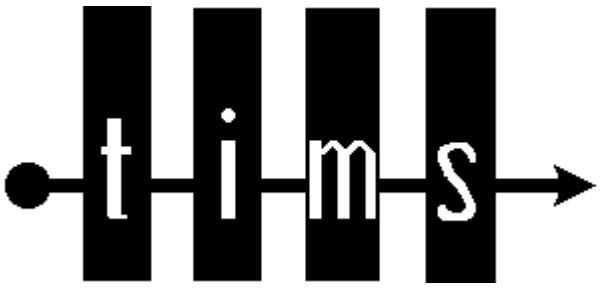
Communication Systems Modelling

with

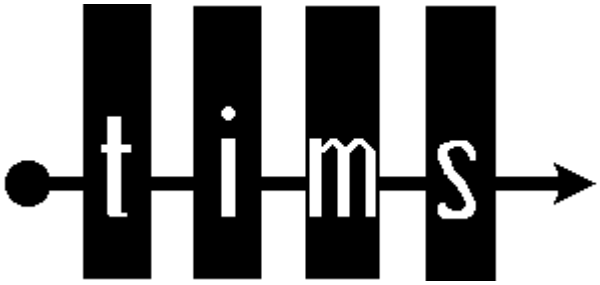


Volume D2
Further & Advanced Digital Experiments

Tim Hooper

Communication Systems Modelling

with



Volume D2
Further & Advanced Digital Experiments

Emona Instruments Pty Ltd
ABN 79 069-417-563
86 Parramatta Road
Camperdown NSW 2050
Sydney AUSTRALIA

WHAT IS TIMS ?

TIMS is a **T**elecommunications **I**nstructional **M**odelling **S**ystem. It models telecommunication systems.

Text books on telecommunications abound with block diagrams. These diagrams illustrate the subject being discussed by the author. Generally they are small sub-systems of a larger system. Their behaviour is described by the author with the help of mathematical equations, and with drawings or photographs of the signal waveforms expected to be present.

TIMS brings alive the block diagram of the text book with a working model, recreating the waveforms on an oscilloscope.

How can TIMS be expected to accommodate such a large number of models ?

There may be hundreds of block diagrams in a text book, but only a relatively few individual block *types*. These block diagrams achieve their individuality because of the many ways a relatively few element *types* can be connected in different *combinations*.

TIMS contains a collection of these block types, or *modules*, and there are very few block diagrams which it cannot model.

PURPOSE OF TIMS

TIMS can support courses in Telecommunications at all levels - from Technical Colleges through to graduate degree courses at Universities.

This text is directed towards using TIMS as support for a course given at any level of teaching.

Most early experiments are concerned with illustrating a small part of a larger system. Two or more of these sub-systems can be combined to build up a larger system.

The list of possible experiments is limitless. Each instructor will have his or her own favourite collection - some of them are sure to be found herein.

Naturally, for a full appreciation of the phenomena being investigated, there is no limit to the depth of mathematical analysis that can be undertaken. But most experiments can be performed successfully with

little or no mathematical support. It is up to the instructor to decide the level of understanding that is required.

EXPERIMENT AIMS

The digital experiments in this Volume build on those covered in Volume D1. It is advantageous to have completed as many of those as possible.

As before, the experiments have been written with the idea that each model examined could eventually become part of a larger telecommunications system, the aim of this large system being to transmit a *message* from input to output. The origin of this message, for the digital experiments in Volumes D1 and D2, is generally a pseudo random binary sequence. For the analog experiments, in Volumes A1 and A2, it would ultimately be speech. But for test and measurement purposes a sine wave, or perhaps two sinewaves (as in the two-tone test signal) are generally substituted.

The experiments are designed to be completed in about two hours, with say one hour of preparation prior to the laboratory session.

The four Volumes of ***Communication Systems Modelling with TIMS*** are:

A1 - Fundamental Analog Experiments

A2 - Further & Advanced Analog Experiments

D1 - Fundamental Digital Experiments

D2 - Further & Advanced Digital Experiments

Contents

BER MEASUREMENT IN THE NOISY CHANNEL

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.

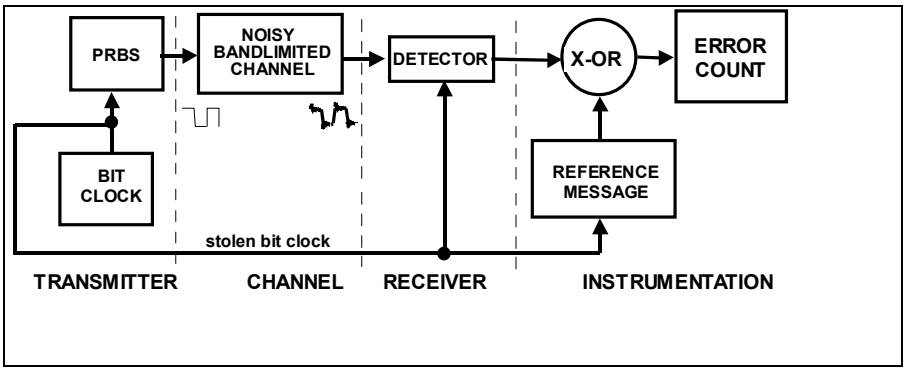


Figure 1: block diagram of system

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. This was introduced in the experiment entitled *Detection with the DECISION MAKER* in Volume D1.
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 modelling with TIMS.

So Figure 1 has been expanded into Figure 2 below.

The detector is the DECISION MAKER module, introduced in the experiment entitled *Detection with the DECISION MAKER*.

The LINE-CODE ENCODER and LINE-CODE DECODER modules were introduced in the experiment entitled *Line coding* in Volume D1.

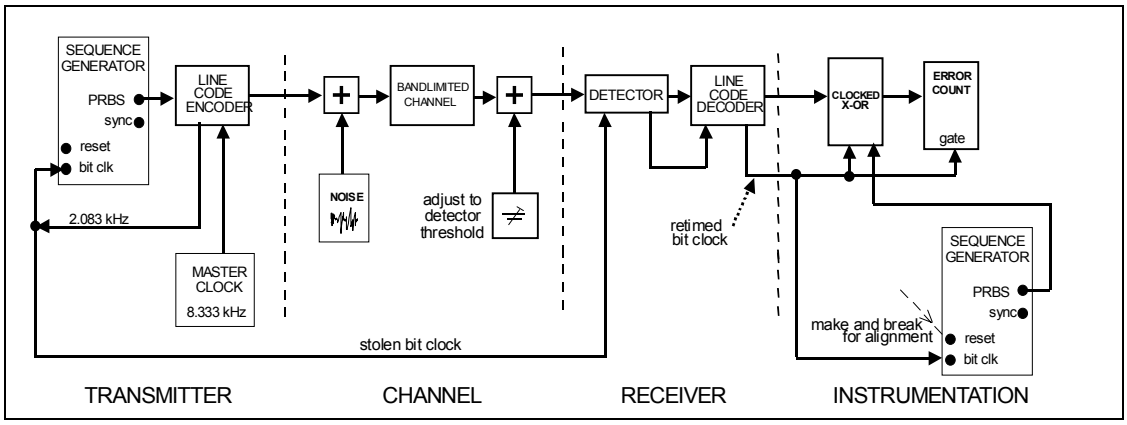


Figure 2: block diagram of system in more detail

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 E_n/N_o. For matched filter reception of bi-polar baseband signalling 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 E_n/N_o , but SNR. Figure 3 shows theoretical predictions, based on eqn(1) above.

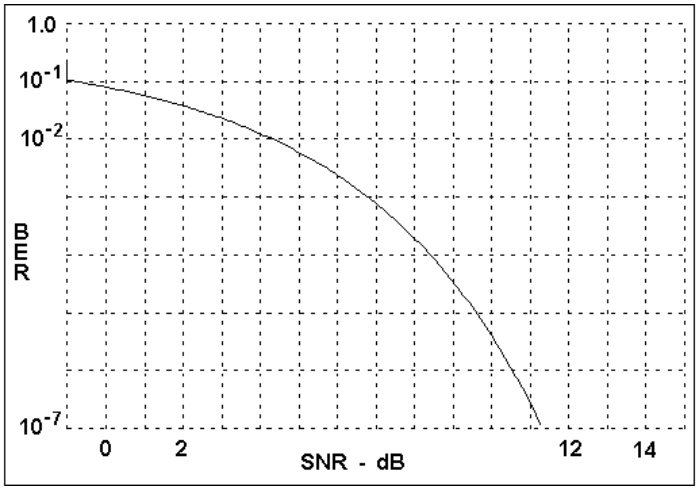


Figure 3 theoretical expectations - BER versus SNR (for bi-polar signalling)

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 modelled 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 was introduced in the experiment entitled *The noisy channel model* in Volume D1, which you should already have completed.

As a reminder, details of the macro CHANNEL MODEL module are reproduced in Figure 4 below.

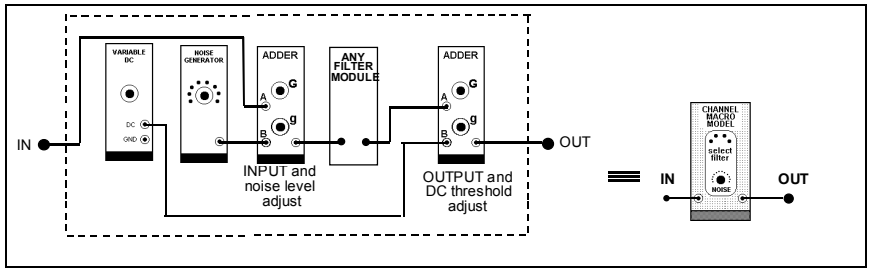


Figure 4: details of the macro CHANNEL MODEL module

Remember that, during testing, and afterwards, the oscilloscope triggering comes from:

- the SYNC output from the transmitter SEQUENCE GENERATOR for snapshots
- the bit clock for eye patterns.

the ERROR COUNTING UTILITIES module

This is the first time the pulse counting capabilities of the ERROR COUNTING UTILITIES module have been used. A complete description of the characteristics and behaviour of the module can be obtained from the *TIMS Advanced Modules User Manual*.

A condensed description of its function is given in the Chapter entitled *Digital utility sub-systems* in this Volume, under the two headings *Timed Pulse* (for the counting function) and *Exclusive-OR*.

modelling the transmission system

The system to be modelled 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.

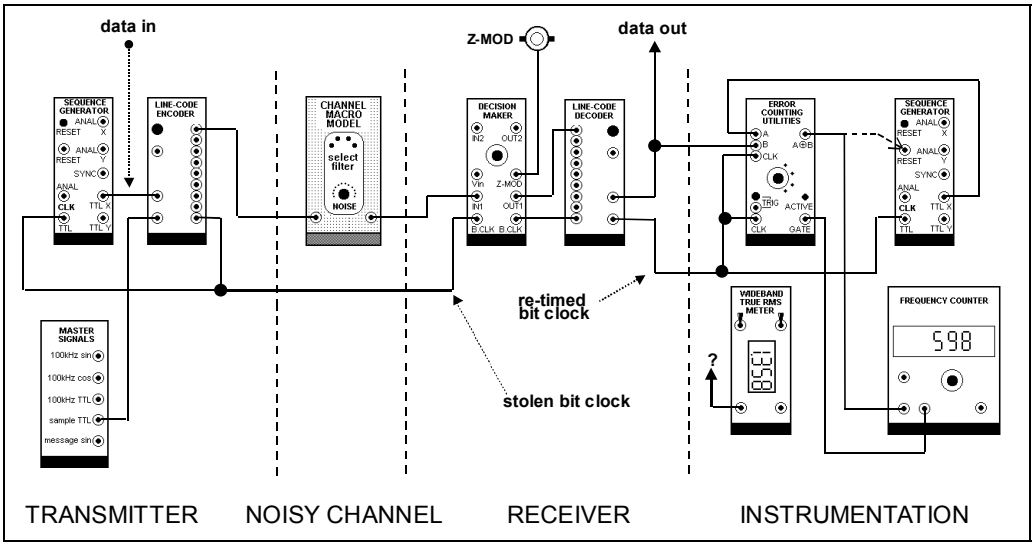


Figure 5: the TIMS model of Figure 2

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* INPUT *ADDER 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* OUTPUT *ADDER 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 INPUT ADDER of the channel, and a second inversion introduced by the OUTPUT ADDER.

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 *patch up the DECISION MAKER, including the Z-MOD output to the oscilloscope. It is assumed that the Z-MOD adjustments have been made on the circuit board to suit your oscilloscope [1].*

T3.3 *trigger the oscilloscope from the bit clock, and obtain an eye pattern at the channel output. Adjust the sampling instant, with the DECISION MAKER front panel control, to the centre of the eye. Remember that some fine adjustment of the intensity control of the oscilloscope will probably be necessary to easily identify the bright spot at the sampling instant.*

T3.4 *trigger the oscilloscope from the SYNC output of the transmitter SEQUENCE 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.*

T3.5 *refer to the DECISION MAKER in the* ***TIMS User Manual*** *for threshold level information. This varies according to the code in use. For the NRZ-L code the threshold is approximately 25 mV. Thus the input signal amplitude must either swamp any possible DC threshold, or, if small, must be adjusted to straddle it. There is provision in the model (the OUTPUT ADDER) for this; it will be checked in the next Section. For now confirm that the output waveform is centred approximately about zero volts.*

T3.6 *patch up the LINE-CODE DECODER, selecting the NRZ-L output.*

T3.7 *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.

[1] refer to the experiment entitled *Detection with the DECISION MAKER*. in Volume D1

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 close 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 patch it into the system, complete with the gate signal from the ERROR COUNTING UTILITIES module.*

T4.6 *switch the gate of the ERROR COUNTING UTILITIES, with the PULSE COUNT switch, to be active for 10^5 bit clock periods. Make a mental calculation to estimate how long that will be !*

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. But ·····

> ***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 shuts. 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 error counting with noise

preparation

T5.1 *increase the message sequence length of <u>both</u> SEQUENCE GENERATOR modules (both toggles of SW2 DOWN). See Tutorial Question Q2.*

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 INPUT ADDER.

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 maximising the gain through the INPUT ADDER, 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 INPUT ADDER, 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 INPUT ADDER.*

T5.7 *introduce some signal with the '**G**' control of the channel INPUT ADDER, 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 <u>not</u> disturb the INPUT ADDER gain settings from now on !*

T5.9 *check the signal level at the channel output. Use the '**G**' gain control of the OUTPUT ADDER 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 $10\log_{10}\left[\frac{S^2}{N^2}\right]dB$

and this will be 0 dB. However you may have your own reasons for selecting some other ratio, but it needs to result in many errors. From now on you can only *reduce* the noise, using the calibrated attenuator of the NOISE GENERATOR. This will *increase* the SNR, which will in turn *reduce* the error rate.

> ***warning****: if alignment is ever lost the noise must be removed before attempting re-alignment !*

T5.10 *set the SNR to, say, 10 dB, and set the decision instant with the aid of an eye pattern.*

6.0 DC threshold adjustment

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

Two methods are suggested.

1. after setting up as above, add a small DC to the signal from the channel. If the error count can be reduced then adjust for the smallest count.
2. set the DC output from the channel to +25 mV. This is the threshold level of the DECISION MAKER in NRZ-L mode. Recall the measurement made in this regard in the experiment entitled *Detection with the DECISION MAKER* in Volume D1.

See Tutorial Question Q3.

Now implement one or the other method of threshold adjustment.

method #1

T6.1 *set the PULSE COUNT on the DECISION MAKER to 10^5 and press the TRIG button. Adjust the noise level with the attenuator so that errors are accumulating at about 10 per second (watch the second last digit).*

T6.2 *rotate the VARIABLE DC level about 45^0 anti-clockwise. Advance the '**g**' control of the OUTPUT ADDER about 20^0. The error rate should increase.*

T6.3 *slowly reduce the DC offset voltage magnitude (rotate the VARIABLE DC control clockwise towards zero). The error rate should slowly reduce, then increase. Return to the lowest rate and stay there. This is an important adjustment. It takes some practice. At all times set the error rate (with the noise source attenuator) so it is about 10 errors per second. Concentrate on the second last, and then the last digit, as the minimum is approached.*

method #2

T6.4 *remove both inputs from the INPUT ADDER. Using both the VARIABLE DC control and the OUTPUT ADDER '**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 INPUT ADDER.*

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 button 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. Relate your results to expectations.*

role of the filter

The characteristics of the filter will influence the result. The theoretical results assume an 'ideal' filter. We do not have that.

conclusion

Future experiments will use this system configuration to measure BER under different conditions - for example, with the addition of error control coding, bit clock regeneration, and so on.

