODER serves as a source of the system bit clock. It is driven by a *master clock* at 8.333 kHz (from the TIMS MASTER SIGNALS module). It divides this by a factor of four, in order to derive some necessary internal timing signals at a rate of 2.083 kHz. This then becomes a convenient source of a 2.083 kHz TTL signal for use as the *system bit clock*.

Because the LINE-CODE DECODER has some processing to do, it introduces a time delay. To allow for this, it provides a re-timed clock if required by any further digital processing circuits (eg, for decoding, or error counting modules).

Terminology

- The word *mark*, and its converse *space*, often appear in a description of a binary waveform. This is an historical reference to the mark and space of the

telegraphist. In modern day digital terminology these have become HI and LO, or '1' and '0', as appropriate.
- *Unipolar signaling:* where a '1' is represented with a finite voltage V volts, and a '0' with zero voltage. This seems to be a generally agreed-to definition.
- Those who treat *polar* and *bipolar* as identical define these as signaling where a '1' is sent as +V, and '0' as -V. They append AMI when referring to three- level signals which use +V and -V alternately for a '1' , and zero for '0' (an alternative name is pseudo ternary).

You will see the above usage in the *TIMS Advanced Modules User Manual,* as well as in this text.

However, others make a distinction. Thus:

- *Polar signaling:* where a '1' is represented with a finite voltage +V volts, and a '0' with -V volts.
- *Bipolar signaling:* where a '1' is represented alternately by +V and -V, and a '0' by zero voltage.
- The term 'RZ' is an abbreviation of 'return to zero'. This implies that the particular waveform will return to zero for a finite part of each data '1' (typically half the interval). The term 'NRZ' is an abbreviation for 'non-return to zero', and this waveform will not return to zero during the bit interval representing a data '1 '.
- The use of 'L' and 'M' would seem to be somewhat illogical (or inconsistent) with each other. For example, see how your text book justifies the use of the 'L' and the 'M' in NRZ-L and NRZ-M.
- Two sinusoids are said to be antipodal if they are 180° out of phase.

Available line codes

For a TTL input signal the following output formats are available from the LINE-CODE ENCODER.

NRZ-L

Non return to zero - level (bipolar): this is a simple scale and level shift of the input TTL waveform.

NRZ-M

Non return to zero - mark (bipolar): there is a transition at the beginning of each '1', and no change for a '0'. The 'M' refers to 'inversion on mark'. This is a differential code. The decoder will give the correct output independently of the polarity of the input.

UNI-RZ

Uni-polar - return to zero (uni-polar): there is a half-width output pulse if the input is a '1'; no output if the input is a '0'. This waveform has a significant DC component.

BIP-RZ

Bipolar return to zero (3-level): there is a half-width +ve output pulse if the input is a '1'; or a half-width -ve output pulse if the input is a '0'. There is a return-to-zero for the second half of each bit period.

RZ-AMI

Return to zero -alternate mark inversion (3-level): there is a half-width output pulse if the input is a '1'; no output if the input is a '0'. This would be the same as UNI-RZ. But, *in addition,* there is a polarity inversion of every alternate output pulse.

Bi Φ-L

Bi-phase - level (Manchester): bipolar ±V volts. For each input '1' there is a transition from +V to -V in the middle of the bit-period. For each input '0' there is a transition from -V to +V in the middle of the bit period.

DICODE-NRZ

Di-code non-return to zero (3-level): for each transition of the input there is an output pulse, of opposite polarity from the preceding pulse. For no transition between input pulses there is no output.

The codes offered by the line-code encoder are illustrated in Figure 2 below. These have been copied from the *Advanced Module User's Manual,* where more detail is provided.

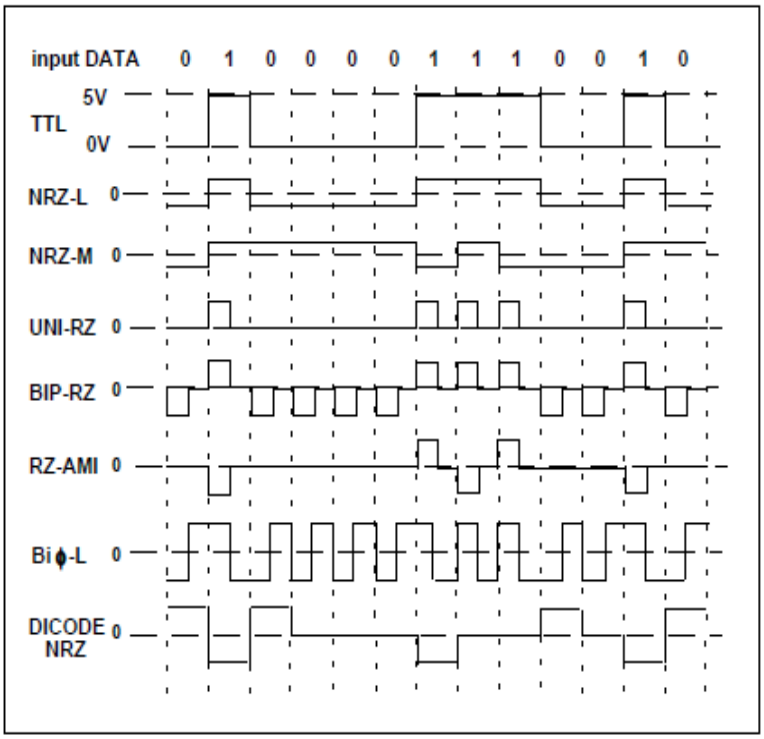


Figure 2: TIMS line codes

The output waveforms, apart from being encoded, have all had their amplitudes adjusted to suit a TIMS analog channel (not explicitly shown in Figure 2).

When connected to the Input of the LINE-CODE DECODER these waveforms are decoded back to the original TTL sequence.

EXPERIMENT

Figure 3 shows a simplified model of Figure 1. There is no source encoding or decoding, no baseband channel, and no detection. For the purpose of the experiment this is sufficient to confirm the operation of the line code modules.

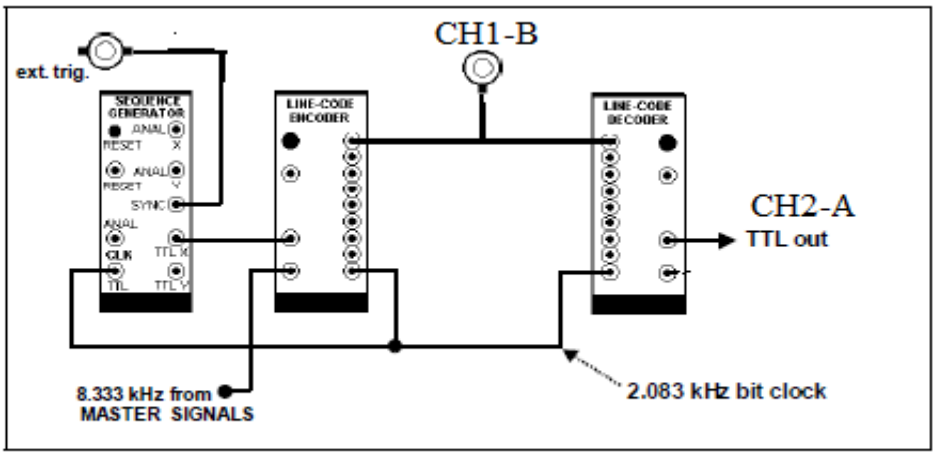


Figure 3: simplified model of Figure 1

Procedure

T1 *Set up the model of Figure* 3. *Select a short sequence (both toggles of the on-board switch SW2 Up).*

T2 *Display the TTL output of the sequence generator module on CH1-A.*

T3 *With the oscilloscope still triggered by the 'start-of-sequence' SYNC signal, observe CH1-A & CH1-B and make an accurate sketch*

. ***T4*** *Examine each code in turn from the encoder (by transferring CH1-B), and repeat the previous step.*

Experiment 8: Bit Clock Regeneration

***ACHIEVEMENTS**: introduction to bit clock regeneration. Evaluation using bitby-bit comparison with system bit clock.*

***ADVANCED MODULES**: BIT CLOCK REGEN.*

PREPARATION

Synchronization

Receivers in a digital environment can require synchronization at least three different levels:

• Carrier synchronization (in the case of bandpass signals)
• bit synchronization (at baseband)
• frame synchronization (at baseband)

This experiment is concerned with the second of these. It assumes either that the signal has been transmitted at baseband, or successfully recovered from a higher frequency carrier from which it has been demodulated

Stolen bit clock

For most TIMS experiments, when a bit clock is required by a receiver, it has been Convenient to use a 'stolen' clock. Bit clock regeneration from the received data Stream itself is not a trivial exercise, and is best avoided in the laboratory if at all possible. This eliminates unnecessary complications, and sources of signal corruption, and allows one to concentrate on other aspects of one`s investigations.

Regenerated bit clock

Bit clock regeneration cannot be avoided in a real-life situation. Techniques can be divided into two fundamental types: open loop, and closed loop.

This experiment is concerned with very basic open loop techniques.

Open Loop

If there is already a component at the bit clock frequency in the spectrum of the data stream, it can be extracted with a bandpass filter (BPF). Alternatively, there may be a

component at a higher harmonic; this, instead, could be extracted, and the fundamental obtained by division.

Figure 1 illustrates the basis of the most elementary example of an open loop system, where a component at bit clock frequency already exists in the data.

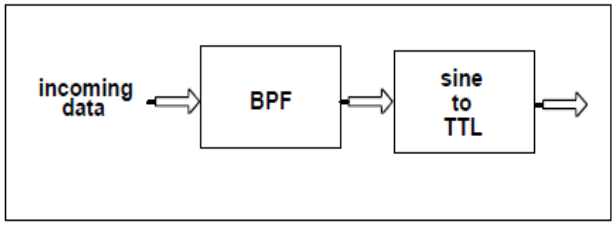


Figure 1: example of elementary open loop bit clock extraction

When there is no component at bit clock frequency or any of its harmonics it can probably be created by a non-linear element, as shown in Figure 2.

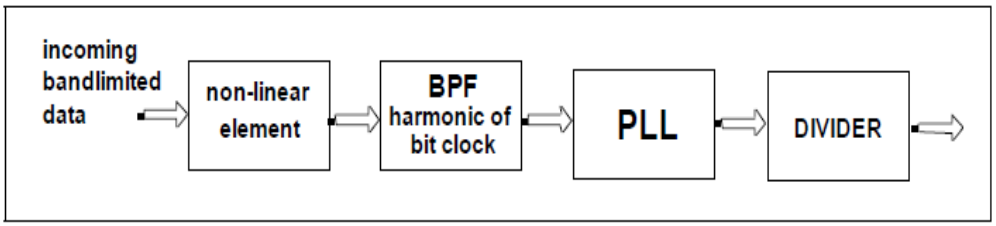
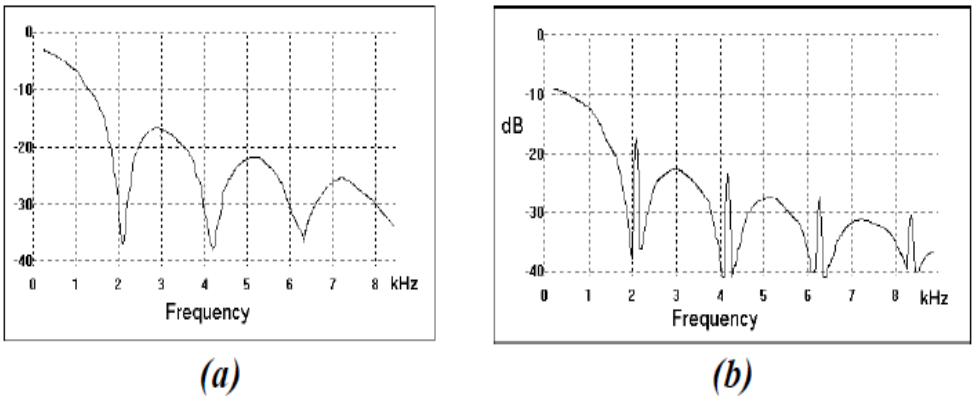


Figure 2: creation, and extraction, of a spectral component at bit clock frequency

TIMS non-linear elements in this context are:

- a MULTIPLIER (used as a squarer)
- the CLIPPER, in the UTILITIES module

For example, the spectrum of a bipolar pseudo random binary sequence from the SEQUENCE GENERATOR is of the form shown in Figure 3(a) below. Notice that there are nulls at all the harmonics of the bit clock frequency (2.0833 kHz). If this signal is first bandlimited, then squared, the spectrum, Figure 3(b), now contains lines at the bit clock frequency and its harmonics. A component at the bit rate can be extracted with, for example, a bandpass filter (BPF – see the BIT CLOCK REGEN module), or a phase locked loop (PLL) – or perhaps a combination of the two.



(a) *(b)*

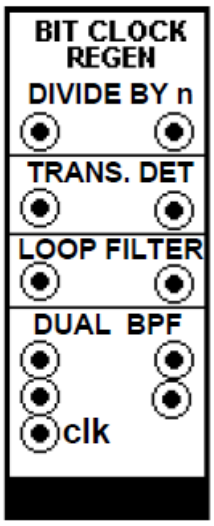
Figure 3: PRBS signal spectrum (a) before and (b) after bandlimiting and squaring

Closed loop

Closed loop circuits use feedback. They make comparisons with received data and expected data. They can involve the transmitter sending known sequences - training sequences - which are used by the receiver to verify synchronization 1.Closed loop systems are more accurate than open loop systems, but can be complex and costly. They are outside the scope of present TIMS modules.

BIT CLOCK REGEN module

This is the first time the BIT CLOCK REGEN module has been used. It is described in detail in the *Advanced Module User Manual*.
As can be seen, from the drawing of the front panel (opposite), the module contains four independent subsystems.
These have been described separately in the Chapter entitled *Digital utility sub-systems* (in this Volume) to which you should refer.
As its name implies, these sub-systems are useful in bit clock regeneration schemes, examples of which are given in the experiment to follow.



You may also devise your own schemes.

EXPERIMENT

Bit clock recovery - method #1

In the first example a bit clock will be recovered from the UNI-RZ coded output from the LINE-CODE ENCODER.
This waveform may be shown to contain energy at the bit clock frequency. So it can be extracted with a BPF according to the scheme of Figure 1.

T1 *acquire a BIT CLOCK REGEN module. Read about it in the Advanced Modules User Manual (see Appendix) before plugging it in locate.*

- *On-board switch SW1. Set the left hand toggle UP and the right hand toggle DOWN. This tunes BPF #1 to 2.083 kHz, and leaves BPF #2 to be tuned by an external TTL signal (at 50 times the desired passband.*

T2 *patch up the diagram of Figure4, which is a model of the open loop Regeneration scheme of Figure 1.*

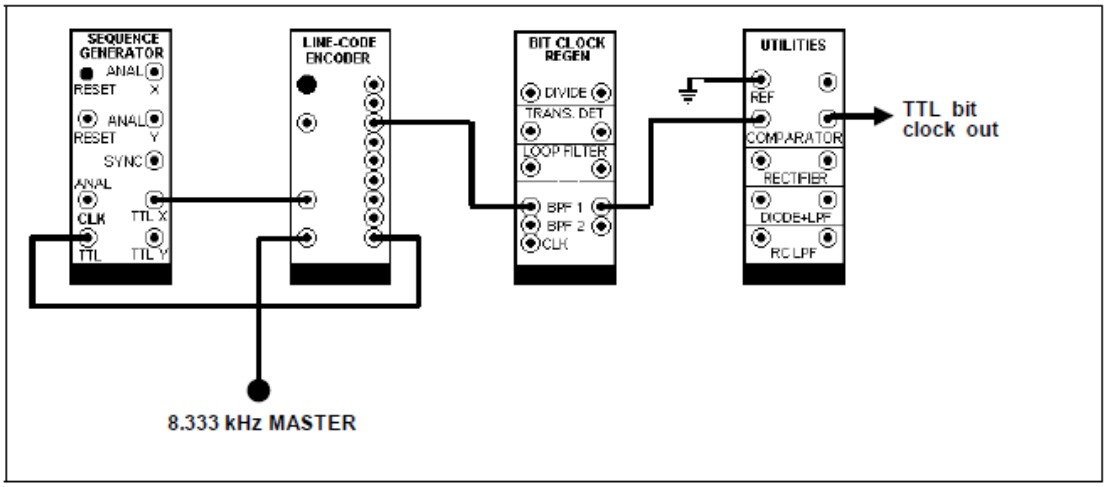


Figure 4: model of Figure 1

T3 *Display the output of the line-code encoder (Uni-RZ) on CH1-A, confirm with the PICO SPECTRUM ANALYZER that this format of data has a spectral line at the bit clock frequency.*

T4 *using the 2.083 kHz as a reference on CH1-A, look at the output of BPF #1 with CH2-A. This will be a sinewave, also on a mean frequency of 2.083 kHz.*

T5 *observe the output of the COMPARATOR on CH1-B. This is a TTL signal, of fixed amplitude, and mean frequency 2.083 kHz.*

Bit clock recovery - method #2

The previous bit clock recovery method extracted a component at bit clock frequency which was already present in the data stream.

This second method is truly a regenerative method, since the data stream will not have such a component present.

It will model the block diagram of Figure 2, using a MULTIPLIER as a squarer.