It is important, then, that you familiarize yourself with the setting up procedures of the basic system which was the subject of this experiment.

TUTORIAL QUESTIONS

Q1 *once sequence alignment is attained, the sliding window correlator is disabled. Explain why alignment is not lost even if the noise level is raised until the BER increases to unacceptably high levels.*

Q2 *why were you advised to use a long sequence when counting errors ?*

Q3 *explain the principle of what you were doing when adjusting the DC at the input to the DECISION MAKER.*

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.

Th 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 settings times unity
SW2	2	ON - to right	

BER INSTRUMENTATION MACRO MODULE

ADVANCED MODULES: *ERROR COUNTING UTILITIES, WIDEBAND TRUE RMS METER. Both the system being measured* ***and*** *this macro module require a SEQUENCE GENERATOR.*

introduction

Bit error rate (BER) measurement techniques were first introduced in the experiment entitled *BER in the noisy channel* in this Volume. That experiment used a macro CHANNEL MODEL module. This 'module' was defined earlier in the experiment entitled *The noisy channel model* in Volume D1.

In subsequent experiments this macro module is represented in patching diagrams as a single module, in order to save space.

Likewise, the BER instrumentation is required in many experiments, and it is convenient to represent it also as a single 'macro module' to save space, and repetition, in patching diagrams.

This Chapter is intended to serve as a convenient reference to the macro BER INSTRUMENTATION module.

This instrumentation has been devised for those experiments which use a pseudo random sequence from a SEQUENCE GENERATOR to provide the source message, and a second SEQUENCE GENERATOR in the instrumentation as a reference.

the BER instrumentation

principle

The instrumentation consists of the following elements:

1. a sequence generator identical to that used at the transmitter. It is clocked by the message bit clock. This locally supplied 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. This was introduced in the experiment entitled *Detection with the DECISION MAKER* in this Volume.

3. a means of measuring differences between the received sequence and the reference sequence (after alignment); ie, the errors. The error signal comes from the output of an X-OR gate (the same one used for the sliding window correlator). There is one pulse per error. The counter counts these pulses, over a period set by a gate, which may be left open for 10^n bit clock periods, where n = 3, 4, 5 or 6.
4. a method of measuring the signal-to-noise ratio (SNR) of the signal being examined. The WIDEBAND TRUE RMS METER is ideal for this purpose.

practice

The above ideas are shown modelled in Figure 1 below. It is assumed that the reference sequence generator is identical to, and set up similarly to, that at the transmitter.

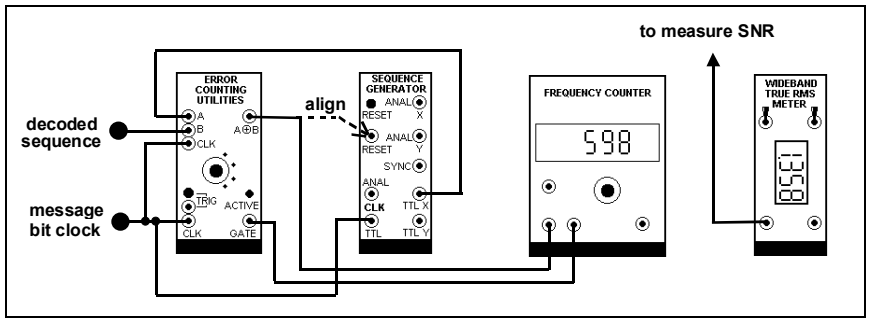


Figure 1: BER measurement instrumentation

In future experiments this model will be represented by the pseudo module shown in Figure 2 below.

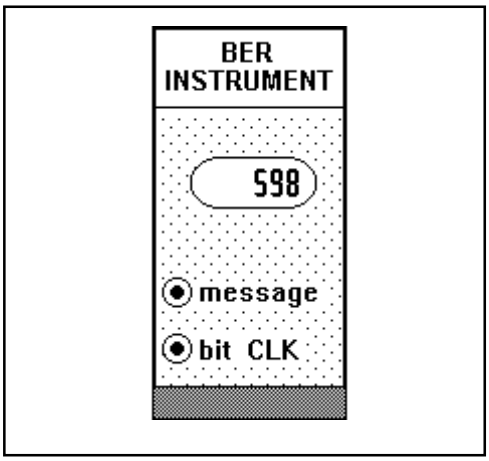


Figure 2: the BER INSTRUMENTATION macro module.

setting up

It is assumed that a transmission system is already in existence.

The procedure for setting up the BER INSTRUMENTATION is as follows:

1. patch up according to Figure 1
2. remove the NOISE from the channel
3. align the two sequences (momentarily connect the reset of the instrumentation SEQUENCE GENERATOR to the output of the X-OR gate of the ERROR COUNTING UTILITIES module).
4. press RESET of the COUNTER. No digits should be displaying.
5. press the TRIG button of the ERROR COUNTING UTILITIES module. The COUNTER should display '1'. This is the 'confidence count', *not* an error count. The COUNTER should remain at '1' for the duration of the PULSE COUNT, verified by the ACTIVE indicator being alight (it flickers during the last 10% of the count period).
6. replace the NOISE at a high level. The COUNTER should start counting bit errors (provided the ACTIVE indicator is alight). Reduce the NOISE and the BER should reduce.

Remember:

- always remove the noise before attempting to align the two sequences.
- the PULSE COUNT indicates the number of bit clock periods for which the GATE remains open (while the ACTIVE indicator is alight), and during which the COUNTER is activated for counting errors.
- the bit error count is the COUNTER display minus '1' (the 'confidence count').
- the ratio (COUNTER DISPLAY - 1) / (PULSE COUNT) is the BER.

theoretical predictions

See your Text book for theoretical predictions of bit error probability of various signals, typically expressed as a function of E_b/N_o, where:

- E_b is the energy per bit
- the only corruption is assumed to be additive white Gaussian noise (AWGN), where N_o is the average noise power per Hz.

From a practical point of view E_b/N_o is interpreted as the signal-to-noise ratio (SNR). This is a power ratio, and is typically expressed in decibels (dB).

The SNR is measured at the decision maker input.

Plots of bit error probability versus E_b/N_o will typically involve the function Q(x), where Q(x) is the complementary error function, given by:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} \exp\left(-\frac{u^2}{2}\right) du \qquad \text{..... 1}$$

There are many papers in the literature concerning the evaluation of this integral, including that given by P.O. Börjesson, C-E. Sundberg, "Simple approximations of the error function Q(x) for communications applications", IEEE Trans. Com, Vol. COM-27, No.3, March 1979, p639-643.

The above paper was pointed out to me by my colleague Bob Radzyner, who extracted the following approximation from it.

$$Q(x) = \frac{4f}{\left(3x + \sqrt{(v+8)}\right)} \qquad \text{..... 2}$$

$$\text{where } f = \frac{1}{\sqrt{2\pi\, e^v}} \qquad \text{..... 3}$$

$$\text{and where } v = x^2 \qquad \text{..... 4}$$

BIT CLOCK REGENERATION

BIT CLOCK REGENERATION

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

PREREQUISITES: *completion of at least some of the early experiments of Volume D1.*

ADVANCED MODULES: *BIT CLOCK REGEN, LINE-CODE ENCODER, ERROR COUNTING UTILITIES, INTEGRATE & DUMP. A BASEBAND CHANNEL FILTERS module is optional.*

TRUNKS: *optional 208 kHz sine. Refer Laboratory Manager.*

PREPARATION

synchronization

Receivers in a digital environment can require synchronization at 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.

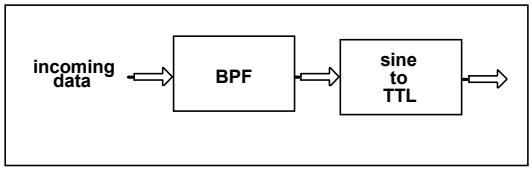


Figure 1: example of elementary open loop bit clock extraction

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.

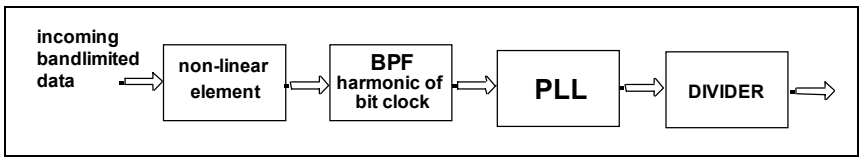


Figure 2: creation, and extraction, of a spectral component at bit clock frequency

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.

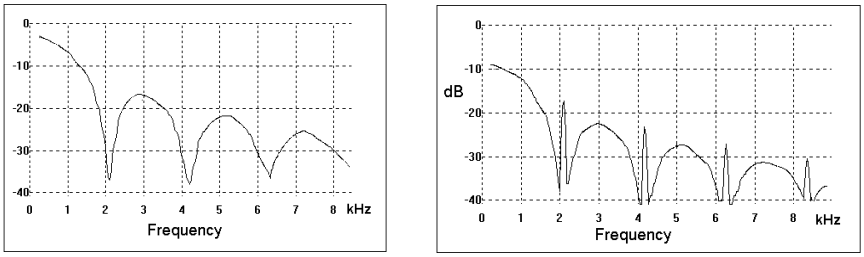


Figure 3: PRBS signal spectrum (a) before and (b) after bandlimiting and squaring

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.

jitter

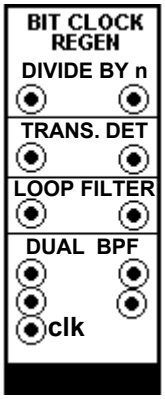
Bit clock recovery circuits can suffer from timing jitter.

Although the recovered clock is of the correct mean frequency, it can be undergoing either or both of linear and non-linear modulation.

The effects of linear modulation can be removed (or at least reduced) by amplitude limiting - by a comparator, for example.

The effects of non-linear modulation are not so easily overcome.

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 sub-systems. 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.

[1] recall the operation of the sliding window correlator for sequence alignment

procedure

Some of the signal formats available from the LINE-CODE ENCODER module can be used to test bit clock recovery schemes.

There are only two examples of bit clock recovery scheme given in the experiment to follow - one in detail, and the other in outline.

When completed, you are invited to investigate other methods of recovery in which you are interested. In preparation, here are some reminders of signals and systems which may be useful.

signal source

As already mentioned, the LINE-CODE ENCODER, driven by a SEQUENCE GENERATOR at 2.084 kHz, is a good source of bit streams having different characteristics. These serve as inputs to your bit clock regenerator, after passing through a noisy, bandlimited channel.

modules

The following modules will be found useful in the work to follow:

- BIT CLOCK REGEN module: not surprisingly, this module will be useful !
- UTILITIES module: contains a CLIPPER/COMPARATOR - a useful odd-order non-linear characteristic, and used for converting a sinewave to TTL format.
- a MULTIPLIER, as a SQUARER, provides an even-order non-linear characteristic.
- VCO: as part of a phase locked loop (PLL). There is a loop filter in the BIT CLOCK REGEN module, as well as a TTL divide-by-two sub-system.
- NOISE GENERATOR: a low SNR will put your regeneration system to the test.
- the INSTRUMENTATION MODEL macro module will check BER performance, although a simplified bit clock 'quality' arrangement is suggested.

bit clock component present ?

Before modelling a regeneration scheme, it might be a good idea to examine each of the line codes to check whether it already has a bit clock component present in its spectrum.

This can be done with the scheme of Figure 1.

bit clock component creation

For those spectra not already containing a spectral line at bit clock frequency try a scheme as illustrated in Figure 2.

BIT CLOCK REGEN BPF

Using an internal clock, the BPF in the BIT CLK REGEN module may be tuned to 2.048 kHz.

It may be tuned to other frequencies by the use of an external sinusoid (eg, a VCO).

For example, to tune the BPF to 4.167 kHz (twice 2.048 kHz) requires a clock at 50 times this frequency, namely 208.33 kHz. This frequency may be obtained from:

1) *a VCO*:

a) by setting the on-board switch SW2 to FSK
b) toggling the front panel switch to HI

c) leaving nothing connected to DATA IN (acceptable as a TTL LO)
d) adjusting RV7 (FSK1) for 208 kHz output (setting the frequency to 208 kHz by watching the frequency counter is acceptable. However it may be easier to connect a 4.167 kHz sine wave to the input of the BPF and then to tune the VCO for a maximum BPF output).

or

2) *TRUNKS*. This would be the preferred option.

bit clock quality

Rather than measure bit error rate over a noisy channel - which tests the complete system - it is instructive to measure just the 'quality' of the recovered bit clock. The term 'quality' is used loosely here. It refers to frequency and phase stability, jitter, and so on. See Tutorial Question Q5.

A method of measuring the quality consists of comparing the regenerated clock with the system clock, using the X-OR in the ERROR COUNTING UTILITIES module, as a performance indicator. A suggested arrangement is shown in block diagram form in Figure 4 below. The phase of the sinusoidal output from the BPF is made adjustable so it may be aligned with the reference bit clock. A digital VARIABLE DELAY is inserted in the gate to the X-OR to control the instant of comparison.

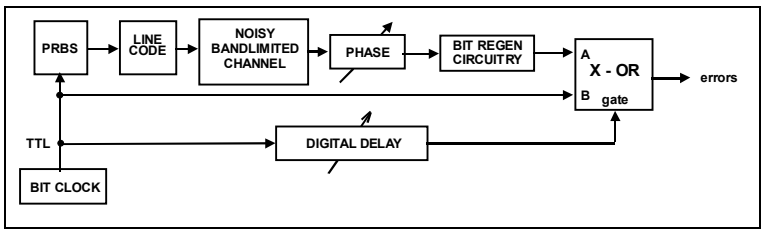


Figure 4: measurement of bit clock quality

By appropriate adjustment of the analog phase and the TTL delay, and with no noise, the arrangement can be set so as to register no errors.

Noise could then be added to the channel in order to make a more demanding test.

Reliable bit clock recovery should be possible for signal-to-noise ratios approaching 0 dB.

This system will be modelled in the experiment.

system performance

If the quality of the recovered bit clock is considered good, by the previous test, then the overall system performance can be measured by carrying out a bit error rate measurement over the noisy channel.

This is perhaps an unnecessary extension of the experiment, the aim of which was to introduce some basic methods of bit clock recovery, without going into great detail.

EXPERIMENT

The complete system to be modelled involves many modules. It will be patched up systematically. It is suggested that the modules be inserted into the TIMS frame in the order shown, starting at the extreme left hand side.

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.

bit clock recovery - method #1

T1 *acquire a BIT CLOCK REGEN module. Read about it in the* ***Advanced Modules User Manual****. Before plugging it in locate the 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 frequency) later on in the experiment.*

T2 *patch up the diagram of Figure 5, which is a model of the open loop regeneration scheme of Figure 1.*

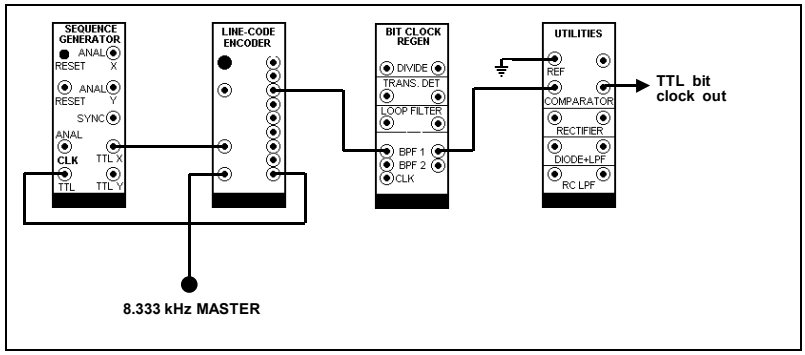


Figure 5: model of Figure 1

T3 *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. However, its amplitude will be varying with time. Is this due to amplitude modulation or phase modulation ? See Tutorial Question Q1.*

T4 *observe the output of the COMPARATOR on CH2-B. This is a TTL signal, of fixed amplitude, and mean frequency 2.083 kHz. Is its phase varying ?*

adding noise

The above procedures demonstrated carrier regeneration from a wideband, noise-free signal. Now pass the test signal through a noisy, bandlimited channel.

T5 *add a noisy, bandlimited channel [2] to the model, as in Figure 6 below. Use a TUNEABLE LPF as the bandlimiting filter, or channel #3 of a BASEBAND CHANNEL FILTERS module. Without noise, adjust the gains of the TUNEABLE LPF (bandwidth set to maximum) and each ADDER to unity. Include the PHASE SHIFTER; it will be required later. Confirm the regenerated carrier is still present at the output of the COMPARATOR.*

T6 *add noise. Estimate at what level of SNR the recovered bit clock might become unusable ? Explain how you made this estimate.*

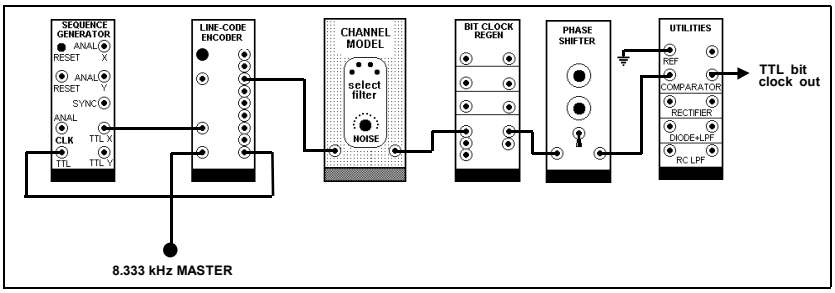


Figure 6: adding a noisy channel

bit clock 'quality'

Now add some instrumentation to measure the 'quality' of the recovered clock.

T7 *add the error counting facility shown in Figure 7 below. This is based on the scheme illustrated in Figure 4. As shown, the regenerated bit clock is patched to the 'A' input f the X-OR gate, and the reference (the system bit clock) into the 'B' input.*

[2] described in the experiment entitled *The noisy channel model* in Volume D1.

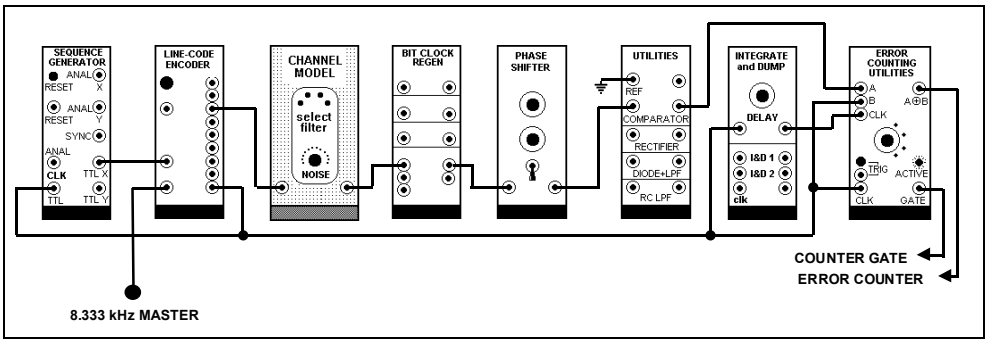


Figure 7: the system with instrumentation added

T8 *first check performance of the error counter with the system bit clock in both inputs 'A' and 'B'.*

T9 *when happy with the previous Task, remove the noise, and replace the stolen bit clock with the regenerated bit clock. Check the alignment of the X-OR inputs. Adjust the DIGITAL DELAY. Adjust the DIGITAL DELAY for no errors* [3]*. This* ***must*** *be possible ! For the record observe the timing of the gating pulse from the DIGITAL DELAY to the X-OR clk with respect to the X-OR inputs (and the range over which it may be moved for no errors to be recorded).*

Describe in your notes what you understand by the statement 'adjust the digital delay'.

T10 *now patch the regenerated bit clock into the 'A' input of the X-OR gate (not via the DIGITAL DELAY). Align the two inputs to the X-OR gate with the PHASE SHIFTER (on-board switch set LO). Adjust the DIGITAL DELAY for no errors.*

T11 *with no errors the recovered clock should be of acceptable quality. Now add noise, and report results.*

Record in your notes your opinion regarding the validity of the 'quality' measurements.

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.

[3] for details of range setting of the DIGITAL DELAY see the Appendix to this experiment.

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

The model is shown in Figure 8 below. It is complete with recovered bit clock 'quality' assessment instrumentation.

Detailed step-by-step Tasks are not provided.

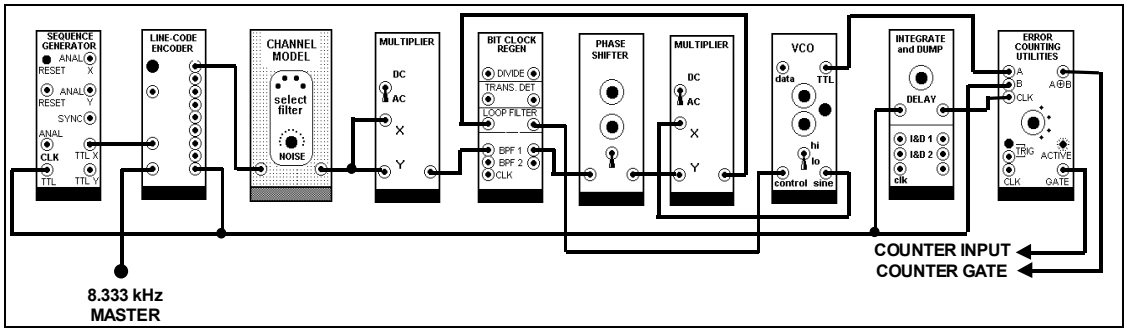


Figure 8: the TIMS model

You will note that the model contains 13 plugin modules (there are five within the CHANNEL MODEL macro module - but one, the VARIABLE DC, is a fixed module so does not require a free slot). These 13 cannot be accommodated within a single TIMS 301 system.

However, if you use a QUADRATURE UTILITIES module which contains two MULTIPIERS, then there is sufficient space in a single TIMS 301.

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.

The second MULTIPLIER, together with the VCO and LOOP FILTER in the BIT CLOCK REGEN module, implements a phase locked loop (PLL). You might query the need for this, since the output of the BPF is already a sinusoid at this frequency (the sinusoid could be converted to TTL, as required by the EXCLUSIVE-OR gate in the ERROR COUNTING UTILITIES, with the COMPARATOR in the UTILITIES module). Consider the merits of both systems, and try each as time permits.

bit error rate

A final check of the quality of any bit clock recovery scheme would consist of measuring the bit error rate of the overall system under different conditions.

TIMS can do that, following the procedures set out in the experiment entitled *BER measurement in the noisy channel* (in this Volume). It would call for a second TIMS 301, or a TIMS Junior, to accommodate the extra modules.

It would also go beyond the intended aim of the experiment, which was to introduce some elementary schemes of bit clock recovery.

TUTORIAL QUESTIONS

Q1 *a bit clock, recovered as a sine wave with varying amplitude, may or may not have uniform zero crossings. Give examples of the two cases. Which one gives rise to timing jitter ?*

Q2 *how would the presence of timing jitter (in your extracted clock) show up on the oscilloscope ?*

Q3 *can you distinguish, using only the oscilloscope, the difference between amplitude jitter and phase jitter on a regenerated clock bit ?*

Q4 *what factors might influence the choice between an open loop and a closed loop bit clock regeneration scheme ?*

Q5 *describe the various imperfections from which a recovered bit clock can suffer.*

APPENDIX

digital delay

The DIGITAL DELAY sub-system is built into the INTEGRATE & DUMP module. It is described in the *Advanced Modules User Manual*, as well as in the Chapter entitled *Digital utility sub-systems* of this Volume.

The delay is adjustable by a front panel control DELAY, in conjunction with a toggle switch SW3 mounted on the circuit board. The delays to be expected are shown in the table below.

SW3-upper toggle	*SW3-lower toggle*	*delay range from front panel, using DELAY*
RIGHT	RIGHT	10 µsec - 100 µsec
RIGHT	LEFT	60 µsec - 500 µsec
LEFT	RIGHT	100 µsec - 1 msec
LEFT	LEFT	150 µsec - 1.500 msec

on-board switch SW3 settings

The bit clock in the present experiment is 2 kHz, so the period is 500 µs.

divide-by-2

There is a TTL divide-by 1, 2, 4, or 8 in the BIT CLOCK REGEN module. The on-board switch setting s are shown in the Table below.

SW2-A (left)	*SW2-B (right)*	*divide by*
DOWN	DOWN	8
DOWN	UP	4
UP	DOWN	2
UP	UP	-1

on-board switch selectable division ratios

CARRIER ACQUISITION

CARRIER ACQUISITION

ACHIEVEMENTS: *carrier recovery from a modulated signal, using a phase locked loop.*

PREREQUISITES: *none*

ADVANCED MODULES: *BIT CLOCK REGEN*

PREPARATION

In a commercial application carrier acquisition (recovery, regeneration) from a (digitally) modulated signal is always required.

In a laboratory situation it has been seen that the use of a stolen carrier is preferred, to sharpen focus on other aspects of the experiment.

In this experiment the main focus is on carrier acquisition.

There are two cases to be examined - those modulated signals which already contain a component at carrier frequency, and those which don`t !

The latter is far more likely to be the case in commercial practice. Both types of signals are present at TRUNKS.

The modulated signals could have been derived from any of the baseband signals already studied, and then have been translated (modulated) to a higher (carrier) frequency.

The scheme outlined in Figure 1 will be modelled.

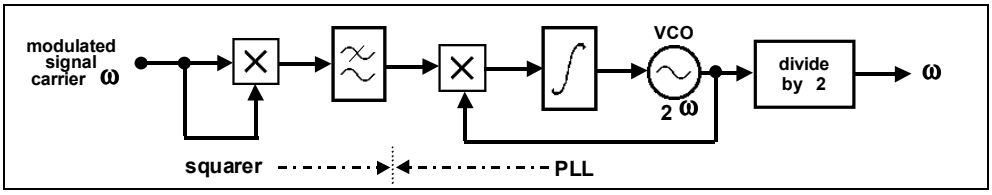


Figure 1: carrier regeneration from a modulated signal

Should there be a carrier component present in the received signal then the SQUARER, and DIVIDE-BY-2, can be omitted. The VCO would be then tuned to ω.

EXPERIMENT

A model of the scheme of Figure 1 is shown in Figure 2.

Observe that the block labelled as DIVIDE-BY-2 in Figure 2 will be a digital (TTL) sub-system, whereas the MULTIPLIER of the VCO requires an analog (sinusoidal) signal. This is easily accommodated by the TIMS VCO since it has both a TTL and an analog output.

The filter in Figure 1 following the SQUARER is perhaps not essential in many cases. It is included for completeness in the block diagram. In this experiment it can safely be omitted. See Tutorial Question Q3.

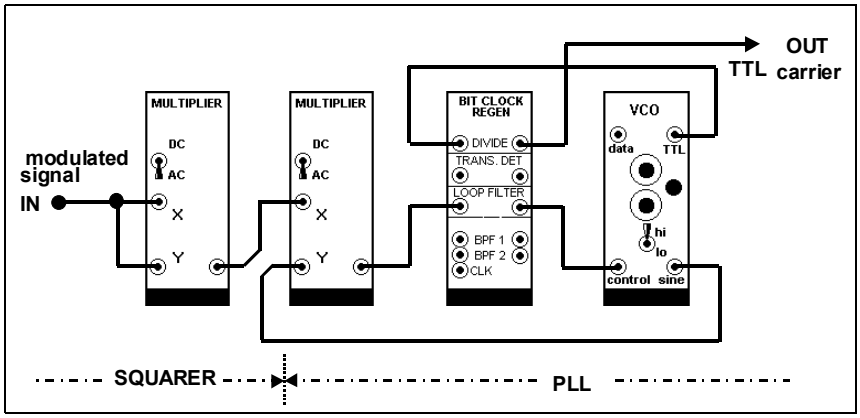


Figure 2: TIMS model of Figure 1

T1 *patch up the model of Figure 2 without the SQUARER. Select the modulated signals appearing at TRUNKS on the 100 kHz carrier.*

T2 *use the oscilloscope to view both the incoming signal and the sinusoidal output of the VCO. Trigger to the latter.*

T3 *with the gain of the VCO set fully anti-clockwise (zero loop gain - no negative feedback) tune the VCO to near 100 kHz. Watch the two oscilloscope traces. See if you can judge when the VCO is near the carrier frequency.*

T4 *when you think you have tuned the VCO close to the incoming carrier then introduce some negative feedback. Watch for indications of phase lock. If and when it occurs report the frequency of the recovered carrier.*

T5 *in your notes describe the technique you have adopted for obtaining and confirming phase lock with the PLL.*

T6 *is your recovered carrier free of linear or non-linear modulation ? What technique did you use to check this ?*

T7 *introduce the SQUARER to the model, and repeat the previous Tasks, this time working with the TRUNKS signal based on a 50 kHz carrier.*

TUTORIAL QUESTIONS

Q1 *how would the scheme illustrated in Figure 1 be modified if the received signal already had a spectral component at carrier frequency ?*

Q2 *it is essential that the MULTIPLIER following the filter of the SQUARER be AC coupled. Why is this ?*

Q3 *what is the purpose of the filter following the SQUARER in Figure 1 ?*

DPSK - CARRIER ACQUISITION AND BER

DPSK - CARRIER ACQUISITION AND BER

ACHIEVEMENTS: *reception and demodulation of a differential phase shift keyed (DPSK) signal, with carrier and bit clock recovery and bit error rate (BER) measurement.*

PREREQUISITES: *completion of the experiment entitled* ***BER measurement in the noisy channel*** *(this Volume) is essential; it would be an advantage to have completed the experiments entitled* ***Carrier acquisition*** *(this Volume) and* ***BPSK - binary phase shift keying*** *(Volume D1).*

ADVANCED MODULES: *NOISE GENERATOR, LINE-CODE DECODER, DECISION MAKER, ERROR COUNTING UTILITIES, BIT CLOCK REGEN, TRUE RMS WIDEBAND METER, DIGITAL UTILITIES.*

EXTRA MODULES: *a total of three MULTIPLIER modules.*

PREPARATION

BPSK

It is essential that you are familiar with setting up a bandlimited noisy channel, and measuring bit error rates (BER) over it. Thus completion of the experiment entitled *BER measurement in the noisy channel* is a prerequisite to this experiment.

It would be helpful, but not essential, if you have completed the experiment entitled *BPSK - binary phase shift keying*, of which the present experiment is an extension.

DPSK

A disadvantage of BPSK is that the receiver requires a knowledge of the frequency *and* phase of the carrier of the incoming signal.

As for BPSK, DPSK requires a local carrier for successful synchronous demodulation. But the phase of this carrier need not be known. It is the *differential coding* at the transmitter that makes this unnecessary.

experiment outline

The experiment is built around the principles investigated thoroughly in the experiment entitled *BER and the noisy channel*, so only an outline of procedures is given below.

A block diagram of the system to be examined is shown in Figure 1.

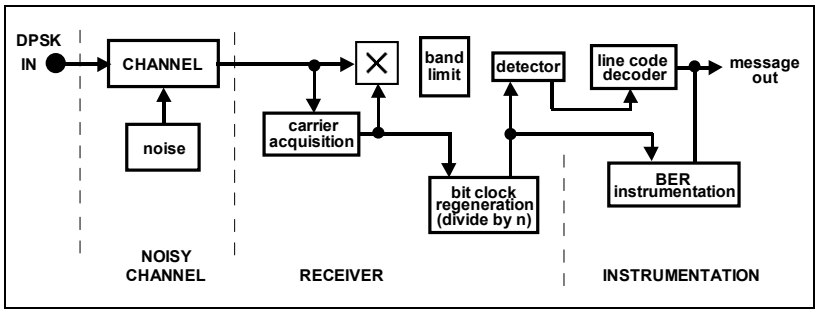


Figure 1: the DPSK receiving system

the transmitter

You will not be concerned with modelling the transmitter. The DPSK signal will come to you via TRUNKS. It will already be bandlimited.