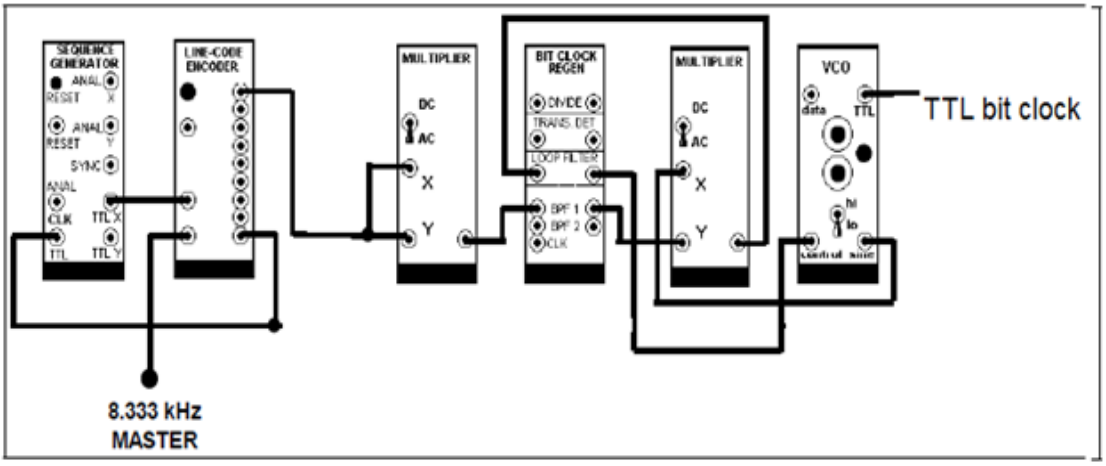


Figure 5: the TIMS model

T6 *Display the output of the line-code encoder (NRZ-L) on CH1-A, confirm with the PICO SPECTRUM ANALYZER that this format of data does not have a spectral line at the bit clock frequency.*

T7 *the BPF in the BIT CLOCK REGEN module must be tuned to 2.083 kHz by setting the on-board switch SW1 to INT CLK.*

T8 *Adjust the controls on the VCO module until you get a stable display of the TTL bit clock on CH1-B.*

This is a TTL signal, of fixed amplitude, and mean frequency 4.167 kHz which is twice the original bit clock frequency. So the next stage must be a divider-by-2 to get the correct frequency of the TTL bit clock.

APPENDIX

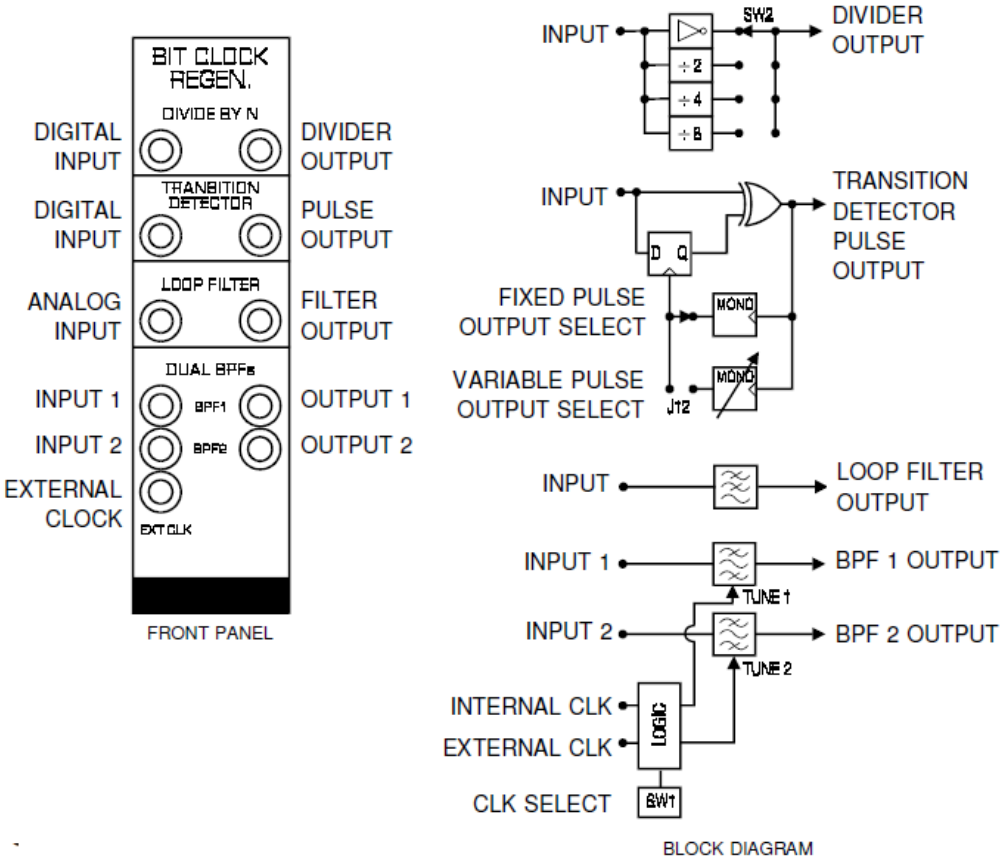
BIT CLOCK REGENERATION

Four independent functional blocks are provided, which may be used independently or in combination with other TIMS modules, to recover the bit clock of any TIMS generated Line-Code.

Schemes which may be constructed and demonstrated using the building block functions of the BIT CLOCK REGENERATION module along with other TIMS modules include:

- Bandpass Filter jitter reduction techniques,
- Bandpass Filter bit-sync derivation and
- Phase Lock Loop bit-sync derivation,

Using filter/square-law, transition detector based and various other clock recovery structures.



DIVIDE BY N

The DIVIDE BY N is a general purpose digital divider. It accepts a standard TTL level signal at the input and outputs a standard TTL level signal. The PCB mounted DIP switch, SW2, is used to select the division factor, as illustrated in the table below.

SW2-1 (A)	SW2-2 (B)	DIV. MODE
OFF	OFF	divide by 8
OFF	ON	divide by 4
ON	OFF	divide by 2
ON	ON	divide by -1 (invert)

A typical application for the DIVIDE BY N may be as part of a Phase Lock Loop system.

TRANSITION DETECTOR

The TRANSITION DETECTOR will produce a TTL level output pulse for every transition in logic level of the input digital sequence. The input sequence must be TTL level.

Operation of the TRANSITION DETECTOR is such that the input sequence is delayed using a clocked flip-flop. The exclusive-OR circuit then performs the equivalent of a multiplication operation. The width of the output pulse is dependent upon the width of the monostable's pulse.

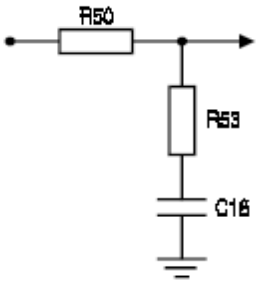
The PCB mounted jumper, J12, allows the user to select either a fixed pulse width, FIX, or a manually adjustable pulse width, VAR. The fixed pulse width monostable optimizes the TRANSITION DETECTOR's operation for use with the LINE-CODE ENCODER module's standard 2.083 kHz bit clock.

An adjustable pulse width monostable is also available to allow the user to determine the effect of different pulse widths on the operation of the TRANSITION DETECTOR under various conditions. The pulse width is varied using the PCB mounted trimmer labeled VARY PULSE WIDTH, RV1. Adjusting the trimmer varies the output pulse width from approximately 10µs to 500µs.

In a bit synchronization system, the output of the TRANSITION DETECTOR would normally pass to a bandpass filter or phase lock loop.

LOOP FILTER

The LOOP FILTER is intended for use in Phase Lock Loop, PLL, applications such as Demonstrating PLL bit-sync derivation. It is a conventional, passive, Type 1, second-order* loop structure, as illustrated below. The factory selected component values are also given



Second order loop

Label	Value
R50	9k1R
R53	1k9R
C18	100nF

Please note that the loop filter's input and output have active buffering using op-amp circuits: (this is not illustrated in the above figure.)

Also note that PLLs are classified according to Type, based on the number of poles of the loop transfer function at the origin. The order of the loop refers to the highest degree of the Polynomial of the characteristic equation, 1 + G(s)H(s). Ref: Digital Communications with Fiber Optics and Satellite Applications, Harold B. Killen, Prentice-Hall Inc.

DUAL BPFs

Two independent, tunable, high-Q bandpass filters are provided, to demonstrate both
- Bandpass filter jitter reduction and bandpass filter bit-sync derivation.
- Each filter accepts and outputs standard TIMS level signals.
- Both filters have the same fixed Q of 22.

The center frequency of each filter is controlled by a digital clock signal. The frequency of the digital clock signal is 50 times the center frequency of the BPF. The source of the digital clock signal may be either the internal (on-board) crystal oscillator or an external oscillator.

The PCB mounted DIP switch, SW1, is used to select each filter's clock source.
The internal crystal derived clock, INT.CLK, is optimized for use with the LINE-CODE ENCODER module's standard 2.083 kHz bit clock.

The external clock, EXT.CLK, may be used to tune the center frequency of either or both of the filters between 1 kHz and 5 kHz. The external TTL level clock source is applied via the front panel EXT CLK input.

The table below lists all possible combinations of clock source for both filters.

SW1-1	SW1-2	BPF 1 SOURCE	BPF 2 SOURCE
OFF	OFF	External	External
OFF	ON	External	Internal
ON	OFF	Internal	External
ON	ON	Internal	Internal

Please note that when BPF 1 and BPF 2 both have External Source selected, both filters receive the same clock signal via the front panel EXT CLK input.

BASIC SPECIFICATIONS

DIVIDE BY N
Input & Output TTL level, digital signals
Clock input <1MHz
Divisors -1, 2, 4 and 8, switch selectable

TRANSITION DETECTOR
Input & Output TTL level, digital signals
Output Pulse Width
with **FIX** selected at **J12**: approx. 250µs
with **VAR** selected at **J12**: adjustable from approx. 10µs to approx. 500µs

LOOP FILTER
Input & Output standard TIMS level, analog signals
Type conventional, passive, Type 1, second-order* loop structure
* refer to the previous page for definitions
Characteristics required to be determined by the student. See previous page for details.
Buffering active

DUAL BANDPASS FILTERS
Input & Output standard TIMS level, analog signals
Number two identical bandpass filters
Type fourth order Chebyshev with 3dB passband ripple
Q approx. 22, fixed
Ratio of Tuning Clock to Filter's Centre Frequency 50
Internal Clock Frequency 104kHz, crystal derived, giving 2.083kHz filter centre frequency
External Clock Frequency Range 50kHz to 250kHz, TTL level

EXPERIMENT 9: Amplitude Shift Keying (ASK)

❖*Generation*

- ***Modeling with a DUAL ANALOG SWITCH***
- ***Modeling with a MULTIPLIER***

❖*Demodulation*

- ***Envelope demodulation***
- ***Synchronous demodulation #1***
- ***Post-demodulation processing***

Lab 5

Experiment 9: ASK - AMPLITUDE SHIFT KEYING

***ACHIEVEMENTS**: generation and demodulation of an amplitude shift keyed (ASK) signal.*

***PREREQUISITES**: it would be advantageous to have completed some of the experiments in Volume A1 involving linear modulation and demodulation.*

***EXTRA MODULES**: DECISION MAKER*

PREPARATION

Generation

Amplitude shift keying - ASK - in the context of digital communications is a modulation process which imparts to a sinusoid two or more discrete amplitude levels (also called on-off keying – OOK). These are related to the number of levels adopted by the digital message.

For a binary message sequence there are two levels, one of which is typically zero. Thus the modulated waveform consists of bursts of a sinusoid.

Figure 1 illustrates a binary ASK signal (lower), together with the binary sequence which initiated it (upper). Neither signal has been bandlimited.

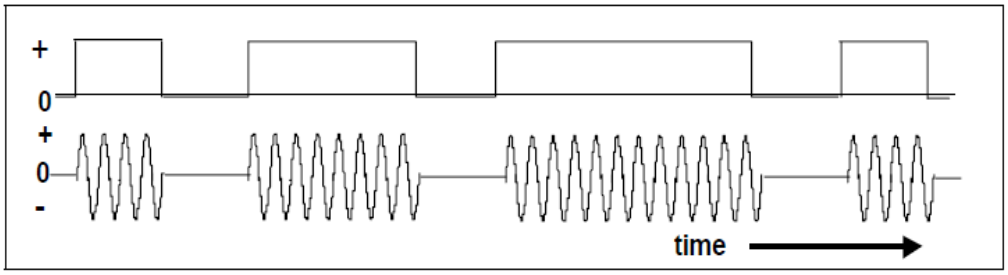


Figure 1: an ASK signal (below) and the message (above)

There are sharp discontinuities shown at the transition points. These result in the signal having an unnecessarily wide bandwidth. Band limiting is generally introduced before transmission, in which case these discontinuities would be 'rounded off'. The band limiting may be applied to the digital message, or the modulated signal itself.
The data rate is often made a sub-multiple of the carrier frequency. This has been done in the waveform of Figure 1.

One of the disadvantages of ASK, compared with FSK and PSK, for example, is that it has not got a constant envelope. This makes its processing (eg, power amplification) more difficult, since linearity becomes an important factor. However, it does make for ease of demodulation with an envelope detector.

A block diagram of a basic ASK generator is shown in Figure 2. This shows band limiting following modulation.

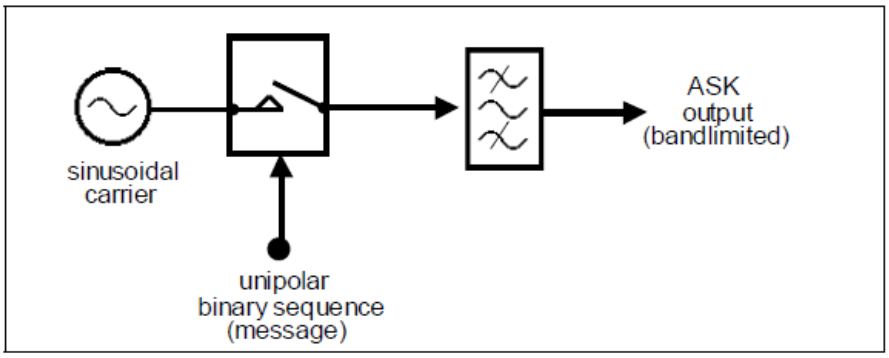


Figure 2: the principle of ASK generation

The switch is opened and closed by the unipolar binary sequence.

Bandwidth modification

As already indicated, the sharp discontinuities in the ASK waveform of Figure 1 imply a wide bandwidth. A significant reduction can be accepted before errors at the receiver increase unacceptably. This can be brought about by band limiting (pulse shaping) the message *before* modulation, or band limiting the ASK signal itself *after* generation. Both these options are illustrated in Figure 3, which shows one of the generators you will be modelling in this experiment.

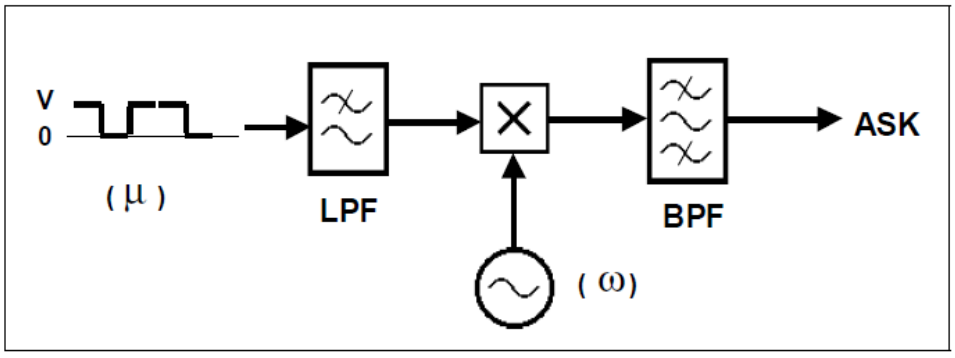


Figure 3: ASK bandlimiting, with a LPF *or* a BPF.

Figure 4 shows the signals present in a model of Figure 3, where the message has been bandlimited. The shape, after band limiting, depends naturally enough upon the amplitude and phase characteristics of the band limiting filter.

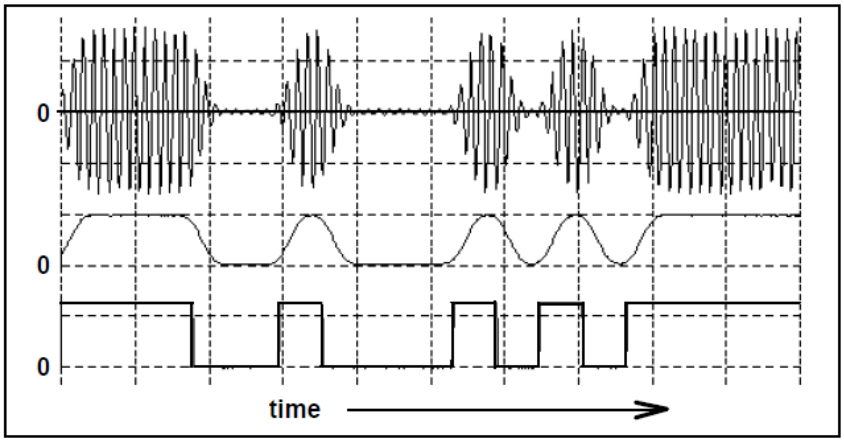


Figure 4: original TTL message (lower), bandlimited message (centre), and ASK (above)

You can approximate these waveforms with a SEQUENCE GENERATOR clocked at about 2 kHz, filter #3 of the BASEBAND CHANNEL FILTERS, and a 10 kHz carrier from a VCO.

Demodulation methods

It is apparent from Figures 1 and 4 that the ASK signal has a well-defined envelope. Thus it is amenable to demodulation by an envelope detector.

A synchronous demodulator would also be appropriate.
Note that:

• Envelope detection circuitry is simple.
• Synchronous demodulation requires a phase-locked local carrier and therefore carrier acquisition circuitry.

With band limiting of the transmitted ASK neither of these demodulation methods would recover the original binary sequence; instead, their outputs would be a bandlimited version. Thus further processing - by some sort of decision-making circuitry for example - would be necessary.