It will be based on a carrier of *exactly* 50 kHz.

The message will be supplied at the transmitter by a SEQUENCE GENERATOR of the type you will have, set to a long sequence. It will be clocked at *exactly* 1/24 of the carrier frequency.

You will be responsible for demodulation and message recovery, both by stolen carrier (from TRUNKS) and by carrier acquisition circuitry.

carrier acquisition

With the data rate a sub-multiple of the carrier frequency then carrier acquisition circuitry is sufficient to recover both the carrier *and* the bit clock.

The method of carrier acquisition to be investigated in this experiment involves a squaring operation, followed by a phase locked loop. It is shown in block diagram form in Figure 2 below. Methods of carrier acquisition (including this one) were examined in the experiment entitled *Carrier acquisition* (in this Volume).

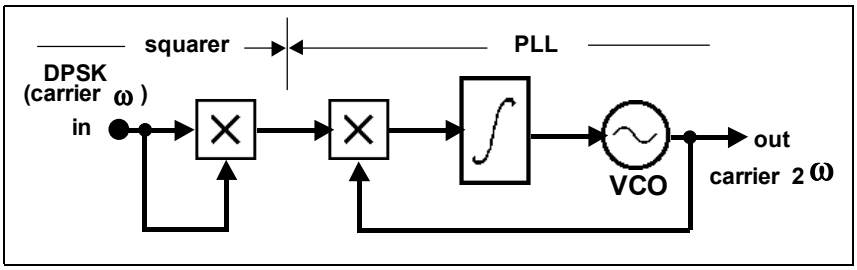


Figure 2: carrier and bit clock acquisition

In the scheme of Figure 2 the squaring operation generates a component at twice the carrier frequency. This is not of constant amplitude. It is smoothed by a phase locked loop, which acts as a narrow band filter.

Digital division-by-two will recover a TTL signal at carrier frequency, and a further division-by-twenty-four the 2.083 kHz clock for the DECISION MAKER.

channel

The channel is the (non-bandlimited) TRUNKS system, followed by an ADDER, which serves as an injection point for the system noise. Noise bandlimiting will occur at baseband. See Tutorial Question Q1.

theoretical predictions

Bit error probability (P_B) is a function of E_n/N_o. For synchronous demodulation of DPSK it has been shown that:

$$P_B = 2Q\left(\sqrt{\frac{2E_b}{N_o}}\right)\left[1-Q\left(\sqrt{\frac{2E_b}{N_o}}\right)\right] \quad \text{..... 1}$$

The symbols in eqn.(1) are defined in the Chapter entitled *BER instrumentation macro module* (in this Volume).

You will measure not P_B, but BER; and not E_n/N_o , but SNR. Figure 3 shows theoretical predictions, based on eqn(1) above.

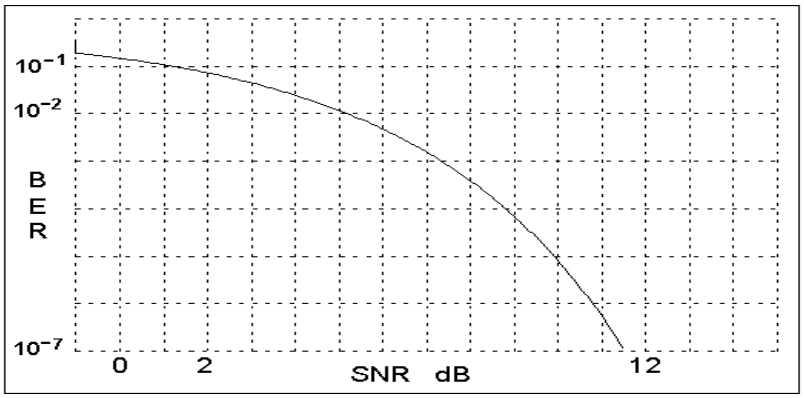


Figure 3: performance prediction - BER v. SNR
(DPSK, coherent detection)

EXPERIMENT

It is expected that you will not be attempting this experiment unless you are an experienced TIMS user. You will have completed the introductory digital experiments, and be familiar with the BER INSTRUMENTATION macro module. It should not be necessary to receive detailed setting up instructions.

This is a big system, requiring more than 12 slots for its modelling. You should plan ahead and decide how to distribute the modules of the receiver, instrumentation, and carrier acquisition models.

receiver

You will be modelling the receiver shown in block diagram form in Figure 1 above, and modelled in Figure 4 below.

T1 *before plugging in the DECISION MAKER set the on-board switch SW1 to accept differential encoding (NRZ-M), and SW2 to 'INT' (manual decision point adjustment).*

T2 *before plugging in the PHASE SHIFTER set the on-board switch to HI.*

T3 *patch up the receiver. Initially steal the 50 kHz carrier from TRUNKS and the bit clock (2.083 kHz) from the MASTER SIGNALS module. Set the bandwidth the same as that at the transmitter (or wider ?).*

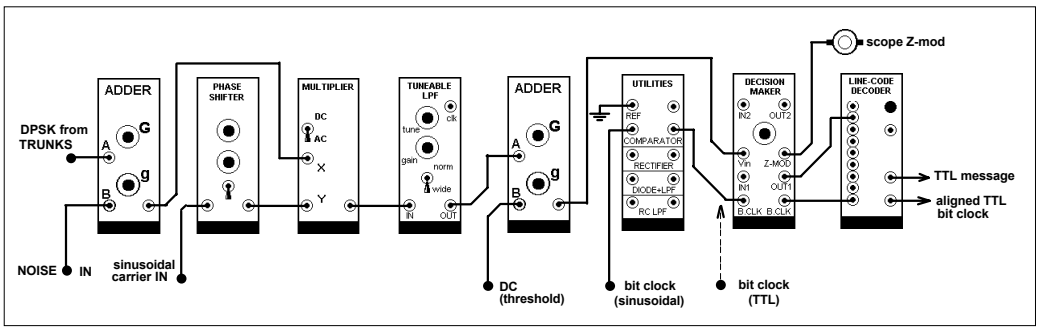


Figure 4: the receiver

T4 *set the receiver carrier phase for maximum input to the DECISION MAKER. Then use the channel gain to set this level to about the TIMS ANALOG REFERENCE LEVEL.*

T5 *set up the oscilloscope for an eye pattern. Set the decision instant to the appropriate part of the eye.*

T6 *confirm the received sequence is a (delayed) copy of the sent message. Confirm the behaviour of differential encoding.*

BER instrumentation

Bit error rate measurements will be made with the model described in the Chapter entitled *BER instrumentation macro module* (in this volume). This is reproduced in Figure 5 below.

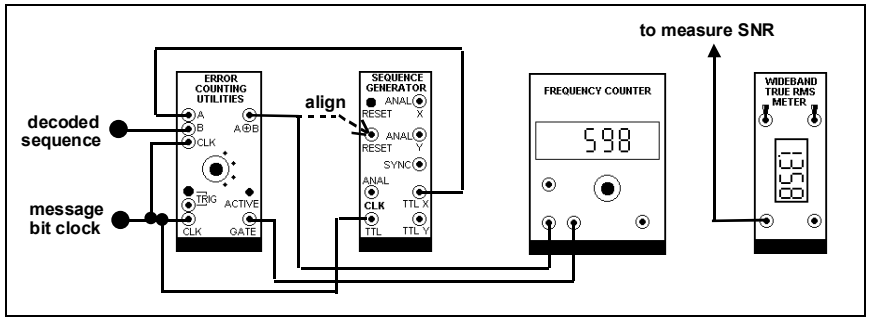


Figure 5: BER measurement instrumentation

T7 *set up the instrumentation. Align the received and reference sequences. With no added noise confirm that there are no errors.*

T8 *add noise. Confirm the error rate worsens as the SNR is reduced.*

T9 *prepare for some serious quantitative BER measurements.*

a) match the signal to the input threshold of the DECISION MAKER (about 25 mV).

b) add noise into the channel. Set up for a DECISION MAKER input SNR of 0 dB, and an absolute level of the TIMS ANALOG REFERENCE LEVEL.

BER measurement - stolen carrier

T10 *using a stolen carrier and bit clock, make some quantitative measurements over a range of SNR, and confirm that BER matches expectations.*

When satisfied that the system is behaving satisfactorily it is time to replace the stolen carrier with one acquired from the received signal.

carrier acquisition

A model of the carrier acquisition scheme shown in block diagram form in Figure 2 is modelled in Figure 6 below.

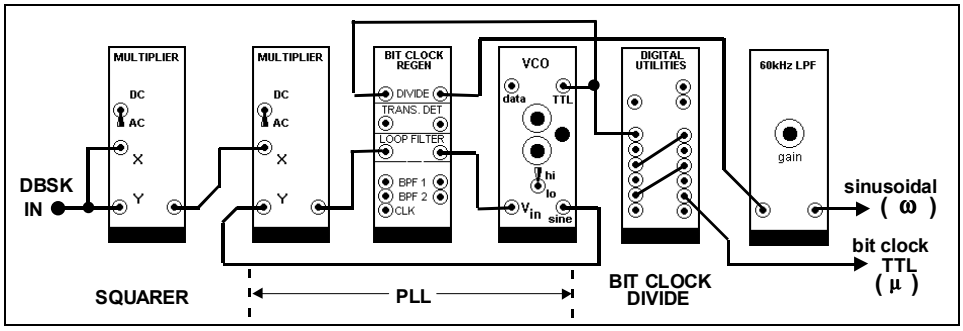


Figure 6: carrier acquisition model of Figure 2

Note that both the MULTIPLIER modules are AC coupled. There should be no component at DC at the input to the first, so AC coupling is merely a precaution against DC offsets. But the output of the squaring process will produce a large DC component, sufficient to overload the second MULTIPLIER, if nothing else. So it should be blocked.

The 100 kHz TTL output from the VCO is divided-by-two with the sub-system in the BIT CLOCK REGEN module (set the on-board switch SW2 with the left toggle UP and the right toggle DOWN). It is then filtered to a sine wave.

There is a TTL signal into an analog module (60 kHz LPF). Whilst this is usually not allowed (in the interests of linearity) here is one of those cases where it is acceptable ! Even if the input stage (of the filter) is overloaded the next filter stage may not be. Provided the output is a sinusoid (have a look) this is acceptable. After all, this is a filter, so it probably will not pass the distortion components anyway. But see Tutorial Question Q2.

bit clock recovery

Division-by-24 is required to derive the 2.083 kHz bit clock from the acquired 50 kHz carrier. This is available in a DIGITAL UTILITIES module.

T11 *patch up the carrier acquisition model. Set it up under no-noise conditions. Confirm it is operating as expected.*

BER measurement - acquired carrier

T12 *have the system measuring BER with high SNR. Check the carrier amplitude and phase into the receiver MULTIPLIER. Retain the stolen bit clock. Prepare the acquired carrier to have the same amplitude and phase, then use it instead of the stolen carrier. With high SNR there should be no change to measured BER.*

T13 *decrease the SNR and observe the deterioration of the BER. Not only is poor SNR to the DECISION MAKER causing errors, but the quality of the recovered carrier will have deteriorated - look for jitter.*

T14 *return to conditions of the penultimate Task (high SNR). Change over to the acquired bit clock. It will be necessary to check the alignment of the decision instant using an eye pattern as before.*

T15 *measure BER with a high SNR and compare with previous results. Reduce SNR - observe further deterioration of the system BER compared with the stolen carrier condition.*

TUTORIAL QUESTIONS

Q1 *noise usually enters the system in the channel. This is at carrier frequency. In the experiment the noise was indeed added into the channel, but it was not bandlimited until it reached baseband. Is this a 'legitimate' experimental technique ? What about the 'image response' of the product demodulator - would this cause a problem ?*

Q2 *suppose the recovered carrier was not a pure sinewave, because of overload of the filter. What would be some of the consequences ?*

PCM-TDM

PCM-TDM

ACHIEVEMENTS: *creation of a time division multiplexed pulse code modulated - PCM-TDM - signal by interlacing two PCM signals. De-multiplexing of same.*

PREREQUISITES: *completion of the experiments entitled* ***PCM encoding*** *and* ***PCM decoding*** *of Volume D1.*

ADVANCED MODULES: *two PCM ENCODER and one PCM DECODER (version 2 preferable). A second PCM DECODER is optional.*

PREPARATION

In the experiment entitled *PCM encoding* the PCM signal was generated as a binary data stream.

The sampling rate was one eighth of the bit clock rate.

Samples were coded into binary digital words, and placed into frames of eight slots, each slot being of length equal to a bit clock period.

Each frame contained a coded version of a 'flat top' sample of an analog signal (obtained with a sample-and-hold operation), together with a frame synchronization bit.

If the contents of every alternate frame were removed from the serial data (leaving eight 'empty' slots), then it would appear that the sampling rate had been halved. A consequence of this would be that the allowable bandwidth of the signal to be sampled would have been halved.

The message could still be decoded if each alternate frame could be identified.

Thus the empty spaces in the data stream could be filled with frames derived by sampling another message. These would not interfere with the frames of the first message. Thus two messages could be contained in the one data stream.

This is a time division multiplexed pulse coded modulated (PCM TDM) signal.

TIMS PCM TDM format

In a PCM TDM system there are several parameters to consider, including:

a) number of message channels 'n'

b) bandwidth of each message channel B_m

c) message sampling rate

d) bit rate of the PCM TDM signal

In what might be called a 'conventional' system B_m (and thus the message sampling rate) would be fixed, and independent of 'n'. So as the number of channels increased so would the bit rate of the PCM TDM signal itself. And so the bandwidth of the transmission channel would need to increase with 'n'. Consider these inter-relationships carefully.

In the TIMS PCM TDM format the opposite approach has been taken. The bit rate of the PCM TDM signal has been kept fixed, independent of 'n' (although 'n' is fixed at n = 2). So the transmission channel bandwidth can remain fixed, and independent of 'n'. Thus it was necessary to *halve* the message sampling rate when 'n' increased from 1 to 2.

TIMS PCM TDM

A PCM TDM signal can be created with two PCM ENCODER modules.

Each is driven by a common clock. One is nominated as the MASTER. By connecting its MASTER output to the SLAVE of the second, the second module becomes a SLAVE.

Their outputs *can be patched together*.

This is *not* a common practice with TIMS modules, but it is allowable in this case (the outputs employ *open collector* circuitry*)*.

Interconnection in this manner automatically (by internal logic) removes every alternate frame from each PCM signal in such a manner that the two outputs can be added to make a TDM signal as described above.

EXPERIMENT

The experiment will begin by patching up two *independent* PCM signals.

When these have been examined they will *then* be combined to make a TDM signal.

However, they will share a common clock

independent channels

The model will be that of Figure 1 below.

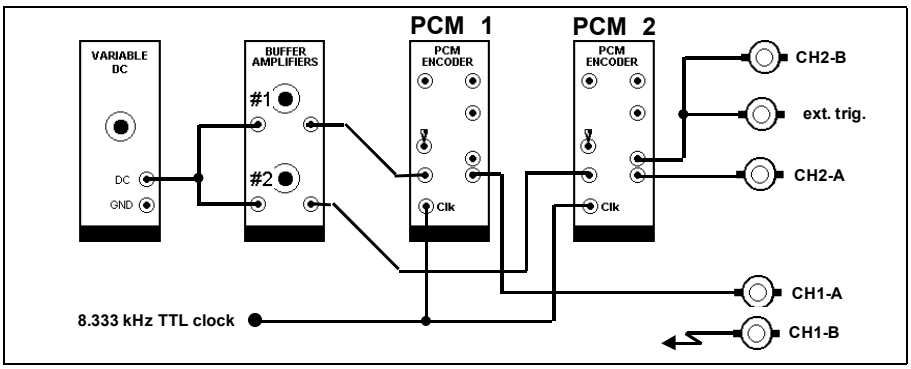


Figure 1: two independent PCM generators

T1 *patch up the model of Figure 1. Initially set both encoders to 4-bit linear (although you may prefer to change this later on).*

T2 *set the VARIABLE DC output to one end of its range. Reduce the gains of both BUFFER AMPLIFIERS to zero.*

T3 *with the oscilloscope triggered to the FS signal, and displaying it on CH2-B, set the sweep speed to display (say) two or three frames across the screen. Line up the FS signal with the graticule so that the positions of the 8 slots of each frame can be identified. Remember the FS signal marks the <u>end</u> of a frame.*

T4 *view CH1-A and CH2-A (the two channels are identified in Figure 1 as 'PCM 1' and 'PCM 2'). Set each channel to a different pattern, using the two BUFFER amplifiers.*

T5 *identify the alternate '0' and '1' pattern in each output in the LSB position.*

they have independently adjustable messages. These are shown here as DC, to ensure stable oscilloscope displays. Later periodic messages will be used.

multiplexing

Imagine what will happen to the displays on CH1-A and CH2-A when:

a) the MASTER/SLAVE relationship is invoked

b) the two PCM DATA outputs are joined to make a common output (as already mentioned, this is *not* normal TIMS practice).

Ostensibly this will make a two-channel TDM signal, with alternate frames being those of channel '1' and channel '2'.

Examine this now. First invoke the MASTER/SLAVE relationship:

T6 *observe the PCM output from PCM 1 as MASTER, and PCM 2 as SLAVE, while making and breaking a patch between the MASTER and SLAVE sockets. Note how alternate frames of each channel go HI.*

T7 *make a permanent connection between MASTER and SLAVE.*

Imagine what will now happen when the two 'common collector' outputs of PCM 1 and PCM 2 are patched together.

T8 *patch together the two PCM DATA outputs and check your expectations.*

T9 *check what has happened to the alternating '0' and '1' embedded frame synchronization bits which were, before combination of the two channels, at the end of each frame.*

T10 *show that the frame synchronization bit is a '1' for the MASTER channel, and a '0' for the SLAVE.*

Currently both messages are DC. Change one to AC. Use an AUDIO OSCILLATOR, or one of the built-in periodic messages. Remember there is no in-built message bandlimiting. The sampling rate is now half what it was before the encoders were slaved to each other, so choose your frequency wisely.

T11 *change one of the messages to a periodic waveform. Adjust its amplitude to about 2 volt peak. Explain what you see.*

demultiplexing

When you have convinced yourself that you have indeed made a two-channel TDM signal it is time to patch up a de-multiplexer and endeavour to separate the two messages. If you wish to be very cautious you could first use only one PCM ENCODER, and one PCM DECODER, and revise your understanding of the operation of the two modules.

The Tasks below take up the procedure assuming you are ready for TDM.

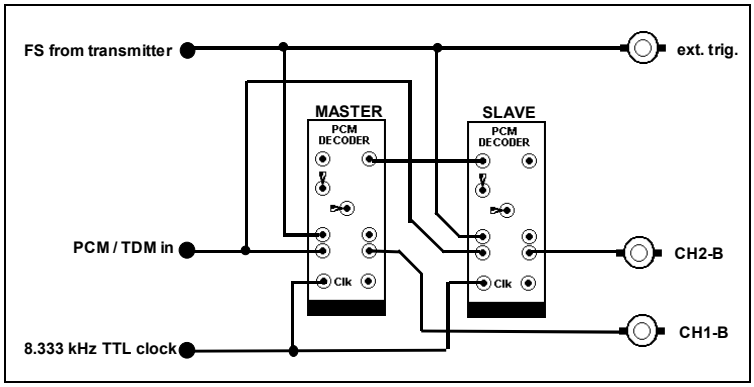


Figure 2: PCM TDM demodulator patching

T12 *patch up the demodulator as shown in Figure 2. Note that:*

a) each module receives the same clock, 'stolen' from the transmitter

b) each module receives an external FS signal (the embedded frame synchronization circuitry is disabled).

c) the coding schemes selected for each channel match those at the transmitter (where they can be different).

T13 *two outputs are available from each PCM DECODER - the quantized samples, and the reconstructed message from the built-in LPF (version 2 modules). Choose the reconstructed outputs. Confirm the two messages have been recovered - one is DC, and the other AC - and appear at the correct outputs.*

T14 *as patched in Figure 2 the frame synchronization signal 'FS' has been 'stolen' from the transmitter. Switch the FS SELECT toggle on either or both PCM DECODER modules to EMBED, and show synchronization is maintained.*

T15 *confirm that the coding schemes of the two channels are independent (eg, use 4-bit in one and 7-bit in the other).*

T16 *suppose only one PCM DECODER module was available, yet a 2-channel TDM signal is being received. From your knowledge of the operation of these modules, what would you expect to see at its output. Consider carefully before answering this question experimentally.*

TUTORIAL QUESTIONS

Q1 *when displaying the PCM TDM signal, it can happen, as the oscilloscope sweep speed is changed (either in fixed steps, or continuously), that sometimes the* MASTER *channel displays first, and sometimes second. How can it be made to appear first on all occasions ?*

BLOCK CODING & DECODING

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*** *in Volume D1.*

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.

[1] instead of being distributed over a number of blocks, as, for example, in a convolutional code.

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* in volume D1.

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.

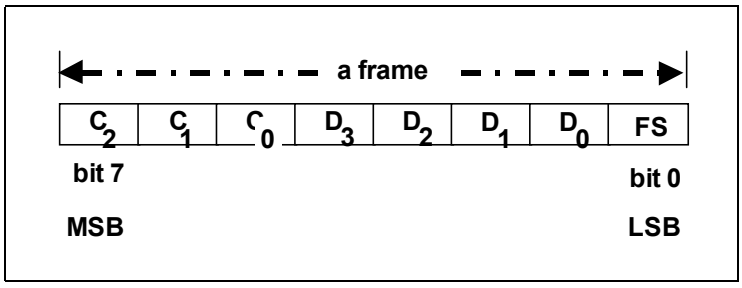


Figure 1: a data frame of eight slots, one per clock period

The message bits are shown as D_3, D_2, D_1, and D_0, where D_3 is the most significant bit of the message.

The frame synchronization bit is shown as FS.

The slots marked C_2, C_1, and C_0 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.

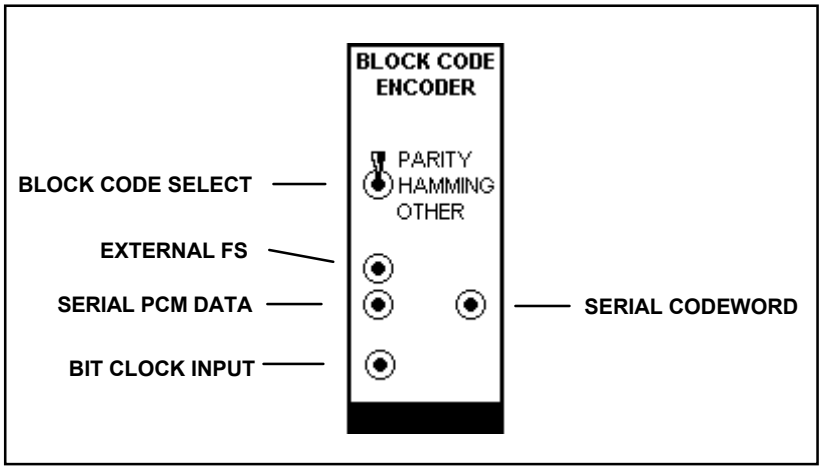


Figure 2: front panel layout - ENCODER

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.

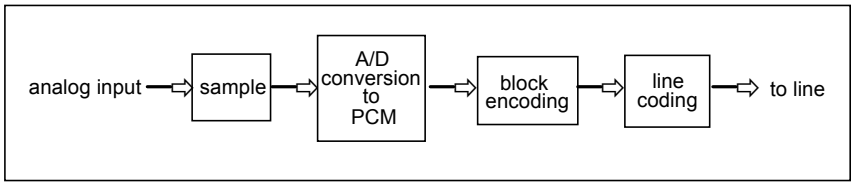


Figure 3: disposition of the block encoder

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.

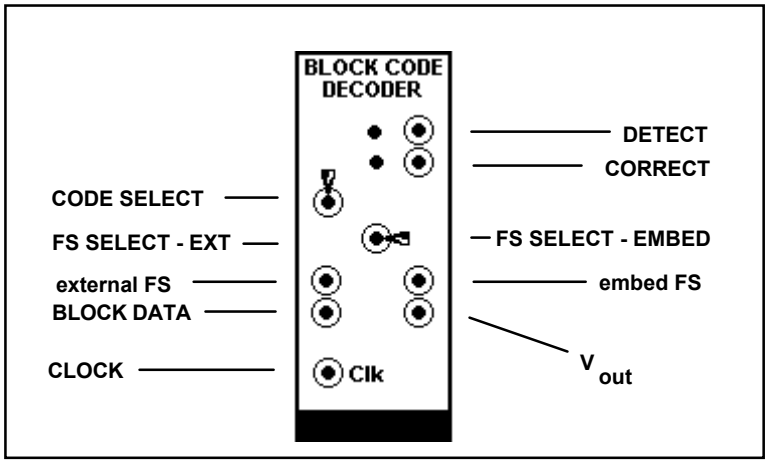


Figure 4: front panel layout - DECODER

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. It is a necessary preliminary to the experiment entitled *Block coding and coding gain* of this Volume.

encoding

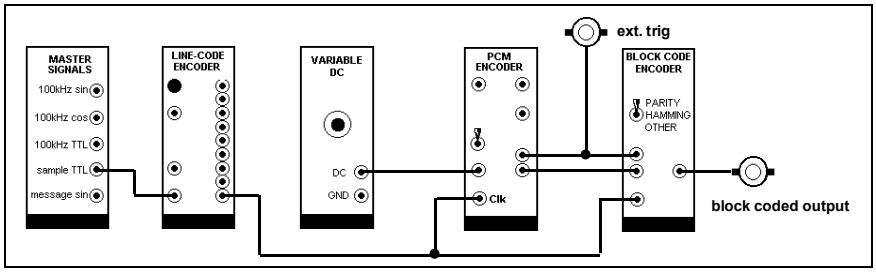


Figure 5: block code encoding

The BLOCK CODE ENCODER requires a TTL clock near 2 kHz. The *TIMS Advanced Modules User Manual* says 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 [2] (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.*

[2] the DIGITAL UTILITIES module, and the BIT CLOCK REGEN module also have divide-by-4 sub systems

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. Accepting jittering displays is unprofessional ! See Tutorial Question Q2.*

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.

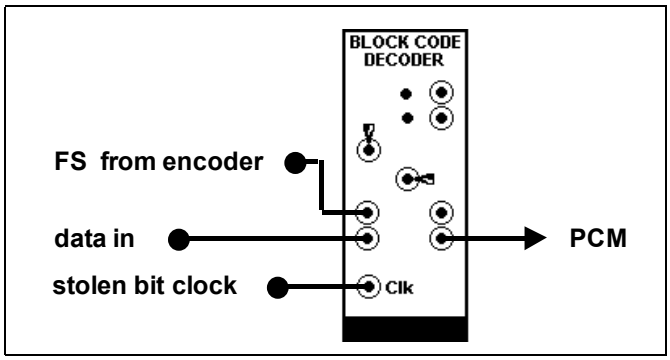


Figure 6: block code decoding

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.*

conclusion

You are now in a position to include block coding in a more complex transmission system (noisy, bandlimited) and to demonstrate its effectiveness in improving the bit error rate. This is the subject of the experiment entitled *Block coding and coding gain*.

During this experiment you should have developed techniques for obtaining oscilloscope displays which show you what you want, without need to constantly adjust and re-adjust the oscilloscope controls. Choice of the appropriate trigger signal for each display is important.

Although, for a DC message, each 4-bit word and added code bits are the same, the alternating pattern of 0-1-0-1- for the FS signal make alternate frames *different*. It would be preferred if the synchronisation technique adopted would always put the same frame first, no matter what the sweep speed.

This is a simple matter to implement. See Tutorial Question Q2.

TUTORIAL QUESTIONS

Q1 *when adding check bits for parity checking, the bits of the alternating pattern 1 - 0 - 1 - 0 - for frame synchronization in the LSB position were ignored. Explain.*

Q2 *explain how dividing the frame synchronization signal by two is often a help in obtaining and maintaining stable, and repeatable, oscilloscope displays in the context of this experiment.*

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

BLOCK CODING AND CODING GAIN

BLOCK CODING AND 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*** *in this Volume.*

ADVANCED MODULES: *PCM ENCODER, PCM DECODER (version 2 or later), 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 frequencise 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, according to the scheme detailed in the experiment entitled *The noisy channel model* in volume D1.

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

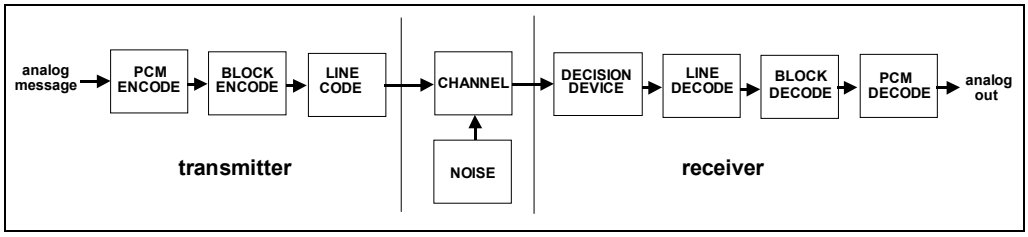


Figure 1: the system block diagram

EXPERIMENT

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

transmitter

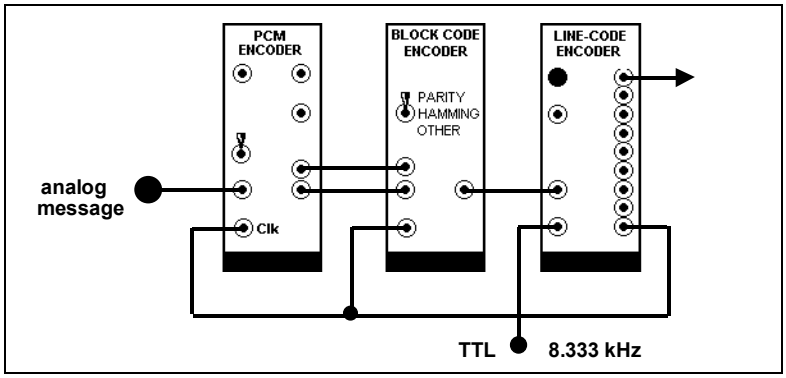


Figure 2: the transmitter

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

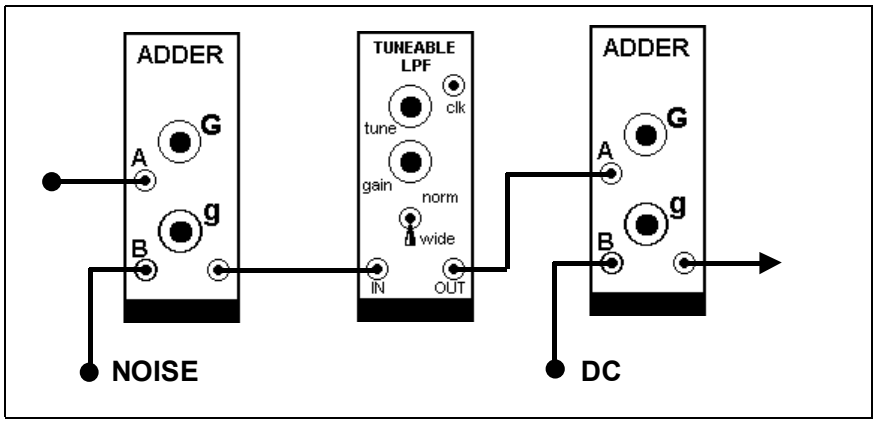


Figure 3: the channel model

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. See Tutorial Question Q1.*

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.*

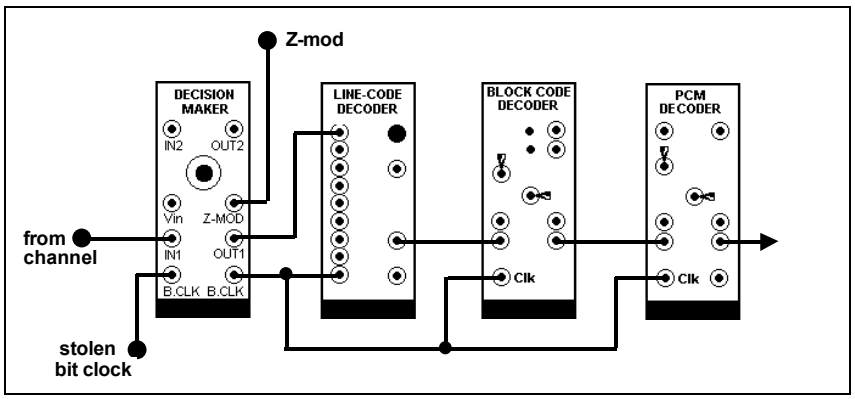


Figure 4: the receiver model

T8 *with the channel output connected to the DECISION MAKER input, while observing either an eye pattern or a snapshot display, and using the decision instant marker, adjust the decision instant. See Tutorial Question Q2.*

T9 *ensure matching line codes are selected at both transmitter and receiver. NRZ-L is suggested.*

T10 *at the BLOCK CODE DECODER select frame synchronization using the EMBEDded frame synch. signal, and PARITY block coding.*

T11 *at the PCM DECODER select 4-bit LINEAR decoding and frame synchronization using the EMBEDded frame synch. signal.*

T12 *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.