Thus demodulation is a two-stage process:
1. Recovery of the bandlimited bit stream
2. Regeneration of the binary bit stream

Figure 5 illustrates.

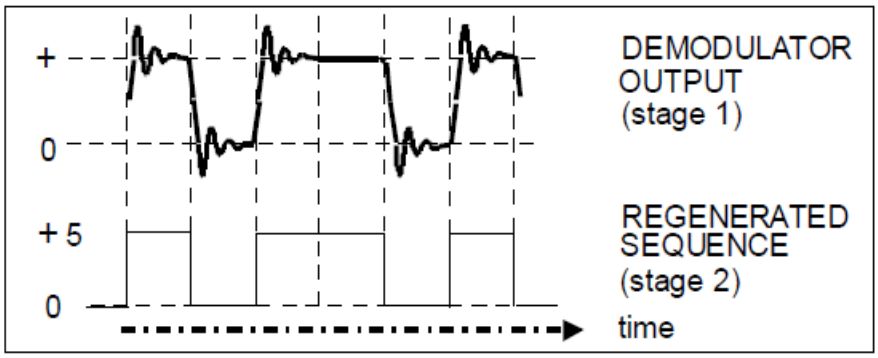


Figure 5: the two stages of the demodulation process

EXPERIMENT

T1.0 Generation

There are many methods of modeling an ASK generator with TIMS. For any of them the binary message sequence is best obtained from a SEQUENCE GENERATOR, clocked at an appropriate speed. Depending upon the generator configuration, either the data bit stream can be bandlimited, or the ASK itself can be bandpass filtered.

Suggestions for modeling the ASK generators are:

T1.1 modeling with a DUAL ANALOG SWITCH

It is possible to model the rather basic generator shown in Figure 2.

The switch can be modeled by one half of a DUAL ANALOG SWITCH module. Being an *analog* switch, the carrier frequency would need to be in the audio range. For example, 15 kHz from a VCO. The TTL output from the SEQUENCE GENERATOR is connected directly to the CONTROL input of the DUAL ANALOG SWITCH. For a synchronous carrier and message use the 8.333 kHz TTL sample clock (filtered by a TUNABLE LPF) and the 2.083 kHz sinusoidal message from the MASTER SIGNALS module.

If you need the TUNABLE LPF for band limiting of the ASK, use the sinusoidal output from an AUDIO OSCILLATOR as the carrier. For a synchronized message as above, tune the oscillator close to 8.333 kHz, and lock it there with the sample clock connected to its SYNCH input.

This arrangement is shown modeled in Figure 6.

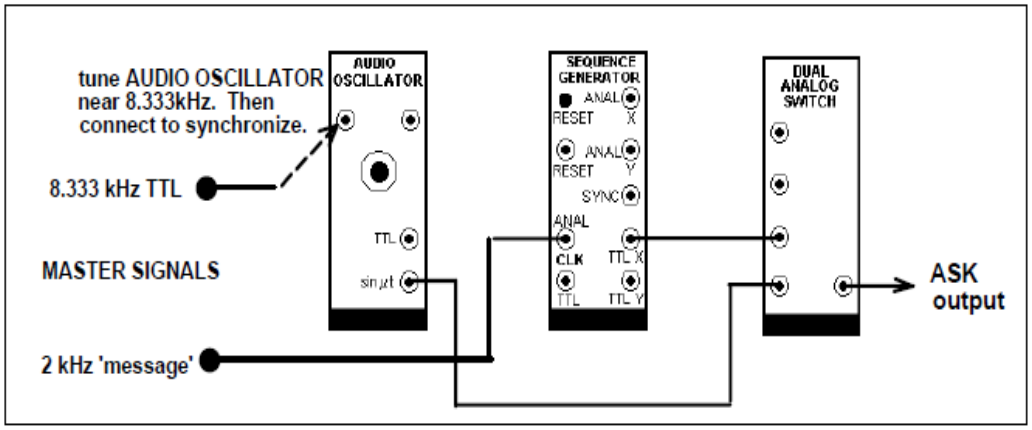


Figure 6: modelling ASK with the arrangement of Figure 2

T1 *Set up the model of Figure* 6. *Select a short sequence (both toggles of the on-board switch SW2 Up).*

T2 *use the sinusoidal output from an AUDIO OSCILLATOR as the carrier. For a synchronized message as above, tune the oscillator close to 8.333 kHz, and lock it there with the sample clock connected to its SYNCH input.*

T3 *With the oscilloscope still triggered by the 'start-of-sequence' SYNC signal, observe both the TTL output of the sequence generator and output of the dual switch on separate oscilloscope channels, and make an accurate sketch*

- Band limiting can be implemented with a filter at the output of the ANALOG SWITCH.

T1.2 Modeling with a MULTIPLIER

A MULTIPLIER module can be used as the switch. The carrier can come from any suitable sinusoidal source. It could be at any available TIMS frequency.

The other input to the MULTIPLIER needs to be the message sequence.

Neither the TTL nor the analog sequence is at an appropriate voltage level. Each requires amplitude scaling. This can be implemented in an ADDER, which will invert the sequence polarity. DC from the VARIABLE DC module can be used to re-set the DC level. The required signal will be at a level of either O V or +2 V, the latter being optimum for the (analog) MULTIPLIER.

This arrangement is shown modeled in Figure 7.

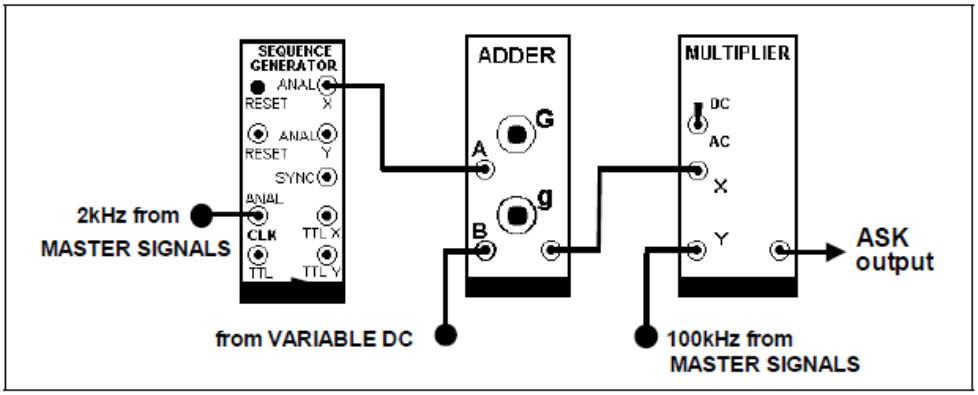


Figure 7: modelling ASK with the arrangement of Figure 3.

T4 *Set up the model of Figure 7. Select a short sequence (both toggles of the on-board switch SW2 Up).*

T5 *Set the gain controls 'G & g' of the ADDER to mid-range.*

T6 *Adjust the variable dc until the required signal be at a level of either O V or +2 V (output of the adder), check by CH1-A.*

T7 *With the oscilloscope still triggered by the 'start-of-sequence' SYNC signal, Display the output of the multiplier on CH1-B , and make an accurate sketch*

- Band limiting can be implemented with a filter at the MULTIPLIER output (a 100 kHz CHANNEL FILTERS module), or the bit sequence itself can be bandlimited (BASEBAND CHANNEL FILTERS module).

T2.0 Demodulation

Both asynchronous and synchronous demodulation methods are used for the demodulation of ASK signals.

T2.1 Envelope demodulation

Having a very definite envelope, an envelope detector can be used as the first step in recovering the original sequence. Further processing can be employed to regenerate the true binary waveform.

Figure 8 is a model for envelope recovery from a baseband ASK signal.

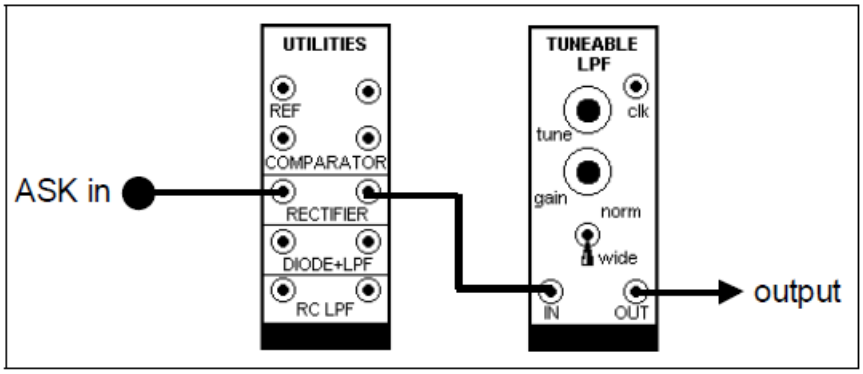


Figure 8: envelope demodulation of baseband ASK

T8 *With the oscilloscope still triggered by the 'start-of-sequence' SYNC signal, make an accurate sketch for the output.*

If you choose to evaluate the model of Figure 8, remember there is a relationship between bit rate and the low pass filter bandwidth. Select your frequencies wisely.

T2.2 Synchronous demodulation

A synchronous demodulator can be used for demodulation, as shown in Figure 9, In the laboratory you can use a stolen carrier, as shown.

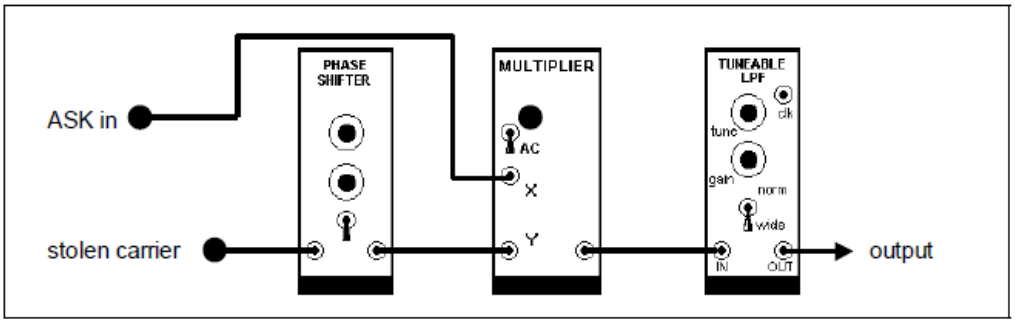


Figure 9 : synchronous demodulation of ASK

T9 *Set up the model of Figure* 9. *Select a short sequence (both toggles of the on-board switch SW2 Up).*

T10 *With the oscilloscope still triggered by the 'start-of-sequence' SYNC signal, observe both the TTL output of the sequence generator and output of the demodulator on separate oscilloscope channels, and make an accurate sketch*

T2.3 Post-demodulation processing

The output from both of the above demodulators will not be a copy of the binary sequence TTL waveform. Band limiting will have shaped it, as (for example) illustrated in Figure 4.

Some sort of decision device is then required to regenerate the original binary sequence. The DECISION MAKER module could be employed, with associated processing, if required. This is illustrated in block diagram form in Figure 10.

This model will regenerate a bi-polar sequence from the recovered envelope.

Figure 11 shows the model of the block diagram of Figure 10

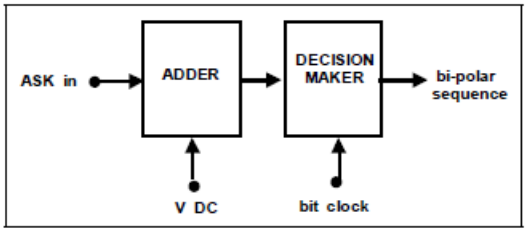


***Figure* 10**

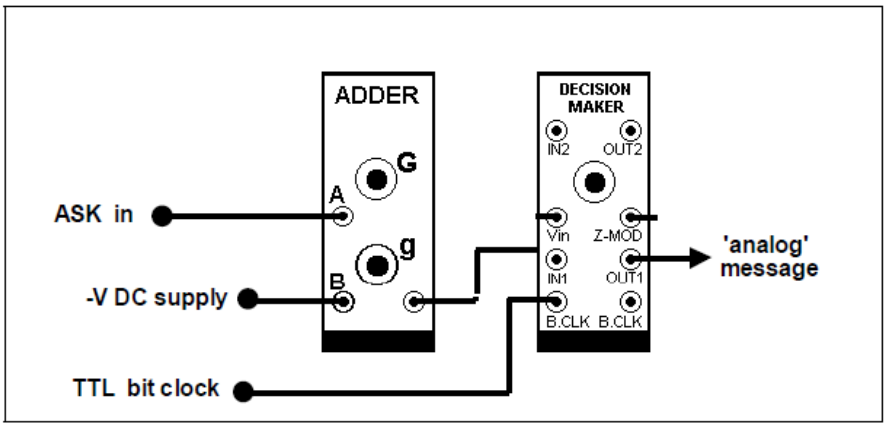


Figure 11: regeneration to a bi-polar sequence

Remember to

- Convert the uni-polar, bandlimited output of the envelope detector to bi-polar (using the ADDER), to suit the DECISION MAKER.
- Set the on-board switch SW1, of the DECISION MAKER, to NRZ-L. This configures it to accept bi-polar inputs.
- Adjust the decision point of the DECISION MAKER
- In the first instance, use a stolen carrier and bit clock

The output will be the regenerated message waveform. Coming from a YELLOW analog output socket, it is bi-polar ±2 V (not TTL).

EXPERIMENT 10: Frequency Shift Keying (FSK)

- ❖ ***Generation***

- ❖ ***Demodulation***
  - ***Asynchronous Receiver***
  - ***PLL - phase locked loop***

Lab 6

Experiment 10: FSK - Frequency Shift Keying

***ACHIEVEMENTS**: Generation and demodulation of a binary FSK signal*

***PREREQUISITES:** It would be advantageous to have completed some of the Part 1 experiments involving linear modulation and demodulation.*

***EXTRA MODULES**: nothing*

PREPARATION

Generation

Digital modulation techniques include: amplitude shift keying (ASK), frequency shift keying (FSK), and phase shift keying (PSK). In this experiment we will concentrate on FSK. The other techniques can be related to it. The block diagram of Figure 1 illustrates the operation of an FSK generator.

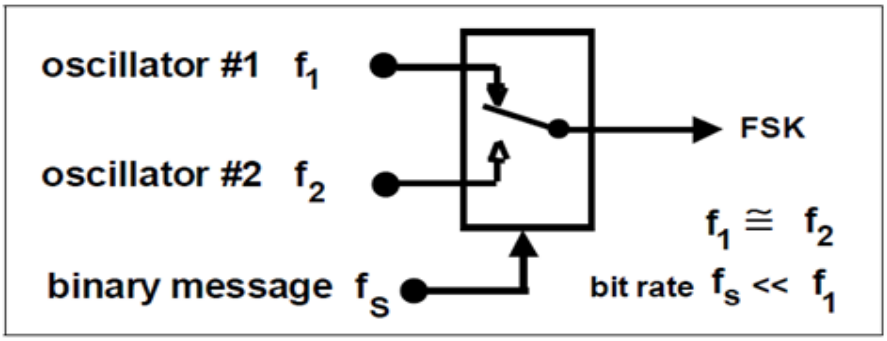


Figure 1: an FSK transmitter

In principle the three frequencies f_1, f_2, and f_s are independent. In practice this is often not so - there are certain advantages in having them related in some way (eg, as submultiples). Oscillator #1 and #2 can be taken from the same source (say a VCO), whose frequency is changed by the message, leading to a *continuous phase* output (CPFSK). This is illustrated in Figure 2, which shows a VCO as the source of f_1 and f_2, and the corresponding CPFSK output waveform.

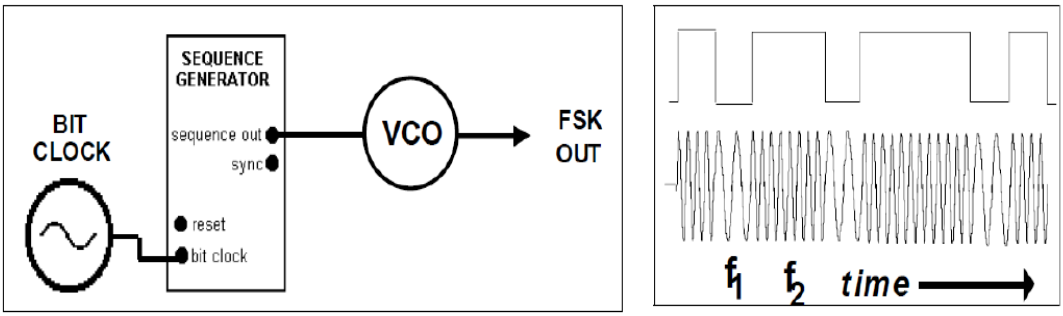


Figure 2: CPFSK generation and output waveform

Demodulation

There are different methods of demodulating FSK. A natural classification is into synchronous (coherent) or asynchronous (non-coherent).

Representative demodulators of these two types are the following:

Asynchronous

A close look at the waveform of Figure 1 reveals that it is the sum of two amplitude shift keyed (ASK) signals. These signals were examined in the experiment entitled *ASK- amplitude shift keying*.

The receiver of Figure 3 takes advantage of this. The FSK signal has been separated into two parts by bandpass filters (BPF) tuned to the MARK and SPACE frequencies.

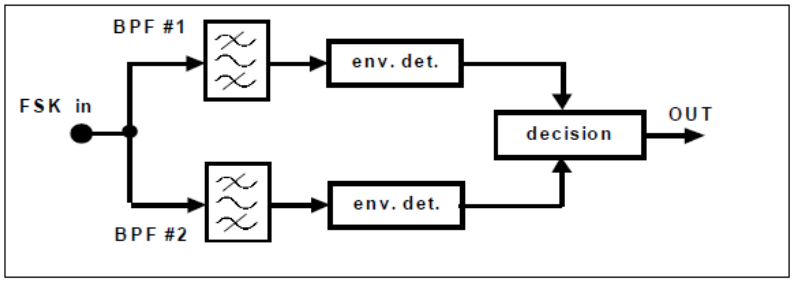


Figure 3: demodulation by conversion-to-ASK

The output from each BPF looks like an amplitude shift keyed (ASK) signal.