T13 *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.*

T14 *while looking at your chosen frame entering the channel (CH1-A), locate it on the other oscilloscope trace (CH2-A) at the channel output.*

T15 *move CH2-A forward to the output of the DECISION MAKER and confirm the regeneration of the desired wave shape.*

T16 *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. See Tutorial Question Q3.*

T17 *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.

See Tutorial Question Q4.

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:

[1] signal-to-noise-plus-distortion ratio

- 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.

T18 *carry out as many of the above 'A - B' comparisons as you consider important. Compare with expectations.*

TUTORIAL QUESTIONS

Q1 *it was suggested in Task T4 that you adjust the channel bandwidth until there was 'obvious bandlimiting'. How did you decide on the bandwidth, and why ?*

Q2 *describe your method of adjusting the decision instant.*

Q3 *what was the approximate signal delay between input and output of the system ? What factors contribute to this delay ? If more than one source of delay, could you attribute contributions to each source ?*

Q4 *some of the observations made were of output SNR. The noise here was made up of at least three components, namely quantization noise, intentionally added noise, and distortion components. How were these accounted for in your findings ?*

Q5 *describe observations you made, not included as specific Tasks, and your conclusions.*

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 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.

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

CONVOLUTIONAL CODING

CONVOLUTIONAL CODING

ACHIEVEMENTS: *setting up and testing of a convolutional encoder and decoder pair. Inclusion into a noisy, bandlimited communication system; observation and measurement of changes to BER.*

PREREQUISITES: *completion of the experiment entitled* ***BER measurement in the noisy channel*** *in this Volume.*

ADVANCED MODULES: *CONVOLUT`L ENCODER, TIMS320 DSP-DB (with decoding EPROMS), and TIMS320 AIB; plus all those modules required for the pre-requisite experiment, namely LINE-CODE ENCODER, LINE-CODE DECODER, DECISION MAKER, ERROR COUNTING UTILITIES, WIDEBAND TRUE RMS METER, an extra SEQUENCE GENERATOR, BASEBAND CHANNEL FILTERS, NOISE GENERATOR. TRUNKS are optional*

PREPARATION

The experiment is divided into two parts - A and B.

Part A introduces the CONVOLUT`L ENCODER module, and a pair of modules which together perform the decoding. These modules are examined in relative isolation.

Part B places them into a communications system, where their contribution is to reduce the errors introduced by the noisy, bandlimited channel.

convolutional encoding

It is assumed you have had some introduction to the concept of coding in general, and of convolutional coding in particular. Suffice to say that for this experiment there is no need to know any of the theory which gave rise to this coding scheme, although it would, of course, add to your appreciation of the experiment.

The aim of the experiment is to show that:

- the form of convolutional encoding implemented is such that extra bits are added to a serial input message (data) stream
- after encoding the output bit rate is twice that of the input bit rate

- it is not easy (impossible?), by observing the input and output simultaneously, to describe what the coding scheme is
- an algorithm exists for recovering (deciphering) the original message from the encoded bit stream
- there are benefits to be gained by performing this encoding !

As in other forms of coding, bits are added to the original data stream. Thus, if the channel over which the message is transmitted is band limited, then the bit rate must remain as before, and so the message rate - the rate at which the wanted message arrives at the far end - will be slowed. But the error rate will be reduced. Overall there is an advantage in this. See Tutorial Question Q1.

Convolutional encoding is implemented with the CONVOLUT`L ENCODER module, the front panel of which is depicted below.

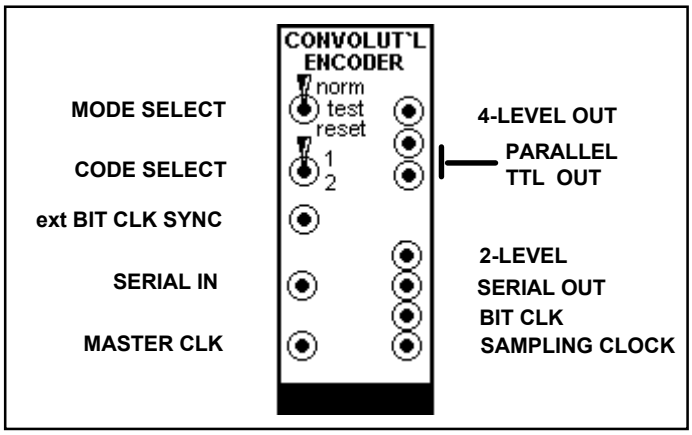


Figure 1: the CONVOLUT`L ENCODER front panel

Descriptions of the various front panel connections are:

inputs

- ***mode select:*** three modes (see panel) selected by the three-position toggle
- ***code select:*** there are two convolutional coding schemes, selected with a two-position toggle switch. Each is rate ½, which means there are as many code bits added as there are original message bits. CODE 1 is of constraint length 3; CODE 2 is of constraint length 4.
- ***ext bit clk sync:*** for the case that there are two modules being driven by the 8.333 kHz MASTER CLOCK (as in this experiment), and where each divides this by four, the resulting 2.083 kHz need to be kept in phase. A patch from the LINE-CODE ENCODER 2.083 kHz output to this 'ext bit clk sync' input will force this condition.
- ***serial data:*** the input data (message) to be coded - from the message source.
- ***master clk:*** from which all other clocks are derived by division. It is four times the output bit rate (and so eight times the message bit rate).

outputs

- ***4-level:*** a 4-level output; not involved in this experiment
- ***parallel TTL:*** two adjacent bits of the output bit stream; not involved in this experiment.

- ***2-level:*** a bi-level ('analog') version of the serial output
- ***serial:*** TTL level encoded version of the *serial data* (message) input.
- ***bit clk:*** in phase with the serial output. Becomes the 'stolen bit clock' for the receiver.
- ***sampling clock:*** half the rate of the output bit clock; correctly phased to drive the message source (a SEQUENCE GENERATOR in this experiment).

The CONVOLUT`L ENCODER module accepts serial data (the message) as input. Its output may be in serial form, but is also available in parallel format (which includes a 4-level signal). Only the serial format will be considered in this experiment.

The common bit rate for most of TIMS experiments is 2.083 kHz, and a clock at this rate is available from the MASTER SIGNALS module. But if this is to be the transmitted bit rate, then a clock at half this rate is required to run the SEQUENCE GENERATOR which will be used to represent the message. Such a clock (1.042 kHz), called the sampling clock, is provided by the CONVOLUT`L ENCODER.

encoding schemes

Reference should now be made to the *Advanced Modules User Manual* for more detail regarding the coding schemes, test patterns, bit formats, and other technical details (including references).

convolutional decoding

The decoder is implemented with a pair of TIMS digital signal processing modules, namely the TIMS320 DSP-DB and the TIMS320 AIB.

TIMS320 DSP-DB

This development board ('DB') module must be fitted with two EPROMS (erasable programmable read only memory) which contain software for the decoding algorithm.

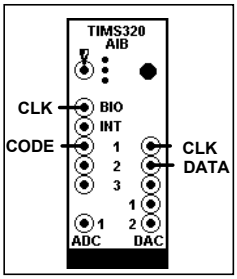
Check that the four on-board MEMORY SELECT jumpers are in the 'A' position.

The decoding algorithm can also be obtained from a PC connected to the front panel SERIAL LINK, but this option will not be invoked for this experiment.

TIMS320 AIB

This analog interface board ('AIB') module serves as the interface to the decoding software of the TIMS320 DSP-DB. For this it is essential that the TIMS320 AIB be inserted into the TIMS frame *immediately to the right* of the TIMS320 DSP-DB.

It is a general purpose module, and the front panel connections are re-defined for each EPROM installed.



AIB front panel

Only four of the front panel connections are required for operation of the TIMS320 AIB module in convolutional code decoding mode, as shown opposite.

The *input* CLK is that associated with the convolutionally encoded CODE input. This is at 2.083 kHz.

The *output* CLK is at the message (DATA) rate of 1.042 kHz. It will be used for the BER instrumentation.

The function of the three-position toggle switch is described in Table 1 below, and explained later.

toggle POSITION	***Decoder mode***	***AUTOMATIC operation***	***MANUAL operation***
UPPER	automatic	requires test code as input	not used
MIDDLE	manual	decodes as 'normal'	initially branch bit randomly selected
LOWER	manual (reverse of middle)	decodes as 'reverse'	branch bits reversed

Table 1: AIB Toggle Switch function

the complete system

A block diagram of the system to be studied, but without BER instrumentation, is shown in Figure 2 below.

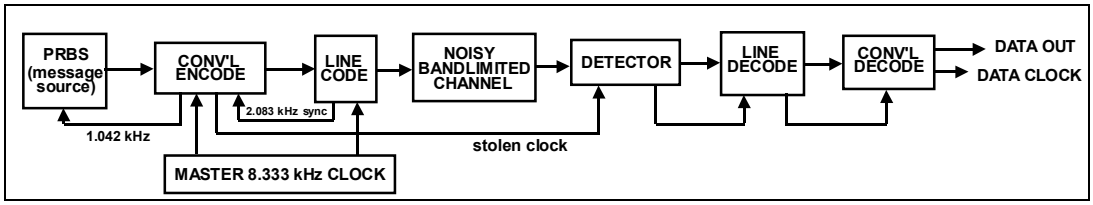


Figure 2: block diagram of the system to be modelled

In particular this shows the sources of each of the clocks, all derived from the TIMS 8.333 kHz MASTER SIGNALS clock.

EXPERIMENT - PART A

In Part B of this experiment the encoder and decoder of Part A will become part of a transmission system operating from the 8.333 kHz clock of the MASTER SIGNALS module.

Part of this system is a LINE-CODE ENCODER module, which produces a clock at one quarter of this rate, namely 2.083 kHz.

The convolutional encoding scheme to be implemented requires input data at half this rate again; so it in turn produces a 1.042 kHz clock for the message, provided by a SEQUENCE GENERATOR.

Detailed information about the three new modules to be examined - the CONVOLUT`L ENCODER, the TIMS320 AIB, and the TIMS320 DSP-DB - may be found in the *Advanced Modules User Manual*. However, it is not necessary to refer to this for the purposes of the experiment. There are several on-board settings to be made, but it is assumed this will have been done by your Laboratory Manager.

encoding

A model of the encoding part of the block diagram of Figure 2 is shown in Figure 3 below.

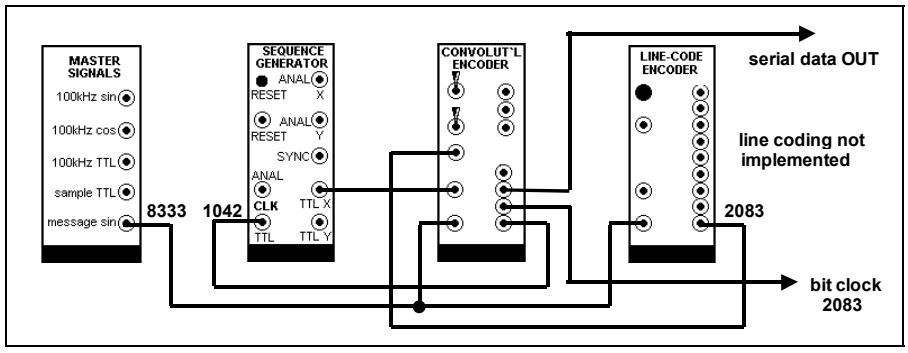


Figure 3: model of the encoding section of Figure 1

To set this model up the following steps are recommended.

T1 *set the SEQUENCE GENERATOR for a short sequence (both toggles of the on-board switch SW2 should be UP).*

T2 *patch up as shown in Figure 3.*

T3 *check that the clock and synchronization signals are present, and on the frequencies indicated in Figure 3.*

The LINE-CODE ENCODER is being used although for the present no line coding is being implemented. There is no need, then, to press its RESET button.

Note that both the LINE-CODE ENCODER and the CONVOLUT`L ENCODER are clocked by the same 8.333 kHz MASTER SIGNAL, which they immediately divide-by-four. To keep their dividers in step, a sync. signal is sent from the former to the latter.

T4 *momentarily press RESET on the CONVOLUT`L ENCODER (upper toggle switch). This must be done again, if ever the clock or synch. signal is broken, then reconnected.*

T5 *on the CONVOLUT`L ENCODER:*

a) select CODE 1 with the lower toggle switch

b) momentarily RESET with the upper toggle switch.

c) select TEST CODE with the upper toggle switch.

T6 *simultaneously observe B.CLK (bit clock of coded data) and S.CLK (sample clock of un-coded message data). Confirm their relative frequencies (as per Figure 3), and phases (edges line up).*

T7 *simultaneously observe B.CLK and the encoded output from DATA. This is a test pattern. It results from sending the encoder (with the upper toggle switch on TEST CODE) a stream of ones (1, 1, 1, 1, 1....). Depending upon the code (CODE 1 or CODE 2) so a different pattern emerges from the encoder. The decoder uses these patterns to recognise the coding scheme and so obtain bit synchronization (see later). The test patterns are described in the* ***Advanced Modules User Manual****.*

T8 *select CODE 1, and switch to NORMAL encoding. Synchronize the oscilloscope to the SEQUENCE GENERATOR SYNC signal, and observe both the input message sequence and the encoded output. Confirm the difference in bit rate. Can you see any relationship between the two patterns ? Unlikely ! But there is, of course; just ask the decoder !*

decoding

The convolutional decoder is implemented in software. Two modules are required, the TIMS320 AIB and the TIMS320 DSP-DB. These should already have been configured, by your Laboratory Manager, for correct operation.

However there is a jumper, J1, on the DB board (located near the EPROM U5). This has two positions, 'L' and 'H'. In the 'L' position the decoder is set up to decode CODE 1 of the encoder module (use 'H' for CODE 2).

T9 *before inserting the TIMS320 DSP-DB check the position of J1 (explained above).*

T10 *insert the two modules into adjacent slots of the TIMS frame, the TIMS320 AIB immediately to the right of the TIMS320 DSP-DB. Set the PROGRAM/RUN toggle on the DB to RUN. Press the RESET button. Set the toggle switch on the AIB to the central position.*

T11 *patch according to Figure 4. The incoming serial data goes direct from the encoder output to the decoder input. The LINE-CODE ENCODER module is being used for clock generation, but will not yet be used for line coding. So a LINE-CODE DECODER is not yet required. Note also that the noisy channel of Figure 1 is not yet implemented. So no detector (decision maker) is required.*

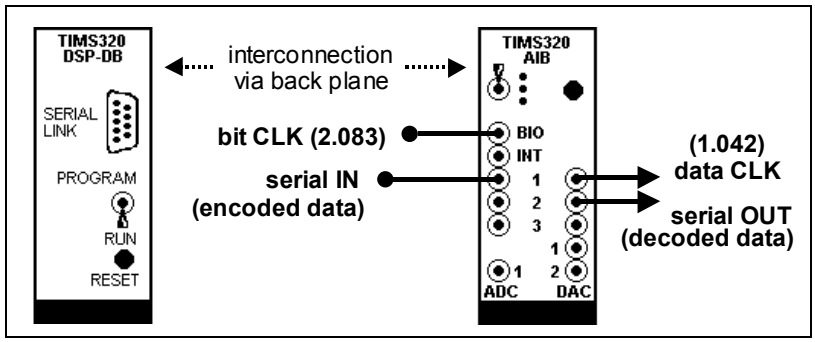


Figure 4: the convolutional decoder model.

T12 *confirm the presence of a data clock (1.042 kHz) from the decoder (#1 TTL output of the AIB). This will be required later for the BER instrumentation.*

T13 *compare the message data, from the SEQUENCE GENERATOR, and the decoded output from the decoder (TTL output #2 of the AIB). There may or not be agreement. In any case, when using the oscilloscope, remember that there will be a considerable delay (many clock periods) between the two sequences, due to the coding and decoding processes.*

If the decoded output is in error, then it (the decoder) must be incorrectly synchronized. Due to the code in use, it can only be one bit out in its timing.

There are two methods of synchronization.

1. ***manual*** synchronization: if the decoder is not correctly synchronized to the clock, this can be corrected by synchronizing to an adjacent clock period. This is accomplished by moving the AIB toggle switch from CENTRE to LOW (or the reverse) position. On a real message (or a very long sequence) it would be impossible to confirm synchronization by merely observing the decoded message; so this could be a 'hit and miss' procedure.
2. ***automatic*** synchronization. To initiate this:
    a) switch the encoder toggle to TEST mode (sends a known pattern)
    b) switch the AIB toggle UP to AUTOMATIC mode

c) synchronization acknowledged by the AIB LED lighting

d) switch the AIB toggle to the central position - normal decoding

e) return the encoder toggle from TEST to NORMAL

T14 *try the above two methods of synchronization. But remember, although the automatic method is reliable, at the moment you have only your eyes (and a short sequence) to confirm it. Soon you will have some instrumentation to support your findings.*

T15 *change to CODE 2 at the transmitter. Move J1 on the DB board to 'H'. Repeat the previous Task.*

T16 *record the delay (in clock periods) between the input and output message. This is a processing delay introduced by the coding and encoding process. There will be an additional delay when a bandlimited channel is introduced.*

manual encoding

If you are interested in personally checking the encoding algorithm, you should first refer to the *Advanced Modules User Manual* for details of the two coding schemes. You could then check that they have been correctly implemented by carrying out a bit-by-bit analysis of the encoder outputs (on a short sequence).

When you are satisfied with your progress it is time to introduce convolutional coding to the transmission system.

EXPERIMENT - PART B

A transmission system incorporating a noisy, bandlimited channel was examined in the experiment entitled *BER measurement in the noisy channel* (in this Volume). This system will be used again, but now with the addition of convolutional coding.

It is shown modelled in Figure 5 below.

Since it is not possible to include all required modules in a single TIMS 301 frame it is convenient that the NOISY CHANNEL MODEL be accommodated in a separate frame. The channel requires only a single wire input and output. It requires no clock signals. So it could be in an adjacent TIMS 301, interconnected by TRUNKS if necessary.

Remember, the rate through the channel will be at 2.083 bits-per-second. Without coding the clock of the SEQUENCE GENERATOR will also be at this rate.

With coding the bit rate in the channel will still be 2.083 bits-per-second. So the SEQUENCE GENERATOR will need to be clocked at *half this rate* (both codes are of rate ½), as in the model already prepared in Part A above.

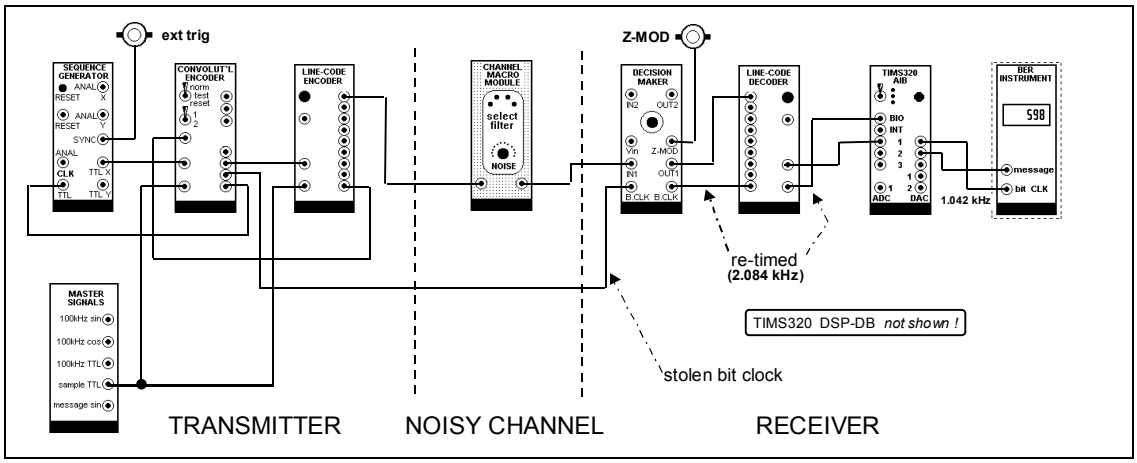


Figure 5: the system with convolutional coding.

For details of the noisy channel model refer to the experiment entitled *The noisy channel model* (Volume D1). For details of the BER instrumentation see the Chapter entitled *BER instrumentation macro model* (this Volume).

patching

A systematic patching procedure is recommended. At each stage always check that, after achieving synchronization, the input sequence has been successfully recovered at the output.

When you are already experienced in patching up these larger systems you may feel such small steps are unnecessary.

T17 *insert just the LINE-CODE ENCODER and LINE-CODE DECODER modules between the transmitter and receiver. Clock each with the* B.CLK *of the CONVOLUT`L ENCODER. Use* NRZ-L *code.*

T18 *insert the DECISION MAKER between the LINE-CODE ENCODER and the LINE-CODE DECODER. Clock the DECISION MAKER with the* B.CLK *of the CONVOLUT`L ENCODER. Clock the LINE-CODE DECODER with the* B.CLK *from the DECISION MAKER. Since there is no bandlimiting the decision point can be set almost anywhere (except on a transition).*

T19 *insert channel #3 of a BANDPASS CHANNEL FILTERS module between the LINE-CODE ENCODER and the DECISION MAKER. Now that bandlimiting is included, it will be necessary to examine the eye pattern at the output of the channel, and adjust for the best decision instant.*

T20 *place an ADDER at both the input and the output of the channel filter. Use the '**G**' inputs, setting both gains near unity.*

T21 *confirm the message is still being decoded successfully.*

T22 *add the instrumentation*

T23 *check that the reference SEQUENCE GENERATOR is on a short sequence, as is that at the transmitter (both toggles of SW2 should be UP). Carry out the alignment procedure of the two sequences going into the EXCLUSIVE-OR of the DECISION MAKER. This can be checked by eye, but also by the instrumentation.*

T24 *change both sequence generators to a long sequence (both toggles of SW2 should be DOWN). Re-align the system (ie, synchronize the CONVOLUT`L DECODER, re-align the reference SEQUENCE GENERATOR). Use the instrumentation to show that there are no errors.*

T25 *patch maximum available noise from a NOISE GENERATOR to the '**g**' input of the INPUT ADDER, and rotate '**g**' fully clockwise.*

T26 *observe the channel output, and set the SNR to 0 dB by reducing the signal into the INPUT ADDER with the '**G**' control. Increase the channel output with the OUTPUT ADDER '**G**' control until the signal approaches the TIMS ANALOG REFERENCE LEVEL; but this may not be achievable.*

T27 *before making serious measurements:*

a) confirm, by watching the COUNTER, that the BER reduces as the SNR is increased (using the ATTENUATOR on the NOISE GENERATOR).

*b) centre the signal into the DECISION MAKER about the 25 mV input threshold (see **BER measurement in the noisy channel**, this Volume, under 'DC threshold adjustment').*

BER measurement with coding

T28 *make some serious BER measurements, with convolutional coding operative.*

BER measurement without coding

To estimate the gain introduced by the convolutional coding it is necessary to repeat the measurements, but *without* coding. To remove the convolutional coding there are seven changes to be made (in Task **T29**). These will be:

to the transmitter:

1. ***bypass the CONVOLUT`L ENCODER***: move the patch lead from the DATA *output* of the CONVOLUT`L ENCODER to the DATA *input*.
2. ***change the clock to the SEQUENCE GENERATOR***: move the patch lead from the S.CLK output of the CONVOLUT`L ENCODER to the B.CLK of the LINE-CODE ENCODER.

to the receiver

3. ***bypass the CONVOLUT`L DECODER***: move the patch lead from the TTL OUTPUT #2 of the AIB module to the TTL INPUT #1.
4. ***change the clock to the reference SEQUENCE GENERATOR***: move the patch lead from the TTL OUTPUT #1 of the AIB to the STROBE output of the LINE-CODE DECODER.

to the system

5. press all re-set buttons (not strictly necessary, but a matter of principle).
6. re-align the reference SEQUENCE GENERATOR (with no noise).
7. re-set the decision instant of the DECISION MAKER.

T29 *remove the convolutional coding, and repeat the BER measurements.*

interpretation

The procedures outlined above have enabled you to make serious measurements of BER, with and without convolutional coding.

The two cases had different message rates, although the same bit rate through the same channel.

Make sure you have enough information to enable an answer to Tutorial Question Q1.

TUTORIAL QUESTIONS

Q1 *convolutional (and other) encoding used over a bandlimited channel results in a reduced message bit rate, but offers the benefit of less errors. Taking account of the different message rates, discuss how you might attempt to estimate the gain obtained with the two convolutional codes provided by the CONVOLUT`L ENCODER.*

TCM - TRELLIS CODING

TCM - TRELLIS CODING

ACHIEVEMENTS: *introduction to trellis code modulation (TCM) implemented as one-dimensional ASK, with soft-decision Viterbi decoding. Measurement of BER over a noisy baseband channel. Comparison with a reference signal to estimate coding gain.*

PREREQUISITES: *completion of the experiments entitled* ***Convolutional coding*** *(in this Volume) and familiarity with bit error rate (BER) measurement.*

EXTRA MODULES: *CONVOLUT'L ENCODER, DECISION MAKER, DIGITAL UTILITIES, ERROR COUNTING UTILITIES, INTEGRATE & DUMP, LINE-CODE DECODER, LINE-CODE ENCODER, NOISE GENERATOR, TMS320 DSP-HS, WIDEBAND TRUE RMS METER, a second SEQUENCE GENERATOR.*

PREPARATION

Trellis coding offers a means of increasing data rate without increasing transmitted bandwidth. This is ideally suited to experimental verification.

The gain is achieved with multi-level, multi-phase signalling. In this experiment it will be implemented with 4-level ASK, which is indeed multi-level, although only one phase dimension. Thus the coding gain is relatively small.

In this experiment the performance advantages of TCM with Viterbi decoding are investigated. Details of the operation of the encoding and decoding processes are not included here. The TCM bit error rate (BER) will be measured under a defined set of conditions. This is then compared with performance when transmitting the same pseudo-random binary sequence (PRBS), of the same bandwidth, operating under similar conditions, but without TCM.

Information regarding the coding (in the CONVOLUT'L ENCODER), and the decoding algorithm (EPROM in the TIMS320 DSP-HS), may be obtained from the *Advanced Modules User Guide*.

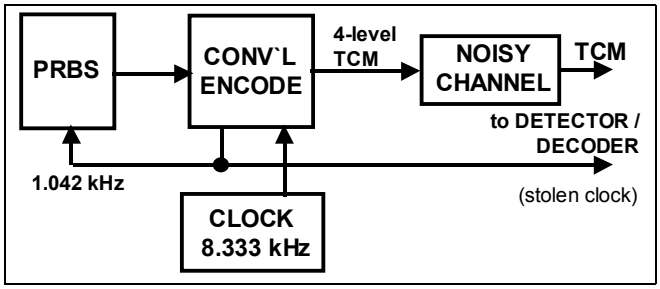


Figure 1: TCM generator & channel

The TCM generator and channel is illustrated in block diagram form in Figure 1 above. Note that this is a baseband system, although it could easily be modified to include modulation to a carrier frequency – typically 100 kHz in a TIMS system. This would be transparent to the TCM and would not materially affect the understanding of the experiment, or its results.

Viterbi decoding

The Viterbi soft-decision decoding algorithm is implemented using the TIMS320 DSP-HS module, in which your Laboratory Manager will have installed the appropriate EPROM. This will be referred to, below, as the *Viterbi decoder*.

The received TCM signal will be reconstituted by a decision maker implemented by an INTEGRATE-&-HOLD subsystem in the INTEGRATE & DUMP module. This will provide performance equivalent to matched filtering (since we are using flat top NRZ pulses).

The output of the INTEGRATE-&-HOLD, a 4-level ASK, is the input to the Viterbi-decoder. In turn, the decoder output (under no-noise conditions) is the original serial PRBS message.

A stolen bit clock will be used.

A block diagram of the detector/decoder is shown in Figure 2 below.

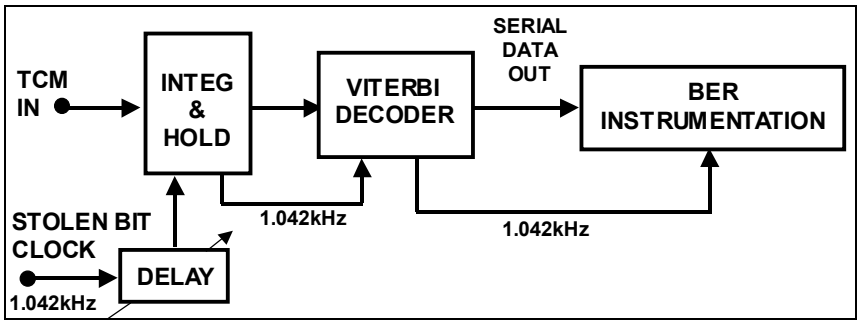


Figure 2: TCM detector/Viterbi decoder & BER instrumentation

overall system

Transmitter and receiver will be connected via a noisy baseband channel. At the receiver output will be instrumentation to measure the BER under defined signal-to-noise-ratio (SNR) conditions.

The SNR will be measured at the *output of the integrate-and-hold sub-system*.

The message data rate will be fixed at 1.042 kbit/s.

reference system

The reference system will be a pseudo-random binary sequence, the same sequence as was used to generate the four-level TCM signal. It will:

1. have the same transmitted signal bandwidth
2. have the same message data rate
3. be transmitted via the same channel (bi-polar format)

The SNR of the reference system is then compared with that of the TCM system for the same BER. The difference in SNR is then the coding gain achieved by the TCM.

note that: the reference system uses a 2-level signal, with no error correction, and so the message bit rate and the raw (symbol) data rate are the same.

The TCM system *adds* bits to the raw data, and so for a 2-level transmitted format the message rate would need to be slower than the raw data rate, for the same transmitted bandwidth. But since in this case the TCM is to be a 4-level signal, and there is one extra bit added per message bit, the message rate will be the same as that of the reference system.

A block diagram of the reference system is shown in Figure 3 below.

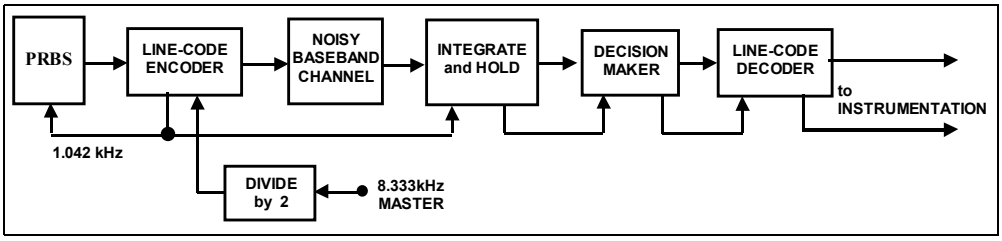


Figure 3: reference system

The line-code modules are present for practical reasons: the decoder provides a convenient conversion from analog-to-TTL between the decision maker and the error counting module. The encoder is included for compatibility.