These can be demodulated asynchronously, using the envelope. The envelope detector is examined in the experiment entitled *Envelope recovery* within *Volume A 1 - Fundamental Analog Experiments.*

The decision circuit, to which the outputs of the envelope detectors are presented, selects the output which is the most likely one of the two inputs. It also re-shapes the waveform from a bandlimited to a rectangular form.

This is, in effect, a two channel receiver. The bandwidth of each is dependent on the message bit rate. There will be a minimum frequency separation required of the two tones.

HINT

You are advised to read ahead, *before* attempting the experiment, to consider the modeling of this demodulator. Unlike most TIMS models, you are *not* free to choose parameters -particularly frequencies. If they are to be tuned to *different* frequencies, then one of these frequencies must be 2.083 kHz (defined as the MARK frequency). This is a restriction imposed by the BIT CLOCK REGEN module, of which the BPF are sub-systems. As a result of this, most other frequencies involved are predetermined. Make sure you appreciate why this is so, then decide upon:

- Bit clock rate
- SPACE frequency
- Envelope detector LPF characteristics

Synchronous

In the block diagram of Figure 4 two local carriers, on each of the two frequencies of the binary FSK signal, are used in two synchronous demodulators. A decision circuit examines the two outputs, and decides which is the most likely.

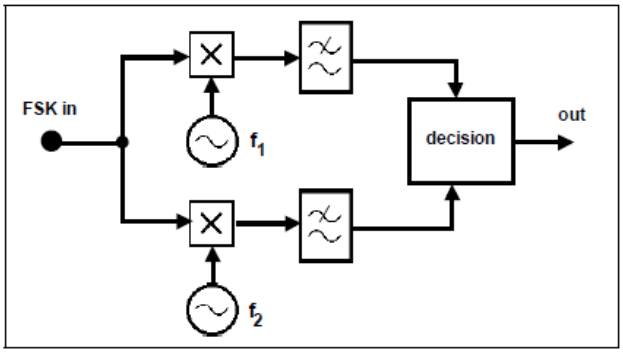


Figure 4: synchronous demodulation

This is, in effect, a two channel receiver. The bandwidth of each is dependent on the message bit rate. There will be a minimum frequency separation required of the two tones. This demodulator is more complex than most asynchronous demodulators.

Phase locked loop

A phase locked loop is a well-known method of demodulating an FM signal. It is thus capable of demodulating an FSK signal. It is examined in the experiment entitled *FM demodulation with the PLL* within *Volume A2 -Further & Advanced Analog Experiments.* It is shown, in block diagram form, in Figure 5 below.

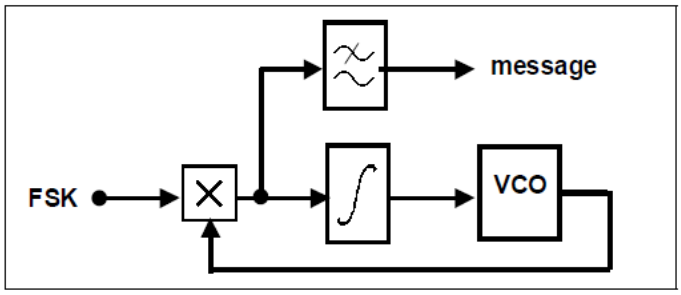


Figure 5: phase locked loop demodulator

The control signal, which forces the lock, is a bandlimited copy of the message sequence. Depending upon the bandwidth of the loop integrator, a separate LPF will probably be required (as shown) to recover the message.

EXPERIMENT

Generation

A VCO module is ideally suited for the generation of a continuous phase FSK signal, as shown in Figure 6.

In FSK mode the VCO is keyed by the message TTL sequence. Internal circuitry results in a TTL HI switching the VCO to frequency f_1 while a TTL LO switches it to frequency f_2. These two frequencies may be in the audio range (front panel toggle switch LO), or in the 100 kHz range (front panel toggle switch HI). The frequencies f_1 and f_2 are set by the on-board variable resistors RV8 and RV7 respectively, while a continuous TTL HI or a TTL LO is connected to the DATA input socket.

In FSK mode neither of the front panel rotary controls of the VCO is in operation

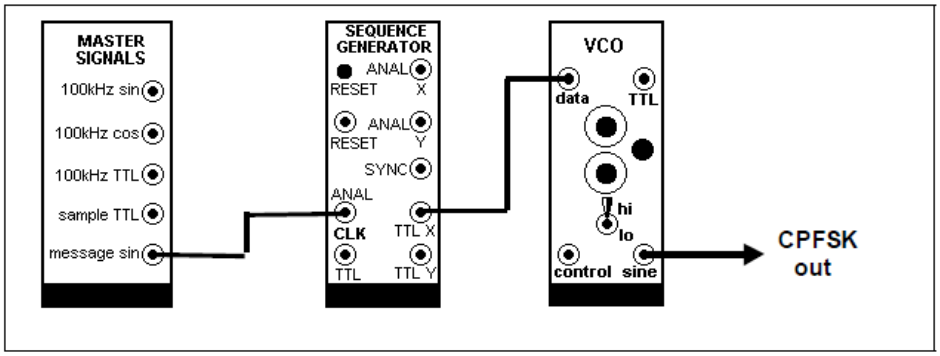


Figure 6: CPFSK

T1 *Set up the model in figure 6. Make sure that the VCO is set to FSK mode. Set the frequency range to the LO.*

Set the MARK and SPACE frequencies of the FSK generator as follows:

T2 Input a +5 V signal from the system unite Variable DC module into the DATA *input of the VCO.*

T3 *Adjust onboard control FSK2 While viewing the sinusoidal output frequency of VCO (using the FREQUENCY COUNTER). Set this MARK frequency to around 4.2 KHz*

T4 *Input 0 V signal from the GROUND terminal on Variable DC panel and adjust FSK1 to give an output frequency of around 9.6 KHz.*

T5 *Finally, connect the MESSAGE data stream to the DATA input of the VCO.*

We now have an FSK generator with a center frequency of round (9.6+4.2)/2 = 6.9KHz, and a difference frequency of (9.6-4.2)/2=3.7KHz.

Leave this FSK model set up for the following demodulation part of the experiment.

Demodulation

Asynchronous Receiver

To keep the experiment as simple as possible, we will simply detect the MARK only using an envelope detector method. Since the MARK is the lower of the two transmitted tones, we can use a TUNABLE LPF to discard the SPACE signal frequency.

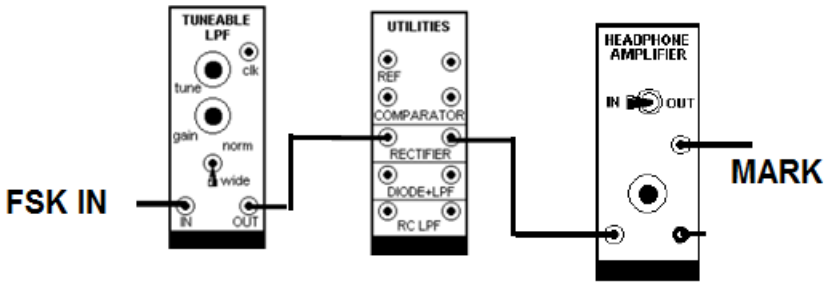


Figure 7: elementary FSK detector

T6 *patch* together the model of the detector in figure 7. Ensure that the message bit rate of the FSK generator is 2.083 KHz.

T7 *While viewing the output of the TUNABLE LPF (CH1-A), set the WIDE mode and vary the controllers until the SPACE component frequency is attenuated from the output. This should now look like an ASK signal, with a frequency component at times when the original data is high (MARK).*

Now we need to envelope recover this signal to create a typical digital data stream.

T8 *Display the output of the LPF in the HEADPHONE AMPLIFIER (CH1-B),*

T9 *compare the recovered data stream with the original message data.*

PLL - phase locked loop

A phase locked loop is shown in block diagram form in Figure 5, and modeled in Figure 8

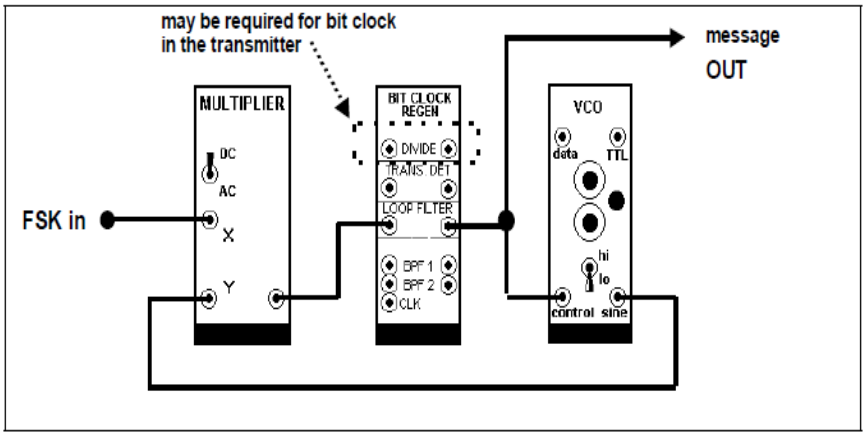


Figure 8: PLL demodulator - the model of Figure 5

T10 *Set up the model of the demodulator of Figure* 8. Set the bit rate of the generator to 300Hz (using the audio generator as the bit clock).

T11 Display the message output on CH1-A, and compare it with the TTL output of sequence generator (CH1-B). (You **must adjust the controllers on the VCO to get a stable display of the message output)**

- *If you are fussy about the appearance of the demodulated output it can be further filtered; say with the LPF in the HEADPHONE AMPLIFIER.*
- *Could you use either the COMPARATOR or the DECISION MAKER in the UTILITIES modules, to regenerate the message as a clean TTL sequence?*

EXPERIMENT 11: Binary Phase Shift Keying (BPSK)

- ***The BPSK generator***
- ***The BPSK Demodulation***

Lab 7

Experiment 11: BPSK - Binary Phase Shift Keying

ACHIEVEMENTS: *Generation of binary phase shift keyed (BPSK) signal; band limiting; synchronous demodulation - phase ambiguities.*

PREREQUISITES: *It would be advantageous to have completed some of the experiments in Volume A1, involving linear modulation and demodulation. Familiarity with the DECISION MAKER, LINE- CODE ENCODER and LINE-CODE DECODER modules is assumed.*

EXTRA MODULES: *DECISION MAKER, LINE-CODE ENCODER and LINE-CODE DECODER. A total of two PHASE SHIFTER modules is required.*

PREPARATION

Consider a sinusoidal carrier. If it is modulated by a bi-polar bit stream according to the scheme illustrated in Figure 1 below, its polarity will be reversed every time the bit stream changes polarity. This, for a sinewave, is equivalent to a phase reversal (shift). The multiplier output is a BPSK***(also sometimes called PRK - phase reversal keying)** signal.

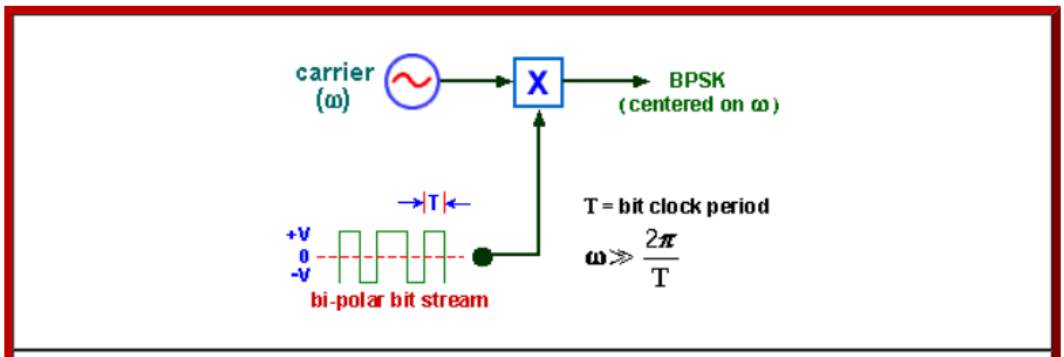


Figure I: generation of BPSK

The information about the bit stream is contained in the changes of phase of the transmitted signal.

A synchronous demodulator would be sensitive to these phase reversals.

The appearance of a BPSK signal in the time domain is shown in Figure 2 (lower trace). The upper trace is the binary message sequence.

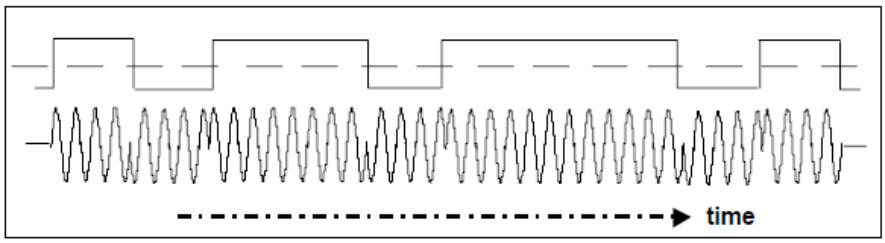


Figure 2: a BPSK signal in the time domain.

There is something special about the waveform of Figure 2. The wave shape is '**symmetrical**' at each phase transition. This is because the bit rate is a sub-multiple of the carrier frequency w / **(**2p**)**. In addition, the message transitions have been timed to occur at a zero-crossing of the carrier.

Whilst this is referred to as 'special', it is not uncommon in practice. It offers the advantage of simplifying the bit clock recovery from a received signal. Once the carrier has been acquired then the bit clock can be derived by division.

Bandlimiting

The basic BPSK generated by the simplified arrangement illustrated in Figure 1 will have a bandwidth in excess of that considered acceptable for efficient communications.

If you can calculate the spectrum of the binary sequence then you know the bandwidth of the BPSK itself. The BPSK signal is a linearly ***modulated*** *DSB*, and so it has a bandwidth twice that of the baseband data signal from which it is derived***(This assumes w > 2B)**.

In practice there would need to be some form of bandwidth control.

Bandlimiting can be performed either at baseband or at carrier frequency. It will be performed at baseband in this experiment.

BPSK Demodulation

Demodulation of a BPSK signal can be considered a two-stage process.

1. Translation back to baseband, with recovery of the band limited message waveform
2. Regeneration from the bandlimited waveform back to the binary message bit stream.

Translation back to baseband requires a local, synchronized carrier.

Stage 1

Translation back to baseband is achieved with a synchronous demodulator, as shown in Figure 3 below.

This requires a local synchronous carrier. In this experiment a stolen carrier will be used.

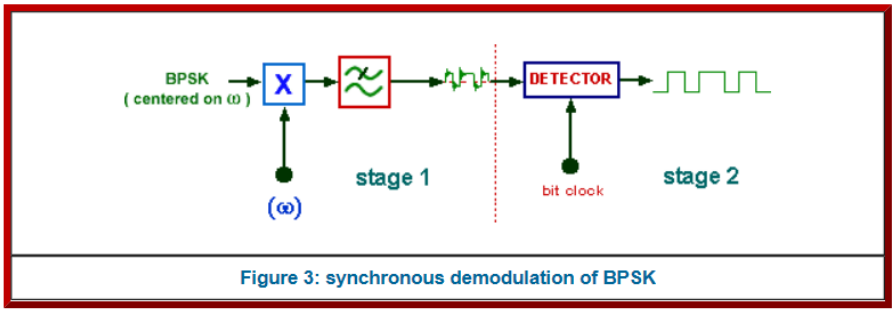


Figure 3: synchronous demodulation of BPSK

Stage 2

The translation process does not reproduce the original binary sequence, but a band limited version of it.

The original binary sequence can be regenerated with a detector. This requires information regarding the bit clock rate. If the bit rate is a sub-multiple of the carrier frequency then bit clock regeneration is simplified.

In TIMS the DECISION MAKER module can be used for the regenerator, and in this experiment the bit clock ***will*** be a sub-multiple of the carrier.

Phase ambiguity

You will see in the experiment that the sign of the phase of the demodulator carrier is important.

Phase ambiguity is a problem in the demodulation of a BPSK signal.

There are techniques available to overcome this. One such sends a training sequence, of known format, to enable the receiver to select the desired phase, following which the training sequence is replaced by the normal data (until synchronism is lost !).

An alternative technique is to use differential encoding. This will be demonstrated in this experiment by selecting a different code from the LINE-CODE ENCODER.

EXPERIMENT

The BPSK generator

The BPSK generator of Figure 1 is shown in expanded form in Figure 4, and modeled in Figure 5

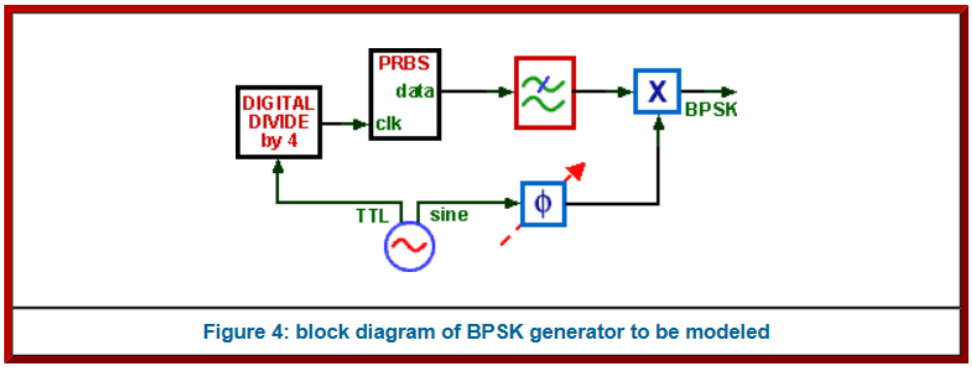


Figure 4: block diagram of BPSK generator to be modeled

Note that the carrier will be four times the bit clock rate.

The low pass filter is included as a band limiter if required. Alternatively a band pass filter could have been inserted at the output of the generator. Being a linear system, the effect would be the same.

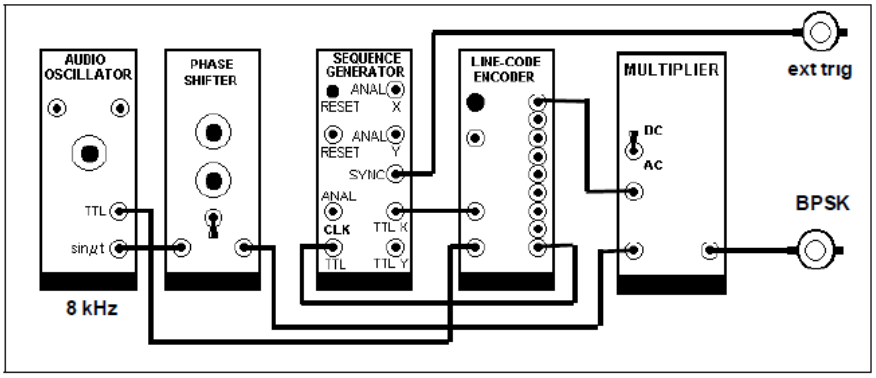


Figure 5: model of the BPSK generator

Tl *Patch up the modulator of Figure 5*; acquaint *yourself with a BPSK signal. Examine the transitions as the phase between bit clock and carrier is altered.*

Notes:

- The AUDIO OSCILLATOR supplies a TTL signal for the bit clock digital DIVIDE- BY -FOUR sub-system in the LINE-CODE ENCODER, and a sinusoidal signal for the carrier.

- The PHASE SHIFTER (set to the LO range with the on-board switch SW1) allows relative phase shifts. Watch the phase transitions in the BPSK output signal as this phase is altered. This PHASE SHIFTER can be considered optional.
- The digital DIVIDE-BY-FOUR sub-system within the LINE-CODE ENCODER is used for deriving the bit clock as a sub-multiple of the BPSK carrier. Because the DECISION MAKER, used in the receiver, needs to operate in the range about 2 to 4 kHz, the BPSK carrier will be in the range about 8 to 16 kHz.
- The NRZ-L code is selected from LINE-CODE ENCODER

BPSK demodulator

Figure 3 shows a synchronous demodulator for a BPSK signal in block diagram form. This has been modeled in Figure 6 below. In the first part of the experiment the carrier and bit clocks will be stolen.

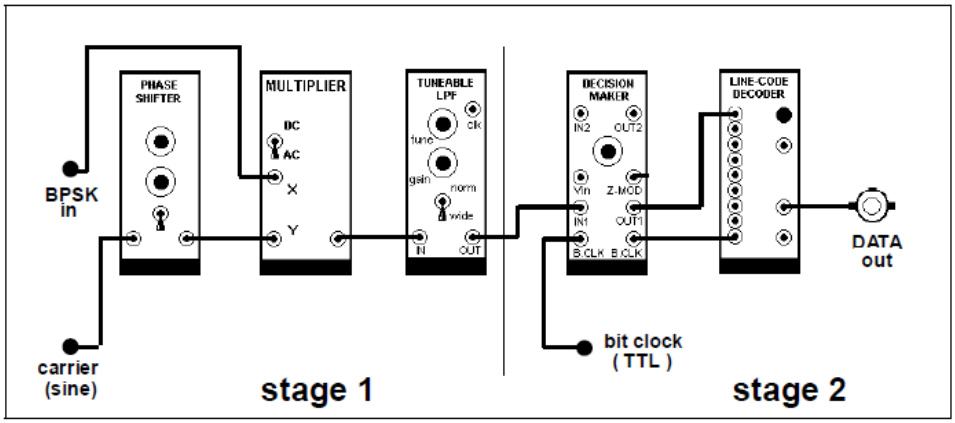


Figure 6: the BPSK demodulator

The phase of the carrier is adjustable with the PHASE SHIFTER for maximum output from the lowpass filter. Phase reversals of 180 can be introduced with the front panel toggle switch. The DECISION MAKER will regenerate the original TTL sequence waveform.

T2 *Patch up the demodulator of Figure* 6. *The received signal will have come from the transmitter of Figure* 5. *Observe the output from the TUNABLE LP F, and confirm its appearance with respect to that transmitted. If the sequence is inverted then toggle the front panel* 180 ‘*switch of the receiver PHASE CHANGER.*

T3 *Set the on-board switch SW1 of the DECISION MAKER to accept NRZ-L coding. Use the gain control of the TUNABLE LPF to set the input at about the TIMS ANALOG REFERENCE LEVEL of* ±2 *volt peak. Adjust the decision point. Check the output.*

T4 *Observe the TTL output from the LINE-CODE DECODER. Confirm that the phase of the receiver carrier (for the NRZ-L line code) is still important.*

EXPERIMENT 12: QAM and 4-PSK

- ***The QPSK transmitter***
- ***The QPSK Demodulator***

Lab 8

Experiment 12: QAM and 4-PSK

ACHIEVEMENTS: *review of the quadrature amplitude modulator (QAM) in digital communications; as a generator of a quadrature phase shift keyed (QPSK, or 4-PSK) signal. Demodulation of QPSK.*

PREREQUISITES: *it would be advantageous to have completed some of the experiments of Volume A1 involving linear modulation and demodulation.*

.

ADVANCED MODULES: *DECISION MAKER. A total of three MULTIPLIER modules is required.*

PREPARATION

The QAM principle

Recall the experiment entitled *Phase division multiplex* in Volume A2. Two double sideband suppressed carrier (DSBSC) signals were combined on a common carrier (and so a common channel), by adding (multiplexing) them in phase-quadrature. In the analog environment the two analog messages are independent, and the signal is called quadrature amplitude modulation - QAM.

The QAM modulator was of the type shown in Figure 1 below. The two paths to the adder are typically referred to as the 'I' (inphase), and 'Q' (quadrature) arms.

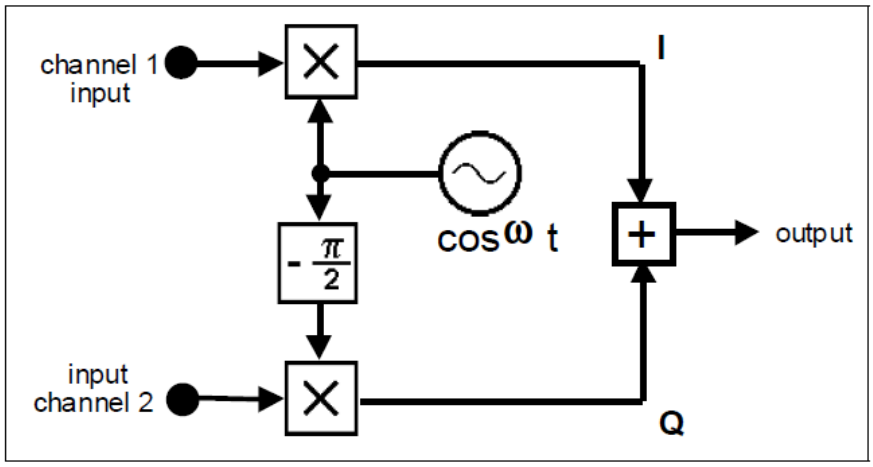


Figure 1: a quadrature modulator

Not shown in Figure 1 is any bandlimiting. In a practical situation this would be implemented either at message level - at the input to each multiplier - and/or at the

output of the adder. Probably both! The motivation for QAM comes from the fact that a DSBSC signal occupies twice the bandwidth of the message from which it is derived. This is considered wasteful of resources. QAM restores the balance by placing two independent DSBSC, derived from message #1 and message #2, in the same spectrum space as one DSBSC. The bandwidth imbalance is removed.

In digital communications this arrangement is popular. It is used because of its bandwidth conserving (and other) properties.

It is not used for multiplexing two independent messages. Given an input binary sequence (message) at the rate of n bit/s, two sequences may be obtained by splitting the bit stream into two paths, each of n/2 bit/s. This is akin to a serial-to-parallel conversion.

The two streams become the channel 1 and channel 2 messages of Figure 1.

Because of the halved rate the bits in the I and Q paths are stretched to twice the input sequence bit clock period.
The two messages are recombined at the receiver, which uses a QAM-type demodulator.
The two bit streams would typically be band limited and/or pulse shaped before reaching the modulator.

A block diagram of such a system is shown in Figure 2 below.

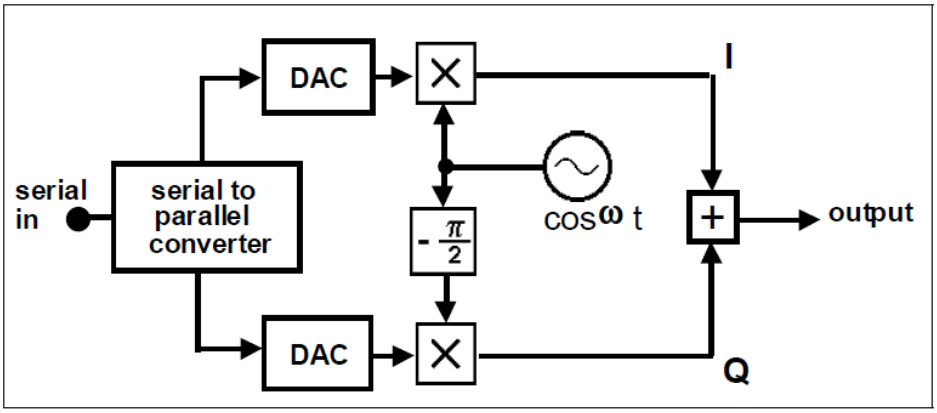


Figure 2: a QPSK modulator

QAM becomes QPSK

The QAM modulator is so named because, in analog applications, the messages do in fact vary the amplitude of each of the DSBSC signals.

In QPSK the same modulator is used, but with binary messages in both the I and Q channels, as described above.
Each message has only two levels, ±V volt. For a non-bandlimited message this does not vary the amplitude of the output DSBSC. As the message changes polarity this is interpreted as 180 phase shift, given to the DSBSC.

Thus the signal in each arm is said to be undergoing a 180 phase shift, or phase shift keying - or PSK.

Because there are two PSK signals combined, in quadrature, the two-channel modulator gives rise to a quadrature phase shift keyed - QPSK - signal.

Constellation

Viewed as a phasor diagram (and for a non-bandlimited message to each channel), the signal is seen to occupy any one of four point locations on the complex plane. These are at the corner of a square (a square lattice), at angles $\pi/4$, $3\pi/4$, $5\pi/4$ and $7\pi/4$ to the real axis.

You will see this *signal constellation* later in the experiment.

M-PSK and M-QAM

The above has described digital-QAM or QPSK. This signal is also called 4-PSK or 4-QAM. More generally signals can be generated which are described as M-QAM or M-PSK.

Here $M = 2\mathbf{L}$, where **L** = the number of levels in each of the I and Q arms. For the present experiment **L** = 2, and so M = 4.

The 'M' defines the number of points in the signal constellation.

For the cases M > 4 then M-PSK is not the same as M-QAM.

It is beyond the intended scope of this experiment to discuss these differences. But it is certainly worth your while to read further on the subject, and to discover the different constellations that these signals generate.

Refer to your text book for more detail.

See also the experiment entitled *Multi-level QAM and PSK* in this Volume.

The QAM receiver

The QAM receiver follows the similar principles to those at the transmitter, and is Illustrated in idealized from in the block diagram of Figure 3.

It is idealized because it assumes the incoming signal has its two DSBSC precisely in Phase quadrature. Thus only one phase adjustment is required.

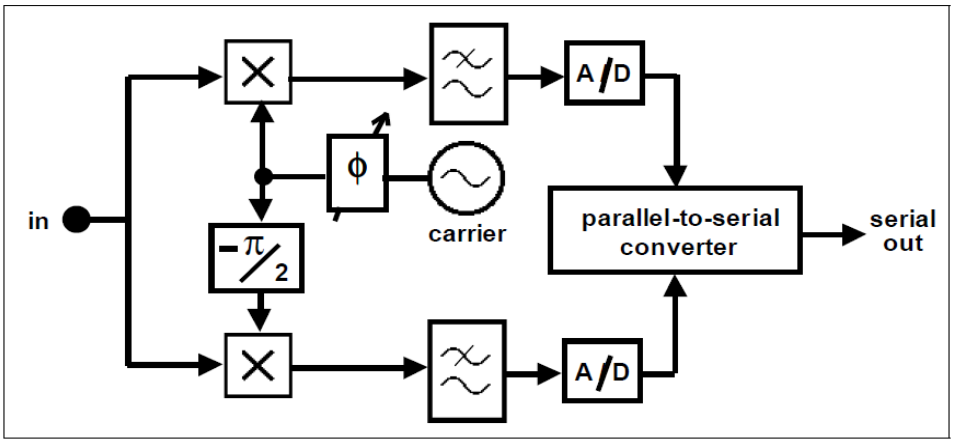


Figure 3: the QAM demodulator for QPSK

The parallel-to-serial converter block performs the following operations:

1. Regenerates the bit clock from the incoming data. See, for example, the experiment entitled *Bit clock regeneration* in this Volume.

2. Regenerates a digital waveform from both the analog outputs of the I and Q arms.

3. re-combines the I and Q signals, and outputs a serial data stream.

Not shown is the method of carrier acquisition. This ensures that the oscillator, which supplies the local carrier signal, is synchronized to the received (input) signal in both frequency *and* phase.

Experiment simplification

You are familiar with the practice of using a stolen carrier. This enables you to concentrate on a particular aspect of a system, without being obliged to spend time becoming involved with carrier acquisition, which can be a complex process.

Likewise, in this experiment, it is not necessary to become involved with details which are not of direct relevance. So two independent data sequences will be used at the input to the modulator, rather than having digital circuitry to split one data stream into two (the serial-to-parallel converter).

For the purposes of demonstration the above mentioned techniques simplify the model.
Two such independent data sequences, sharing a common bit clock (2.083 kHz), are available from a single SEQUENCE GENERATOR module. The data stream from which these two channels are considered to have been derived would have been at a rate of twice this - 4.167 kHz.

Low pass filter band limiting and pulse shaping is not a subject of enquiry in this experiment. So a single bandpass filter at the ADDER (summer) output will suffice, providing it is of adequate bandwidth. A 100 kHz CHANNEL FILTERS module is acceptable (filter #3).

EXPERIMENT

The QPSK transmitter

A model of the generator of Figure 1 is shown in Figure 4.

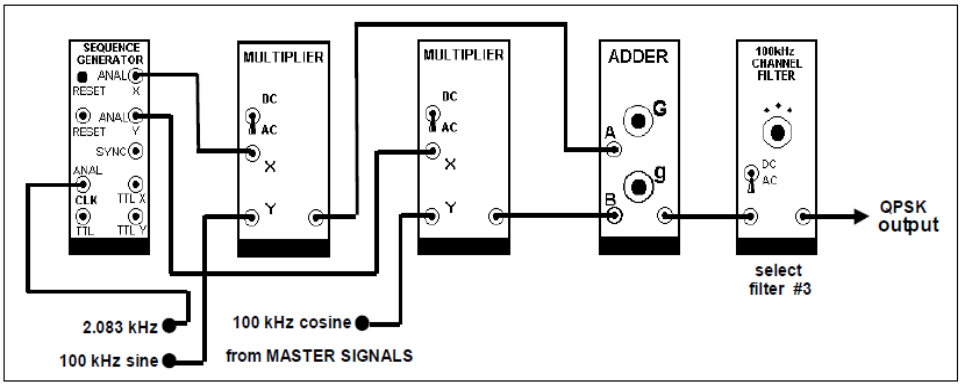


Figure 4: the QAM modulator for QPSK

T1 *patch up the modulator according to Figure 4. Set the on-board switch SW1 of the PHASE SHIFTER to HI. Select channel #3 of the 100 kHz CHANNEL FILTERS module (this is a bandpass filter of adequate bandwidth).*

T2 *there are no critical adjustments to be made. Set the signals from each input of the ADDER to be, say, 1 volt peak at the ADDER output.*

T3 *for interest predict the waveforms (amplitude and shape) at all interfaces, then confirm by inspection. What will be a suitable oscilloscope trigger in each case?*

Constellation

You can display the four-point constellation for QPSK:

T4 *set the oscilloscope in X-Y mode. With no input, select equal gains per channel. Locate the 'spot' in the center of the screen; then connect the two data streams entering the QAM to the scope X and Y inputs.*

The demodulator

Modelling of the demodulator of Figure 3 is straightforward. But it consumes a lot of modules. Consequently only one of the two arms is shown in Figure 5.

If you have insufficient modules to retain your QPSK modulator, then you can use a QPSK signal supplied at TRUNKS with which to test your demodulator.

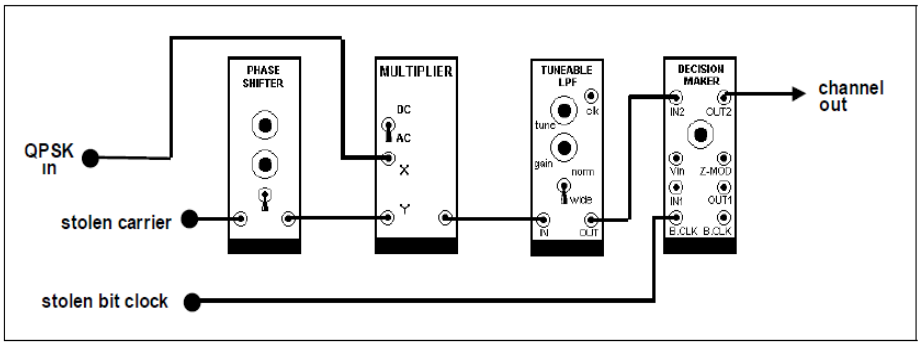


Figure 5: one channel of the demodulator

The PHASE SHIFTER can be used to select either channel from the QAM signal. If both channels re required simultaneously, as in practice, then a second, identical demodulator must be provided.

T5 *patch up the single channel demodulator of Figure 5.*

T6 *while watching the 'I' channel at the transmitter, use the PHASE SHIFTER to Match the demodulator output with it.*

T7 *while watching the 'Q' channel at the transmitter, use the PHASE SHIFTER to match the demodulator output with it.*

EXPERIMENT 13: Block Coding
&
Decoding

EXPERIMENT 14: Block Coding
&
Coding Gain

Lab 9

Experiment 13: Block Coding & Decoding

ACHIEVEMENTS: *viewing of a serial data stream before and after block encoding. Decoding. SNR improvement due to block coding.*

PREREQUISITES: *completion of the experiment entitled* ***PCM encoding***

ADVANCED MODULES: *PCM ENCODER, BLOCK CODE ENCODER, BLOCK CODE DECODER, LINE-CODE ENCODER.*

PREPARATION

Block coding

This experiment examines the BLOCK CODE ENCODER and BLOCK CODE DECODER modules.

Block coding refers to the technique of adding extra bits to a digital word in order to improve the reliability of transmission. The word consists of the message bits (often called information, or data) *plus* code bits. It may also, as in the present case, contain a frame synchronization bit.

A block code adds bits to existing message bits, or blocks, *independently* of adjacent blocks 1.

In this experiment the blocks will be prepared by the PCM ENCODER module. These blocks were examined in the experiment entitled *PCM encoding*.

PCM encoded data format

When extra code bits are added to a PCM word (initially containing only message bits) then the word will get longer. If the bit rate remained the same then the message bits would arrive at a slower rate than before. To maintain the same message rate the bit rate would need to be increased. This would require an increased transmission bandwidth.

In the TIMS PCM ENCODER module a different scenario has been adopted.

The PCM word has been generated from the input message and placed in a frame of fixed length. These are the *message bits*. Not all slots in the frame are used. When extra *coding bits* are added, they go in the previously unused slots. Thus, in either case (extra code bits or not):

• The frame length remains the same.

• The message rate remains the same.
• The channel bandwidth will remain the same, as the bit rate has not changed.

The TIMS arrangement may waste time in the un-block-coded state (there are three unused slots in the frame), and so be called inefficient (which it is). But it is convenient for our purpose.

The PCM ENCODER module was examined in the experiment entitled *PCM encoding*.

Block code format

The BLOCK CODE ENCODER module is designed to expect input blocks of length eight slots, where some of these slots are empty. These come from the PCM ENCODER module (in the 4-bit mode).

The incoming data frame is illustrated in Figure 1 below.

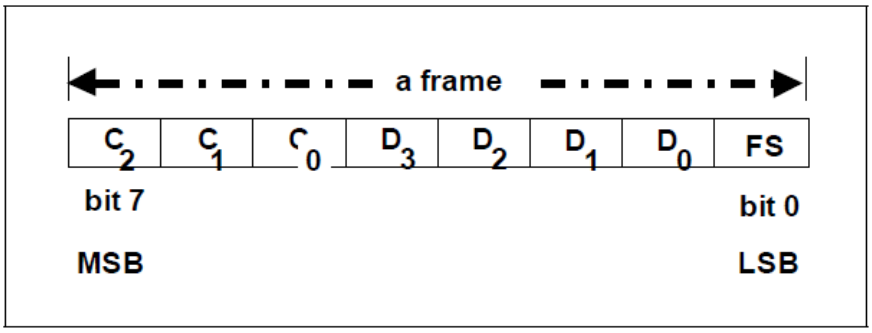


Figure 1: a data frame of eight slots, one per clock period

The message bits are shown as D3, D2, D1, and D0, where D3 is the most significant bit of the message.

The frame synchronization bit is shown as FS.

The slots marked C2, C1, and C0 will be used by the BLOCK ENCODER for code bits.

For the BLOCK CODE ENCODER module to function correctly it must always receive three digital signals:

1. TTL binary data in an 8-bit wide frame (typically from a PCM ENCODER in 4- bit mode). The data must occupy frames 4, 3, 2, and 1 (as defined in Figure 1 above).
2. a TTL clock, to which the incoming data is synchronized. Typically this will be at 2.083 kHz (the module is restricted to a clock rate below 8 kHz).
3. a TTL frame synchronization signal FS, which signals the *end* of the frame.

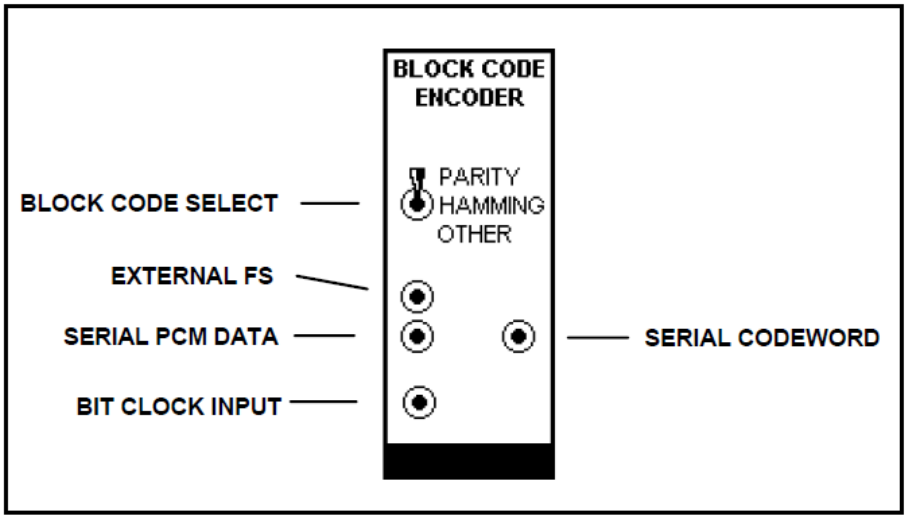


Figure 2: front panel layout - ENCODER

The front panel of the module is illustrated in Figure 2 above. The features should be self-explanatory, except for the BLOCK CODE SELECT toggle switch.

Block code select

Each BLOCK CODE ENCODER module offers three different coding schemes. These are contained in an EPROM. More than one EPROM is available, any one of which can be installed in the module. The codes they offer are set out in Table 1 below.

EPROM	code 1	code 2	code 3
BLKe1.x	even parity - single bit error detect	Hamming (7,4) - single bit error correct	*Setup - with C_x bit error detect
BLKe2.x	even parity - single bit error detect	Hamming (7,4) - single bit error correct	odd parity - single bit error detect
BLKe3.x	even parity - single bit error detect	Hamming (7,4) - single bit error correct	Cyclic

Table 1: EPROM codes

Any one of the three codes in the installed EPROM can be selected with the front panel toggle switch.

In performing parity checks the FS bit is ignored.

Typical usage

In a typical digital communications system, the configuration at the transmitter might appear as in the block diagram of Figure 3 below.

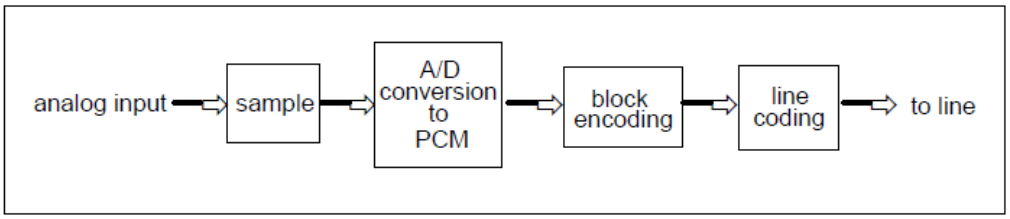


Figure 3: disposition of the block encoder

Block decoding

The signals from the BLOCK CODE ENCODER need to be interpreted by a complementary BLOCK CODE DECODER module, the front panel of which is illustrated in Figure 4 below.

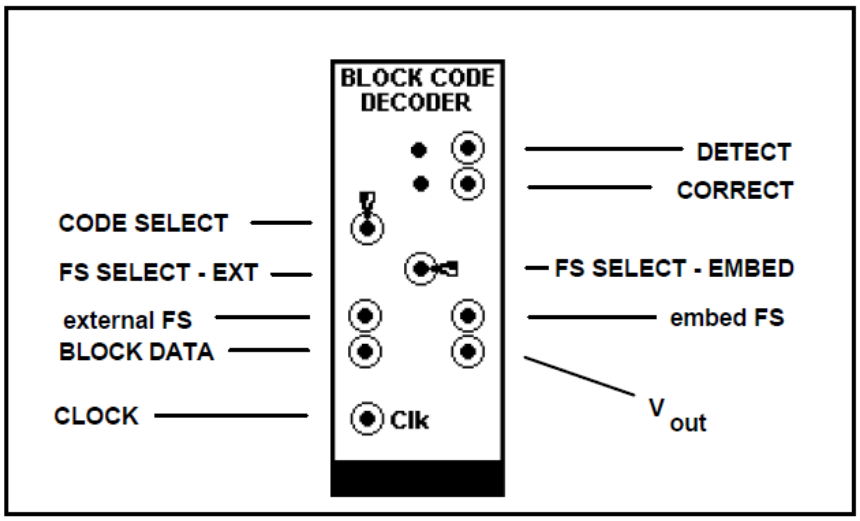


Figure 4: front panel layout - DECODER

The front panel of the decoder module is illustrated in Figure 4 above. The features should be self-explanatory, except for the following:

• **DETECT**: for codes which can detect but not correct errors. The LED flashes when an error is detected, but not corrected. There is a TTL high, one bit wide, at the adjacent socket, during the frame in which the error occurred.
• **CORRECT**: for codes which can detect and correct errors. The LED flashes when an error is detected and corrected. There is a TTL high, one bit wide, at the adjacent socket, during the frame in which the error occurred.

The **DETECT** and **CORRECT** outputs (LED and bit-wide TTL HI) are mutually exclusive.

• **FS SELECT - EXT**: frame synchronization may be attained by accepting a 'stolen' FS signal from the transmitter, patched to the FS input socket.
• **FS SELECT - EMBED**: frame synchronization may be achieved automatically, using the embedded information in the LSB of the frame itself. For verification the recovered FS signal is available at the FS output socket. When a stolen FS signal is used there is *no output* from this socket.

EXPERIMENT

This experiment is intended to help familiarize you with some aspects of the operation of the BLOCK CODE ENCODER. It will also confirm the decoding process performed by the BLOCK CODE DECODER module.

Encoding

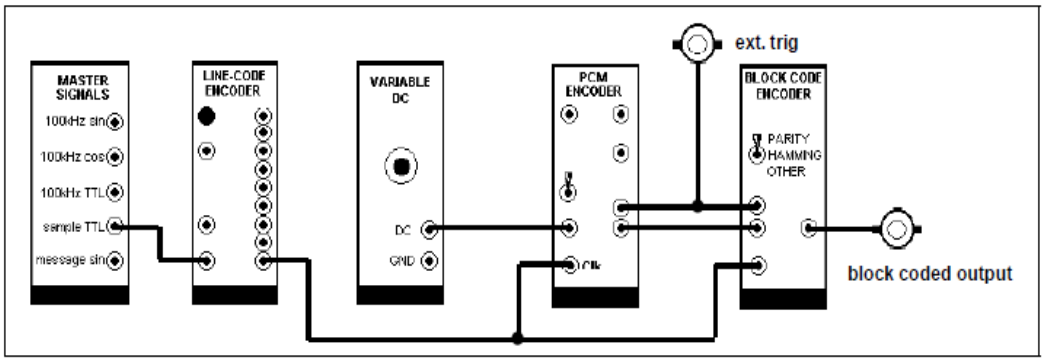


Figure 5: block code encoding

The BLOCK CODE ENCODER requires a TTL clock near 2 kHz, and it must be operated at a clock speed below 8 kHz.

You may have notice that it is customary TIMS practice (but not mandatory) to use a clock locked to the MASTER 100 kHz source. Typically this has been the 8.333 kHz TTL signal from the MASTER SIGNALS module. Since the BLOCK CODE ENCODER requires something lower than this, a convenient source is obtained by dividing this by 4. The LINE-CODE ENCODER module has just such a divider (and typically forms part of a data transmission system). The model of Figure 5 above illustrates this method.

For stable oscilloscope displays from the PCM source a DC message is used, together with a suitable source of external triggering signal.

T1 *patch up the model of Figure 5.*

T2 *set up simultaneous displays of the PCM input, and the block coded output, of the BLOCK CODE ENCODER, over two or three frames. Spend some time investigating different methods of oscilloscope synchronization.*

T3 *verify, where possible, that each of the codes has been implemented correctly.*

Decoding

Having successfully block encoded a PCM signal, it is time to demonstrate its decoding. For this purpose transmission will be via a direct connection.

You will have noticed the ERROR INDICATION front panel LEDs on the decoder. These will be useful when transmission via a noisy, bandlimited channel, with the inclusion of line encoding, is examined in a later experiment. There the benefits of block coding will be demonstrated and evaluated.

Patching for the decoding process is shown in Figure 6.

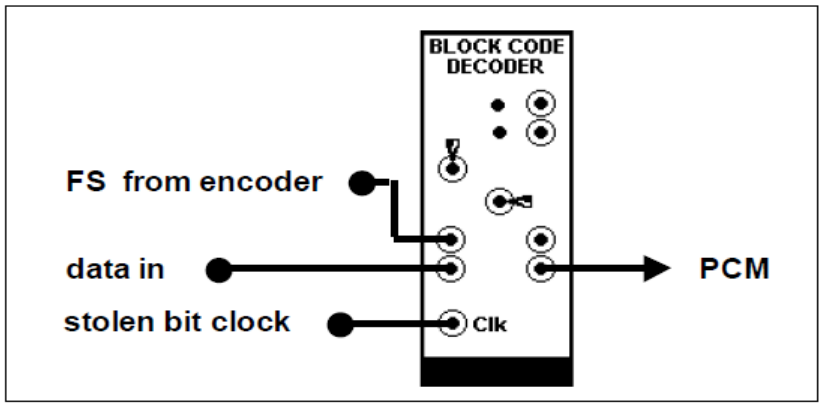


Figure 6: block code decoding

Note that a stolen bit clock is used.

Frame synchronization can be achieved by either a 'stolen' FS signal from the encoder, or by internal decoding of the alternating pattern of 1 - 0 - 1 - 0 – 1 embedded as the LSB of the PCM code word (in location '0' of the frame). This scheme was introduced in the experiment entitled *PCM decoding*.

T4 *patch up the BLOCK CODE DECODER according to Figure 6. This uses the 'stolen' frame synchronization signal FS from the transmitter, connected to the EXT. FS input, and selected with the front panel toggle switch FS SELECT.*

T5 *verify that successful decoding back to the original PCM is possible for all codes.*

T6 *switch the front panel toggle switch FS SELECT to EMBED. Confirm that the internal circuitry for extracting the frame synchronization signal FS from the PCM signal itself is operating correctly. Refer to the Appendix to this experiment for more information.*

APPENDIX

Automatic frame synchronization

The BLOCK CODE DECODER module has built-in circuitry for locating the position of each frame in the serial data stream. The circuitry looks for the embedded and alternating '0' and '1' every 8 bits (which occur in the LSB position of each frame).

The search is made by examining a section of data whose length is a multiple of eight bits.

The length of this section can be changed by the on-board switch SW3. Under noisy conditions it is advantageous to use longer lengths.

The switch settings are listed in Table A-1 below.

left toggle	right toggle	groups of eight bits
UP	UP	4
UP	DOWN	8
DOWN	UP	16
DOWN	DOWN	32

Table A-1: synchronization search length options

Experiment 14: Block Coding & Coding Gain

ACHIEVEMENTS: *transmission of a block coded PCM signal over a noisy baseband channel; comparison of 7-bit linear versus 4-bit block coded PCM.*

PREREQUISITES: *completion of the experiment entitled* ***Block coding and decoding****.*

ADVANCED MODULES: *PCM ENCODER, PCM DECODER, BLOCK CODE ENCODER, BLOCK CODE DECODER, LINE-CODE DECODER, LINE-CODE ENCODER, NOISE GENERATOR. A WIDEBAND TRUE RMS METER is optional.*

PREPARATION

This experiment is an extension of the introductory experiment entitled *Block coding and decoding* in this Volume.

The extension involves the transmission of the coded signal via a noisy baseband channel.

Since the message is analog, the evaluation of performance is made by measuring the recovered message signal-to-noise ratio under different conditions. This can be a quantitative measurement, using the WIDEBAND TRUE RMS METER, otherwise qualitative by observation of the recovered periodic message waveform.

You are free to determine which measurements are of interest. The Tasks outlined below are there to guide you in setting up the system.

System parameters

Clock speeds and message frequencies are determined by the DECISION MAKER, which has an upper rate of operation. Thus:

1. The DECISION MAKER module has a clock rate limited to the vicinity of 2 kHz, so a rate of 2.083 kbit/sec has been chosen.

2. The BLOCK CODE ENCODER operates on blocks (or *frames*) of eight bits. These are provided by a PCM ENCODER

3. To provide room for coding bits within the frame, which is of 8-bit width, the PCM ENCODER will operate in the 4-bit mode.

4. For an eight bit frame the sampling rate will be 260 samples/sec (2083/8).

5. The maximum message frequency will be limited to 130 Hz (Nyquist). This is below the range of the AUDIO OSCILLATOR module. But there are four fixed frequency sinusoidal (and near-sinusoidal) messages available within the PCM ENCODER module. Their frequencies, and access details, are given in the Appendix to this experiment.

A TUNEABLE LPF will be used for the bandlimiting channel,

A simplified block diagram of the system is shown in Figure 1.

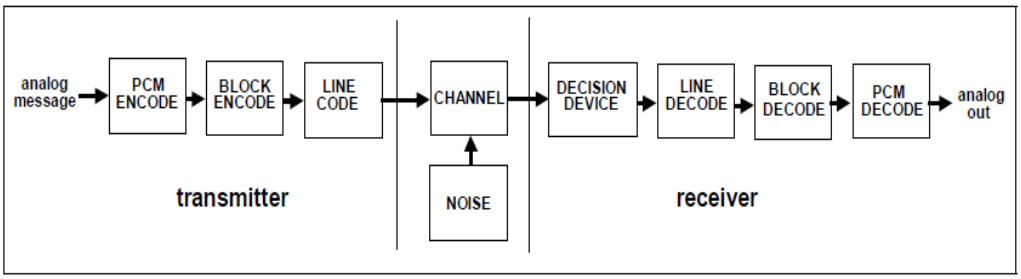


Figure 1: the system block diagram

EXPERIMENT

The block diagram of Figure 1 can be modelled as shown in Figures 2 to 4

Transmitter

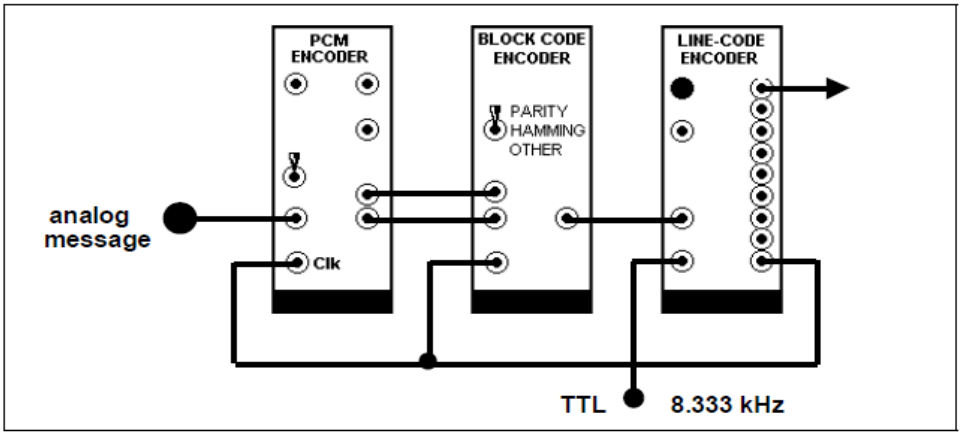


Figure 2: the transmitter

T1 *patch up the transmitter model of Figure 2. It is convenient to use DC for the message during the setting up procedure. Select the 4-bit LINEAR PCM code, and PARITY block coding.*

Channel

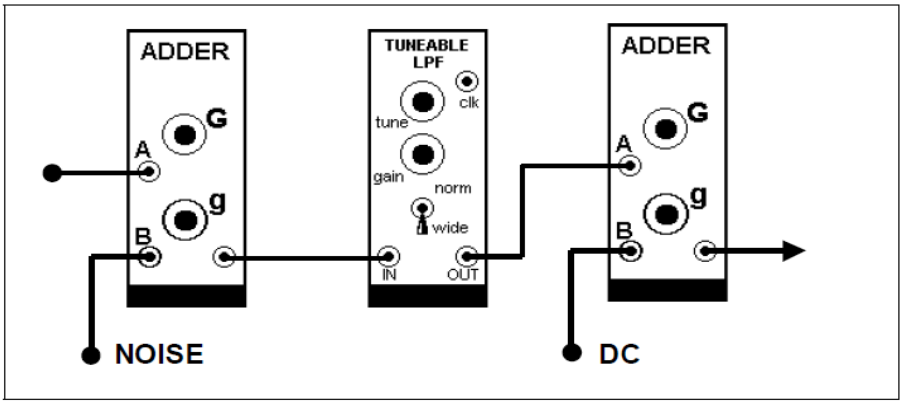


Figure 3: the channel model

T2 *patch up the noisy channel according to Figure 3.*

T3 *disconnect any DC from the output ADDER input.*

T4 *observe the wave shape at the channel output for full channel bandwidth, then tune the filter until there is obvious bandlimiting.*

T5 *with full output from the noise source set the SNR at the DECISION MAKER input output to a few dB (by oscilloscope observation), and at about the TIMS ANALOG REFERENCE LEVEL.*

T6 *reduce the noise by the full available attenuation of the NOISE GENERATOR front panel attenuator.*

The signal is now ready for demodulation, presumably without errors, since the SNR should be well above 0 dB (above 22 dB).

Receiver

T7 *patch up the receiver of Figure 4 below.*

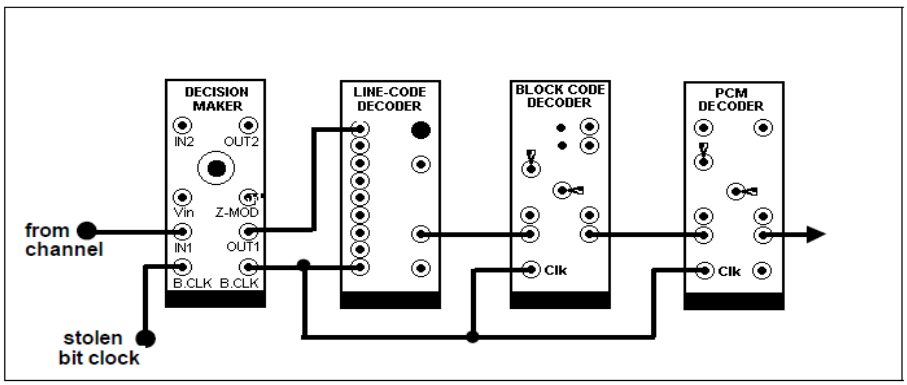


Figure 4: the receiver model

T8 *ensure matching line codes are selected at both transmitter and receiver. NRZ-L is suggested.*

T9 *at the BLOCK CODE DECODER select frame synchronization using the Embedded frame synch. Signal, and PARITY block coding.*

T10 *at the PCM DECODER select 4-bit LINEAR decoding and frame synchronization using the embedded frame synch. Signal.*

T11 *check the bit clock patching. Note that the STROBE from the LINE-CODE DECODER is not used.*

Now check signals and waveforms from input to output. The message is DC. The oscilloscope displays should be stable and identifiable if the FS signal is used as the oscilloscope synchronization signal.

T12 *identify a frame at the output of the PCM ENCODER (CH1-A), and follow it through the BLOCK ENCODER and the LINE-CODE ENCODER to the channel input.*

T13 *while looking at your chosen frame entering the channel (CH1-A), locate it on the other oscilloscope trace (CH2-A) at the channel output.*

T14 *move CH2-A forward to the output of the DECISION MAKER and confirm the regeneration of the desired wave shape.*

T15 *move CH1-A back one stage to the input of the LINE-CODE ENCODER, and CH2-A forward to the LINE-CODE DECODER output. Confirm the two waveforms agree in shape, and that there is a delay.*

T16 *if considered necessary, fine trim the DC level to match the threshold (about +25 mV) of the DECISION MAKER.*

Evaluation

You will not be making bit error rate (BER) measurements using the BER instrumentation techniques investigated in earlier experiments. These required a precise knowledge of the signal-to-noise ratio at the decision device *input,* and a known data sequence for bit-by-bit comparison.

Instead you are looking for changes in SNR (or waveform quality) of the recovered *output* message. A measure of error rates is available from the error detector circuitry of the BLOCK CODE DECODER module. The error rate can be used as a reference condition.

The noise at the output will be made up of quantization noise (unavoidable), errors introduced by the noise added to the signal, and perhaps distortion components.

There are many ‘A - B’ comparisons which can now be made.

Most evaluations will be qualitative, by observing the recovered sinusoidal message via the built-in reconstruction filter of the PCM DECODER, under the two conditions ‘A’ and ‘B’.

The error counter in the BLOCK CODE DECODER will be used as a guide to the digital errors (caused by noise) in the ‘A’ state, but cannot be used as a comparison measure, since the ‘B’ state will generally not be using block coding.

The technique is to reduce the SNR until a change is seen in the reconstituted message waveform under condition ‘A’. Then, with this SNR, to switch to condition ‘B’ and to look for a variation in the message waveform (or, with the WIDEBAND TRUE RMS METER, to measure a change of SNR, or SNDR 1).

It is important that the reconstruction filter does not prevent message distortion being observed. Thus it is important to ensure that:

• The reconstruction filter bandwidth is close to the Nyquist bandwidth (ie, as wide as possible)
• The message frequency is low enough to allow the passage of at least an even (2nd) and an odd (3rd) harmonic through the reconstruction filter.

Suggested comparisons could be:

• 4-bit PCM encoding with and without block coding.
• 7-bit linear PCM encoding (no block coding) versus 4-bit linear PCM.
• 7-bit linear PCM encoding (no block coding) versus 4-bit linear PCM encoding with block coding.

Remember that there are three block codes to investigate for each case involving block coding.

T17 *carry out as many of the above ‘A - B’ comparisons as you consider important. Compare with expectations.*

APPENDIX A

For a MASTER CLOCK of 8.333 kHz, Table A-1 below gives the frequencies of the synchronized message at the SYNC. MESSAGE output for the setting of the onboard switch SW2.

For other clock frequencies the message frequency can be calculated by using the 'divide by' entry in the Table.

These messages are periodic, but not necessarily sinusoidal in shape. The term 'sinuous' means sine-like.

LH toggle	*RH toggle*	*divide clock by*	*freq with 8.333kHz clock*	*approx. ampl. and waveform*
UP	UP	32	260.4 Hz	0.2 V_{pp} sine
DOWN	UP	64	130.2 Hz	2.0 V_{pp} sine
UP	DOWN	128	65.1 Hz	4.0 V_{pp} sinuous
DOWN	DOWN	256	32.6 Hz	4.0 V_{pp} sinuous

Table A-1

EXPERIMENT 15: BER Measurement in the Noisy Channel

Lab 10

Experiment 15: BER Measurement in the Noisy Channel

ACHIEVEMENTS: *ability to set up a digital communications system over a noisy, bandlimited channel, with provision for line-coding, and instrumentation for BER measurements. This system will be used for many future experiments.*

PREREQUISITES: *completion of the first five experiments in Volume D1 would be an advantage, especially those entitled* ***the noisy channel model****, and* ***Detection with the DECISION MAKER****.*

EXTRA MODULES: *LINE-CODE ENCODER, LINE-CODE DECODER, DECISION MAKER, NOISE GENERATOR, ERROR COUNTING UTILITIES, WIDEBAND TRUE RMS METER, an extra SEQUENCE GENERATOR, BASEBAND CHANNEL FILTERS.*

PREPARATION

Overview

This experiment serves as an introduction to bit error rate (BER) measurement. It models a digital communication system transmitting binary data over a noisy, bandlimited channel. A complete instrumentation setup is included, that allows measurement of BER as a function of signal-to-noise ratio (SNR).

> *Many variations of this system are possible, and the measurement of the performance of each of these can form the subject of separate experiments.*

In this first experiment the system is configured in its most elementary form.

Other experiments can add different forms of message coding, line coding, different channel characteristics, bit clock regeneration, and so forth.

The basic system

A simplified block diagram of the basic system is shown in Figure 1 below.

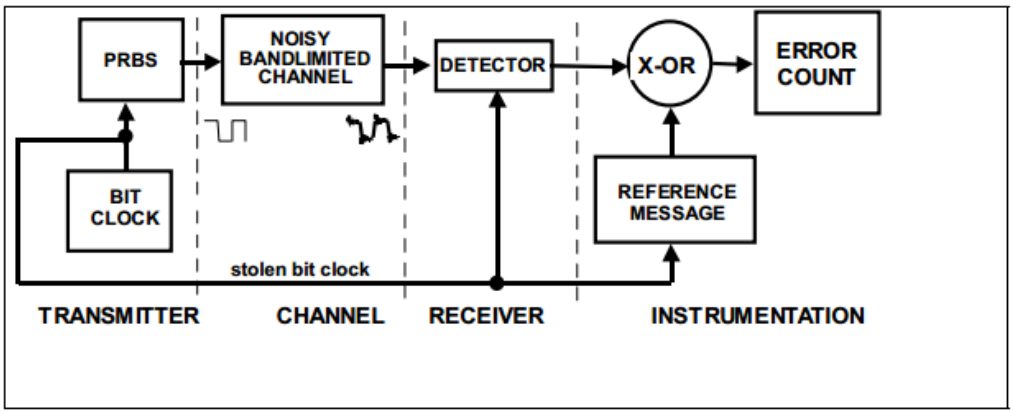


Figure 1: block diagram of system

The system can be divided into four sections:

The transmitter

At the transmitter is the originating message sequence, from a pseudo random binary sequence (PRBS) generator, driven by a system bit clock.

The channel

The channel has provision for changing its bandlimiting characteristic, and the addition of noise or other sources of interference.

The receiver

The receiver (detector) regenerates the transmitted (message) sequence. It uses a stolen bit clock.

The BER instrumentation

The instrumentation consists of the following elements:

1. a sequence generator identical to that used at the transmitter. It is clocked by the system bit clock (stolen, in this case). This sequence becomes the reference against which to compare the received sequence.

2. a means of aligning the instrumentation sequence generator with the received sequence. A sliding window correlator is used.

3. a means of measuring the errors, after alignment. The error signal comes from an X-OR gate. There is one pulse per error. The counter counts these pulses, over a period set by a gate, which may be left open for a known number of bit clock periods.

A more detailed description

Having examined the overall operation of the basic system, and gained an idea of the purpose of each element, we proceed now to show more of the specifics you will need when modeling with TIMS.

So Figure 1 has been expanded into Figure 2 below.

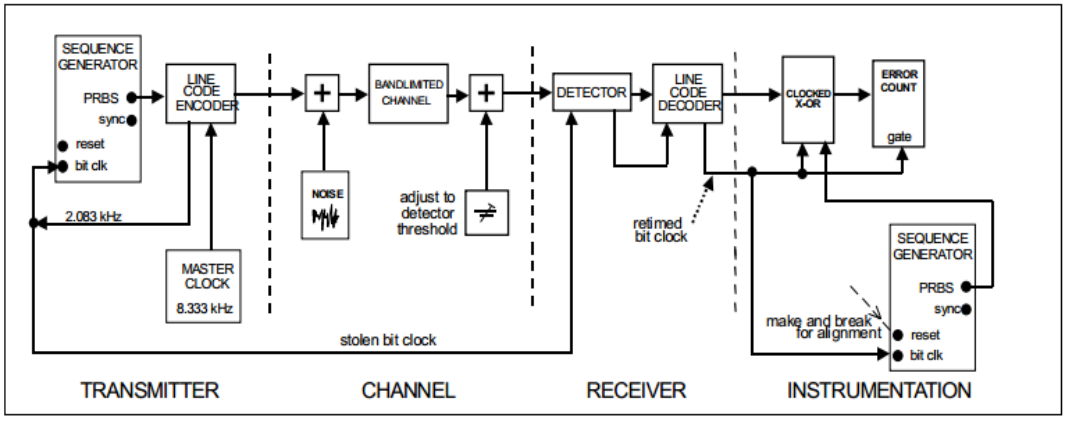


Figure 2: block diagram of system in more detail

The extra detail in Figure 2 includes:

1. Provision for transforming the data before transmission, using any one of a number of line codes. In this experiment we use the NRZ-L code which provides level shift and amplitude scaling, to suit the analog channel.

2. Bit clock generation. Because the line coder requires quarter-bit-period timing information, it is driven by a master clock at four-times the bit-clock rate. The timing information is obtained by dividing the master clock by four (within the LINE-CODE ENCODER). This divided-by-four version of the master clock becomes the system bit clock.

3. Provision for adding noise to the channel via the adder on the input side of the bandlimiting channel.

4. Inclusion of an ADDER on the output side of the channel. This restores the polarity change introduced by the input ADDER (for line codes which are polarity sensitive). It also provides an opportunity to fine-trim the DC level to match the threshold of the DECISION MAKER.

5. A decoder for the line code.

6. Instrumentation for SNR adjustment (not shown) and BER measurement.

Theoretical predictions

Bit error probability (P_B) is a function of En/No. For matched filter reception of bipolar baseband signaling it has been shown that:

$$P_B = Q\left(\sqrt{\frac{2E_b}{N_o}}\right) \qquad \text{..... 1}$$

The symbols are defined in the Chapter entitled BER instrumentation macro module in this Volume

You will measure not P_B, but BER; and not En/No, but SNR. Figure 3 shows theoretical predictions, based on eqn(1) above.

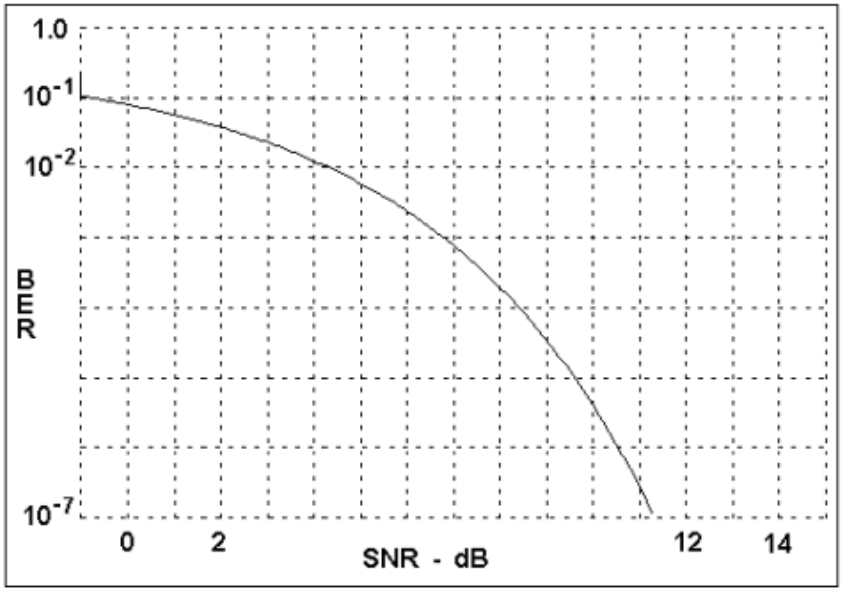


Figure 3 theoretical expectations - BER versus SNR (for bi-polar signalling)

EXPERIMENT

Familiarity with the setting up of a transmitter, receiver, and noisy channel, using a stolen clock for bit clock synchronization, and the sliding window correlator for sequence alignment, is assumed.

The system under examination, the principle of which is illustrated in block diagram form in Figure 1, is shown modeled by the patching diagram of Figure 4 on the next page. Within that diagram is included the macro CHANNEL MODEL module, and the BER INSTRUMENTATION macro module.

The macro CHANNEL MODEL module is reproduced in Figure 4 below.

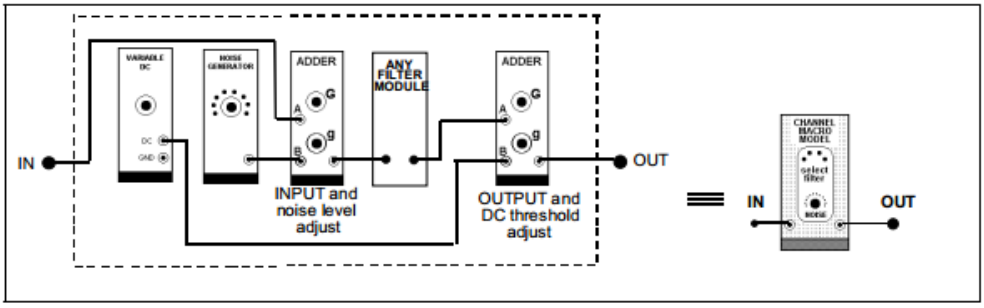


Figure 4: details of the macro CHANNEL MODEL module

This is the first time the pulse counting capabilities of the ERROR COUNTINGUTILITIES module have been used. For a complete description of the characteristics and behavior of the module see the appendix A.

Modeling the transmission system

The system to be modeled is shown in Figure 5. It will be patched up systematically, section by section, according to the scheme detailed below. It has not been cluttered by showing oscilloscope connections. You should set up the SCOPE SELECTOR for maximum usage of the facility for toggling between the A and B options for each channel.

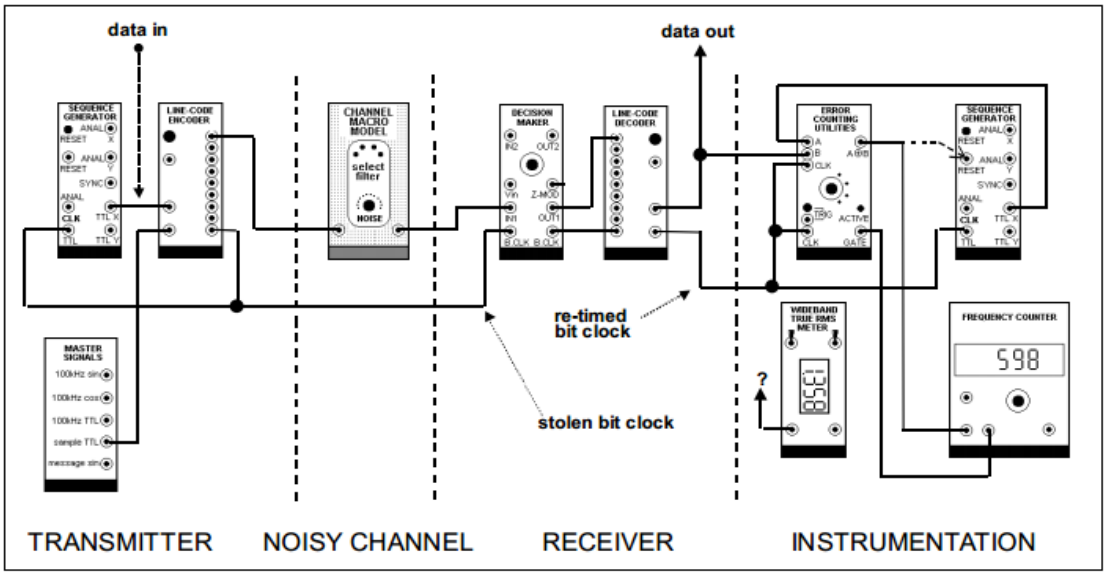


Figure 5: the TIMS model of Figure 2

1. 0: the transmitter

T1.1 *patch the transmitter according to Figure 5, from a SEQUENCE GENERATOR (set to a short sequence - both toggles of SW2, on circuit board, UP), a LINE-CODE ENCODER (using NRZ-L), and the MASTER SIGNALS module. Note that the LINE-CODE ENCODER accepts the master clock, which is the 8.333 kHz TTL 'sample clock' from the MASTER SIGNALS module, and divides it by four to produce the 2.083 kHz system bit clock for the SEQUENCE GENERATOR.*

T1.2 *press the reset on the LINE-CODE ENCODER. Check on CH1-A that a short TTL sequence has been generated by the SEQUENCE GENERATOR.*

T1.3 *simultaneously with the previous observation on CH1-A, check the NRZ-L output of the LINE-CODE ENCODER on CH2-A. Relative to the TTL on CH1-A it will be delayed half a bit period. This is the signal being transmitted to the channel. Confirm the code format.*

2.0: the channel model

The macro CHANNEL MODEL module is shown modelled in Figure 4.

T2.1 *patch up the channel according to Figure 4 and insert it into the position shown in Figure 5.*

T2.2 *set the front panel attenuator of the NOISE GENERATOR to maximum output; but reduce the channel noise to zero by rotating the INPUT ADDER gain control 'g' fully anti-clockwise.*

T2.3 *adjust the amplitude of the signal into the BASEBAND CHANNEL FILTERS module to near the TIMS ANALOG REFERENCE LEVEL (say, 2 volt peak-to-peak)*

with the INPUTADDER gain control 'G'. This level will need resetting when noise is added.

T2.4 *select channel #3 of the BASEBAND CHANNEL FILTERS module.*

T2.5 *set the gain of the DC threshold adjustment path through the OUTPUT ADDER to zero.*

T2.6 *adjust the amplitude of the signal out of the CHANNEL MODEL to, say, 2 volt peak-to-peak with the OUTPUTADDER gain control 'G'. The gain through the channel is now unity.*

T2.7 *confirm that the signal at the OUTPUT ADDER, although of different shape, and further delayed, is clearly related to the input sequence.*

When tracing the sequence through the system, notice that there is a polarity inversion introduced by the INPUTADDER of the channel, and a second inversion introduced by the OUTPUTADDER.

3.0: the receiver

The receiver consists of the DECISION MAKER and LINE-CODE DECODER modules.

T3.1 *before plugging in the DECISION MAKER:*

a) switch the on-board switch SW2 to 'IN' (DECISION POINT can now be adjusted with the front panel control).

b) select the expected line code with the on-board rotary switch SW1 (upper rear of board). For this experiment it is NRZ-L.

T3.2 *trigger the oscilloscope from the SYNC output of the transmitter SEQUENC GENERATOR. Check that the reconstructed 'analog' output from the DECISION MAKER is a delayed version of, but otherwise the same shape as, that at the channel input.*

T3.3 *patch up the LINE-CODE DECODER, selecting the NRZ-L output*

T3.4 *press the reset on the LINE-CODE DECODER. Check that the TTL output sequence is identical, except for a delay, with that at the transmitter SEQUENCE GENERATOR output.*

Do not proceed unless these two TTL signals are identical!

4. 0: the BER measurement instrumentation

The transmission system is now fully set up. You will now proceed to verify its overall operation.

The BER measurement instrumentation system is used to generate an identical sequence to that transmitted, and aligned with that from the receiver detector. These two sequences will be compared, bit by bit, and any disagreements counted. The count is made over a pre-determined number of bit clock periods, and so the bit error rate (BER) may be calculated.

You will record the BER for various levels of noise, and compare with theoretical expectations.

T4.1 *patch up according to Figure 5. Note the instrumentation (receiver) SEQUENCE GENERATOR uses the LINE-CODE DECODER strobe as its bit clock. Trigger the oscilloscope for a snapshot. Check that there is a short sequence coming from the instrumentation SEQUENCE GENERATOR output.*

T4.2 *see the Appendix to this experiment for a short description of the ERROR COUNTING UTILITIES module, including on-board jumper and switch settings. Plug it in. Check that the line from the X-OR output to the instrumentation SEQUENCE GENERATOR RESET is open.*

T4.3 *observe the two inputs to the X-OR gate simultaneously. It is unlikely that they are aligned, but they should be synchronized.*

Your good work is about to be rewarded with the sight of the two sequences snapping into alignment.

T4.4 *momentarily closes the line from the X-OR output to the instrumentation SEQUENCE GENERATOR RESET. Confirm that the two sequences, already synchronized, are now aligned.*

If you want to see the sliding window correlator at work again, press the reset on the Instrumentation SEQUENCE GENERATOR, and alignment will be lost. Re-align by repeating the last Task.

T4.5 *set the FREQUENCY COUNTER to its COUNT mode, and patches it into the system, complete with the gate signal from the ERROR COUNTING UTILITIES module.*

T4.6 *witch the gate of the ERROR COUNTING UTILITIES, with the PULSE COUNT switch, to be active for* 10^5 *bit clock periods*

T4.7 *to make an error count:*

a) Reset the FREQUENCY COUNTER.

b) Start the error count by pressing the TRIG button of the ERROR COUNTING UTILITIES module.

The 'active' LED on the ERROR COUNTING UTILITIES module will light, and remain alight until 90% of the count is completed, when it will blink before finally extinguishing, indicating the count has concluded.

With no noise there should be no errors.

> ***Warning****: every time a count is initiated one count will be recorded immediately. This is a 'confidence count', to reassure you the system is active, especially for those cases when the actual errors are minimal. It does not represent an error, and should always be subtracted from the final count.*

Despite the above single confidence-count you may wish to make a further check of the error counting facility, before using noise.

T4.8 *if the ERROR COUNTING UTILITIES GATE is still open press the instrumentation SEQUENCE GENERATOR reset button (else press the TRIG to open the GATE). The sequences should now be out of alignment.*

The counter will start counting (and continue counting) errors until the GATE huts. It will record a count of between 2 and 10^n (with the PULSE COUNT switch set to make 10^n counts). You will record a different count each time this is repeated. Why would this be?

Well done!

You have just completed a major setting-up procedure. If it was achieved without any problems you are to be congratulated! Although TIMS itself will behave reliably, it is easy to make patching errors, and their discovery and rectification is all part of the learning process.

You are now almost ready to sit back and let TIMS do the measurements for you.

5.0 errors counting with noise

Preparation

T5.1 *increases the message sequence length of both SEQUENCE GENERATOR modules (both toggles of SW2 DOWN).*

T5.2 *re-establish sequence alignment by pressing all the reset buttons, in order input to output, then momentarily connect the X-OR output to the instrumentation SEQUENCE GENERATOR RESET input.*

Adding noise - principle

It is now time to add the noise to the signal. Noise must be introduced before bandlimiting, since the channel bandlimiting filters are required to bandlimit the noise as well.

The noise from the NOISE GENERATOR is wideband. Its peak amplitude must not overload an analog module, so its output has been restricted to 4 volt peak-to-

peak (the TIMS ANALOG REFERENCE LEVEL). As soon as it is bandlimited, this amplitude is reduced. Amplification cannot be used to bring it up to a convenient level until after bandlimiting. But by this time the signal has been added, so that is not possible.

So the only way to obtain a small signal-to-noise ratio (relatively high noise) is to reduce the signal level. This is done with the INPUTADDER.

To set the noise level:

a) Remove the signal from the channel input
b) Add as much noise as is available, to implement the worst SNR possible, by maximizing the gain through the INPUTADDER, and setting the attenuator of the NOISE GENERATOR for maximum noise output. The SNR can later be increased - less noise - with this attenuator.
c) Measure the noise level into the DECISION MAKER with the WIDEBAND TRUE RMS METER. Then remove the noise, replace the signal, and adjust it to the same level.
d) Replace the noise. The SNR is 0 dB. The system is now set up for the worst conditions under which measurements are to be made. From now on the SNR will be improved, in calibrated steps of the NOISE GENERATOR attenuator, and BER measurements recorded.

The above steps will now be implemented.

Adding noise - practice

T5.3 *Patch both the oscilloscope and the WIDEBAND TRUE RMS METER to observe the signal at the output of the channel.*

T5.4 *reduce the signal amplitude to zero with the 'G' gain control of the INPUT ADDER.*

T5.5 *Set the attenuator of the NOISE GENERATOR for maximum output. Increase the noise level into the channel, with the INPUTADDER, to maximum.* ***Record the reading of the rms meter (N volt rms amplitude).***

T5.6 *Remove the noise by unplugging the patch cord from the INPUTADDER.*

T5.7 *Introduce some signal with the 'G' control of the channel INPUTADDER, until the rms meter is reading the same as the previous noise reading.* ***Record this reading (S volt rms amplitude).***

T5.8 *replace the noise. Do not disturb the INPUTADDER gain settings from now on!*

T5.9 *check the signal level at the channel output. Use the 'G' gain control of the OUTPUTADDER to raise the input level to the DECISION MAKER to the TIMS ANALOG REFERENCE LEVEL (4 V peak-to-peak is allowable, although there may be insufficient gain in the ADDER).*

The SNR is now set up to a reference value $10\log_{10}\left[\frac{S^2}{N^2}\right]dB$

The effect of any DC threshold of the DECISION MAKER must be offset with DC introduced by the OUTPUTADDER.

T6.*4 remove both inputs from the INPUTADDER. Using both the VARIABLE DC control and the OUTPUTADDER 'g' control, set the DC level at the input to the DECISION MAKER +25 mV (use the WIDEBAND TRUE RMS METER). Replace the inputs to the INPUTADDER.*

Measuring the BER

Everything is now set up for some serious measurements. It is assumed that:

• Both SEQUENCE GENERATORS are set for long sequences (both toggles of the on-board switch SW2 are DOWN).

• Line code NRZ-L has been patched (for this experiment) on the LINE-CODE ENCODER and LINE-CODE DECODER.

• Line code NRZ-L has been selected with SW1 on the DECISION MAKER board.

• All reset buttons have been pushed (in turn from input to output).

• Levels throughout the system have been set correctly (typically with SNR = 0 dB with max noise from the NOISE GENERATOR).

• DC threshold at the DECISION MAKER has been accounted for.

• Signal into the DECISION MAKER is ideally at the TIMS ANALOG REFERENCE LEVEL (but probably considerably lower with the model of Figure 4).

• The DECISION POINT of the DECISION MAKER has been set up, using an eye pattern (with 'moderate' noise present - say an SNR of 10 dB).

• The SEQUENCE GENERATOR at the receiver has been aligned with the incoming sequence (carried out with no noise present - a high SNR).

• Conditions for a known (reference) SNR are recorded.

• Channel bandwidth is recorded (eg, which filter of the BASEBAND CHANNEL FILTERS module is in use).

T6.5 *Measure BER according to the procedure in Task T4.7. Record the measurement, and the conditions under which it was made. Compare results with counts over short and long periods.*

T6.6 *decrease the noise level by one increment of the NOISE GENERATOR front panel attenuator. Go to the previous Task. Loop as many times as appropriate.*

T6.7 *plot BER versus SNR.*

Appendix

ERROR COUNTING UTILITIES module

A full description of this module is available in the TIMS Advanced Module User Manual. This should be essential reading before the module is used.

Before use it is necessary to check the settings of the on-board switches SW1 and SW2, and the jumper J1.

Briefly, the module consists of two sub-systems:

X-OR gate

This has two modes:

1. *pulse mode*: with a clock signal connected. Acts as a gated sub-system Somewhere near the middle of each clock pulse it makes an X-OR decision regarding the two TTL inputs. Its output is a TTL HI if they are different, otherwise a LO. In the present application it compares each bit of the regenerated received signal with a reference generator. Differences – which represent errors - are counted by the FREQUENCY COUNTER in COUNT mode.

2. *normal mode*: with no clock input

Gate timing pulse

This clocked sub-system, on receipt of a trigger pulse - manual or electronic -outputs a pulse of length (number of clock periods) determined by the front panel switch PULSE COUNT, the toggles of the on-board switch SW2, and jumper J1.

In this experiment the trigger pulse is initiated by the front panel TRIG push button.

The GATE output pulse (a LO, selected by toggle 2 of the on-board switch SW1) is used to activate the FREQUENCY COUNTER, in COUNT mode.

On-board settings for this experiment.

switch/jumper	*toggle*	*position*	*comments*
J1		NORM	
SW1	1 - TRIG	HI - to left	suits press button
SW1	2 - GATE	LO - to right	counter activated on LO
SW2	1	ON - to right	PULSE COUNT switch
SW2	2	ON - to right	settings times unity