EXPERIMENT

You are about to model, separately, two relatively large systems (with some common sub-systems). As usual, it is suggested that these be built and tested in stages, as outlined below.

the TCM system

transmitter

T1 *before plugging in the SEQUENCE GENERATOR MODULE select a short sequence (both toggles of the on-board switch SW2 UP). Then patch up the transmitter as shown in Figure 4 below.*

T2 *confirm the SEQUENCE GENERATOR is clocked at 1.042 kHz (one eighth of the 8.333 kHz MASTER CLOCK).*

T3 *on the CONVOLUT`L ENCODER select NORMAL and CODE 2 with the two toggle switches. Confirm a 4-level output from OUT_4.*

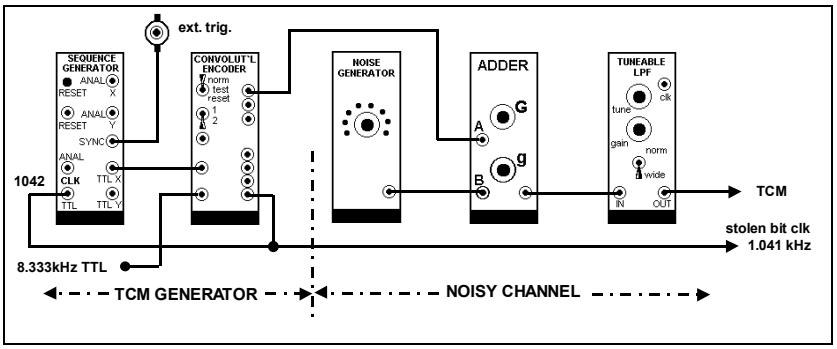


Figure 4: TCM generator and channel model of Figure 1.

the channel

The channel will be modelled with a TUNEABLE LPF module, set to its widest bandwidth. This will have negligible effect upon the signal waveform. It is present as a formality, but also convenient in that it provides some gain adjustment. At its input is an ADDER, to combine the TCM signal with NOISE.

T4 *patch up the channel as just described. Initially add no noise. Set the front panel GAIN control of the TUNEABLE LPF to a mid-way position.*

T5 *adjust the 4-level signal from the ADDER to about 0.2 volt peak-to-peak (this will be reset later) to the TUNEABLE LPF.*

TCM detector

T6 *before inserting the INTEGRATE & DUMP module read about it in the* ***Advanced Modules User Guide****. Then:*

a) set the on-board switch SW1 to I&H1. This makes the I&D1 sub-system perform an integrate and hold operation

b) *set the on-board switch SW2 to I&D2. This makes the I&D2 sub-system perform an integrate and dump operation.*

c) set the toggles of the on-board switch SW3; upper to the LEFT, and lower to the RIGHT. These govern the range of delay introduced by the DELAY front panel control.

T7 *patch the detector/decoder and instrumentation according to the details of Figure 5 below.*

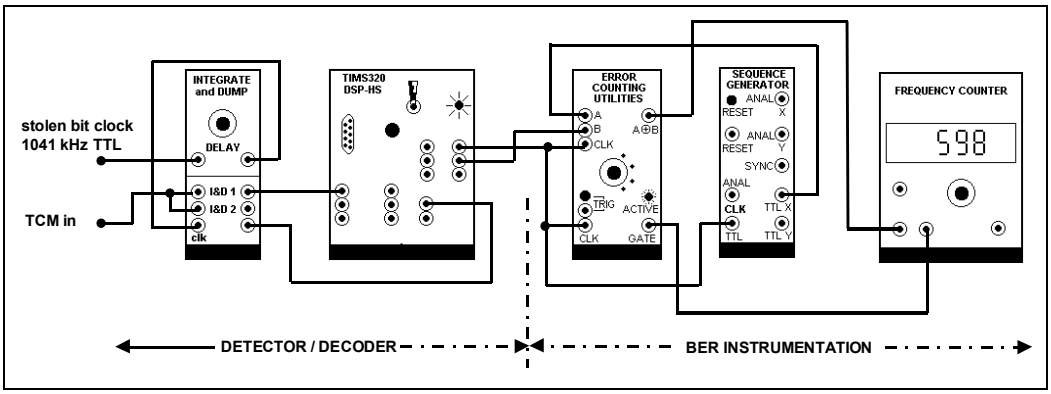


Figure 5: TCM detector/decoder model of Figure 2

The bit clock phase (delay) is adjusted so that the integration of the INTEGRATE & HOLD operation is timed correctly. There are two methods of adjusting the timing, namely:

a) observe the INTEGRATE & HOLD operation (I&D 1), and adjust for a 4-level waveform (otherwise it will be an 8-level waveform).
b) watch the output of the INTEGRATE & DUMP operation (I&D 2) and adjust for single slope ramps within the bit clock period.

T8 *connect the TCM to both the I&D 1 and the I&D 2 inputs of the INTEGRATE & DUMP module. Observe the INTEGRATE & HOLD operation from the I&D 1 output, and the INTEGRATE & DUMP operation from the I&D 2 output. Adjust the delay (start from the fully anti-clockwise, or minimum, delay condition) for a 4-level signal from the INTEGRATE & HOLD output (otherwise will be an 8-level signal); alternatively adjust for a constant slope ramp from the INTEGRATE & DUMP output. With no noise these are simple operations, and both results should occur simultaneously.*

T9 *re-set the GAIN of the TUNEABLE LPF so that the input to ADC 1 of the TIMS320 DSP-HS module is 3 volt peak-to-peak (the 4 levels should be ±1.5 and ±0.5 volts). Correct operation of the Viterbi decoder is dependent on this adjustment.*

Viterbi decoder

T10 *confirm the stolen clock to the BIO input of the Viterbi decoder is at 1.042 kHz.*

T11 *confirm that there is a TTL output sequence from the Viterbi decoder (DSP-HS TTL output #2), and a clock of 1.042 kHz from TTL output #1.*

The TTL output sequence should be the same as that sent from the transmitter. Confirm this by inspection (it is a short sequence) of the two waveforms. Remember that there is a considerable off-set (processing and other delay) between the two waveforms.

If there is a polarity inversion this can be reversed by flipping the USER I/O toggle on the front panel of the Viterbi decoder module (UP is one polarity, CENTRE and DOWN the other).

Checking for no errors is easier, and certainly more positive, by using the error counting instrumentation.

BER instrumentation

This sub-system is common to both the TCM and the reference system, although with different input signals.

Refer to the experiment entitled *BER instrumentation macro model* (in this Volume) for patching, alignment, and measurement procedures.

T12 *error measurements are best made with a long sequence, but during set-up a short sequence is more convenient. So before plugging in, set the reference SEQUENCE GENERATOR to a short sequence to match that at the transmitter.*

T13 *patch up the instrumentation.*

T14 *using the X-OR gate of the ERROR COUNTING UTILITIES align the received and locally generated sequences (momentarily connect the output of the X-OR gate to the RESET input of the reference SEQUENCE GENERATOR - the procedure is described in the experiment entitled* ***PRBS generation*** *- Volume D1).*

T15 *demonstrate the absence of errors when the two sequences are aligned (but confirm their presence when mis-aligned).*

T16 *add noise into the channel and confirm that errors start accumulating for low SNR - say below about 10 dB.*

When satisfied all is operating satisfactorily, increase the sequence length (transmitter and reference), re-align, and then make some BER measurements.

T17 *change the SEQUENCE GENERATOR modules to long sequences (both toggles of the on-board switch SW2 DOWN).*

T18 *re-align received and reference sequences at the X-OR gate.*

T19 *adjust SNR to give about 1 error per few seconds. Record the error rate as* BER_1. *Record this as* SNR_1 *in dB. Confirm this result is repeatable.*

the reference system

The TCM system has been evaluated. The reference system will now be set up.

T20 *before plugging in the DECISION MAKER, set the on-board switch SW1 to NRZ-L, and SW2 to INT. It is assumed the z-modulation jumper J1 will have been set by your Laboratory Manager to suit the oscilloscope in use*

T21 *patch up the reference system. The model is shown in Figure 6 below. There are sub-systems common with the TCM system. Clock signals need to be re-arranged. Leave the bandwidth of the channel (the TUNEABLE LPF) as set for TCM, and set the gain to maximum (control fully clockwise). Leave the INTEGRATE & DUMP module on-board switches as set for the TCM system.*

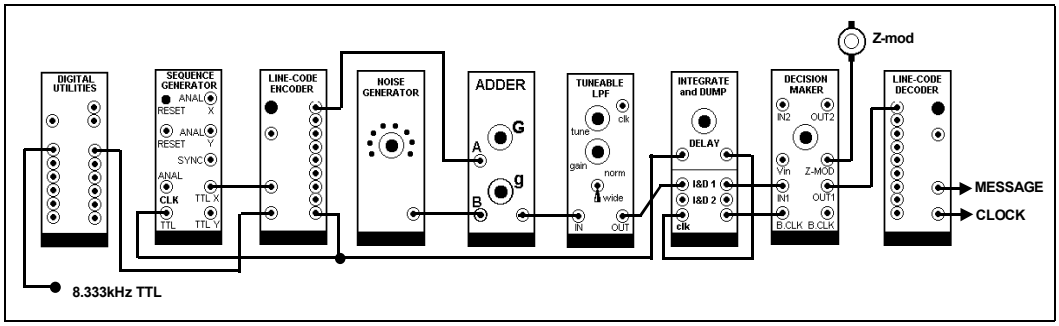


Figure 6: model of reference system

T22 *for simplicity start with a short sequence, and no noise. Adjust the gain of the ADDER so that the signal from the I&D 1 output of the INTEGRATE & DUMP module is about 2 volt peak. Adjust the INTEGRATE & DUMP clock signal delay so this is a two-level waveform.*

T23 *confirm the same signal as observed in the previous TASK appears at the output of the DECISION MAKER. This will be a TTL waveform. With a wide band channel the decision point setting is unlikely to require precise setting. Explain.*

T24 *confirm that alignment of the two sequences into the ERROR COUNTING UTILITIES module is possible.*

T25 *change to long sequences in both SEQUENCE GENERATOR modules, reset them and the LINE-CODE DECODER, and re-align the system.*

T26 *adjust SNR to match the error rate BER_1 (of the TCM arrangement). See the comment on this matching in the Appendix to this experiment. Record the SNR as SNR_2 in dB.*

T27 *the difference between SNR_1 and SNR_2 is the coding gain obtained using TCM. What might it be expected to be ?*

TUTORIAL QUESTIONS

Q1 *what was the message bit rate of the TCM system ?*

Q2 *discuss the terms message data rate, symbol rate, and raw data rate (you may have different names again for them), and the relationships between them, in this experiment.*

Q3 *what was the symbol rate of the TCM system ?*

Q4 *the noise was combined with the signal at the input to the channel. Could it have been added at the output of the channel? Explain.*

Q5 *in this experiment for how long will the gate of the COUNTER stay open if the front panel switch of the ERROR COUNTING UTILITIES is set to count 10^5 pulses ?*

Q6 *the TCM system added symbols to the message serial data stream, and transmitted the message at the same rate, yet had the same transmitted bandwidth as the reference system. How was this achieved ?*

Q7 *in aligning the received and reference sequences you were instructed to 'momentarily connect the output of the X-OR gate to the RESET input of the reference SEQUENCE GENERATOR'. In fact, this connection should be for more than a fraction of a second – explain.*

Q8 *in this experiment you will have found that with trellis coding there is a net advantage compared with the uncoded case. This advantage is called the coding gain and is expressed as the SNR degradation that can be tolerated to maintain a specified error probability when coding is introduced. In this instance it will be found to be approximately 2.5dB. You will have noted that with trellis coding four levels are used compared with only two in the absence of coding (i.e., with peak power remaining unchanged). This means that the raw noise margin is significantly less in the coded case.*

a. state the noise margin disadvantage of the coded case.

b. hence determine the gross difference in noise margin in the experimental results.

APPENDIX 1

matching bit error rates

An examination of a typical curve of BER versus SNR will show that the rate of change of BER for small change of SNR is rapid when the SNR is above 0 dB. Thus, if the matching of the BER of the reference system (BER_2) to that obtained previously for TCM (BER_1) is within, say, ±50%, then the error in the measured SNR will be small.

Thus the matching need only be approximate.

PWM AND PPM

PWM AND PPM

ACHIEVEMENTS: *generation and demodulation of pulse width modulated (PWM) and pulse position modulated (PPM) signals.*

PREREQUISITES: *completion of the experiment entitled* ***The sampling theorem*** *(Volume A1). It would be an advantage if the experiment entitled* ***Sampling with SAMPLE & HOLD*** *(Volume D1) has also been completed.*

ADVANCED MODULES: *INTEGRATE & DUMP*

PREPARATION

general philosophy

It has been shown that an analog message can be converted to a pulse amplitude modulated (PAM) signal, and the message recovered from it, with tolerable distortion, by simple lowpass filtering.

For example, see a suitable Text book, or the experiment entitled *Sampling with SAMPLE & HOLD*.

Suppose a train of rectangular pulses exists, of fixed width, and at the same pulse rate as a PAM signal.

- if the *width* of each of these rectangular pulses is varied according to the amplitude of each of the corresponding PAM pulses, then the new signal is said to be pulse *width* modulated - PWM (sometimes called pulse duration modulation - PDM).
- if the *position* of each of these rectangular pulses is varied according to the amplitude of each of the corresponding PAM pulses, then the new signal is said to be pulse *position* modulated - PPM

Conceptually there should be no problem with the generation of these signals, and in practice there is not. No mathematics is required to see this. In principle they may be generated exactly; and in practice, almost so.

why bother ?

If one already has a PAM signal, why bother to convert it to PWM or PPM, transmit it, then convert back to PAM before finally demodulating it ?

This is a rhetorical question. See Tutorial Question Q1.

generation

The INTEGRATE & DUMP module can be used to generate both PPM and PWM signals.

As in other experiments, there is no need to know *how* this is done. What we investigate is that it *is* done.

The same module has sub-systems which perform DIGITAL DELAY and INTEGRATE-AND-HOLD operations. Both of these are used in the demodulators to be examined. These two sub-systems were examined in the Chapter entitled *Digital utility sub-systems* in this Volume.

demodulation

Demodulation might present a problem ? As for generation, it is conceptually easy to see how each of them might be converted back, after transmission, to PAM signals. And it has been shown how the message may be recovered (reconstructed) from a PAM signal - by interpolation with a LPF. This was investigated in the experiment entitled *Sampling with SAMPLE & HOLD* (in Volume D1).

EXPERIMENT

important: throughout the work to follow:

1) detailed step-by-step instructions are *not* given in this experiment. It is left to you to plan a course of action to achieve the required results.
2) use a fixed clock frequency of 8.333 kHz.
3) make sufficient measurements to enable you to answer all the Tutorial Questions.

PWM (or PDM)

A pulse width modulator is incorporated within the INTEGRATE & DUMP module.

All it needs as input is a TTL clock and an analog message.

T1 *obtain an INTEGRATE & DUMP module. Before inserting it into the TIMS frame, set the on-board switches thus:*

a) select PWM1 with the rotary switch SW1 to position 7 or 8
b) select I&H 2 with the switch SW2 to position 2
c) select the short integrator time constant - set J1 open
d) the toggles of SW3 both to the RIGHT (required later for the DIGITAL DELAY sub-system)

T2 *insert the module into the TIMS frame. Patch an 8.333 kHz TTL clock to the CLK input, and a DC signal from the VARIABLE DC module to the I&D 1 input socket.*

T3 *synchronize the oscilloscope via its ext. trig. facility to the 8.333 kHz clock.*

T4 *observe the clock signal on CH1-A, and the output from I&D 1 socket on CH2-A.* ***note****: although the socket is labelled 'I&D1' the actual operation performed is determined by the on-board switch SW1; the 'I&D1' labelling refers to 'the output from the INTEGRATE & DUMP module sub-system #1'.*

You will see two trains of TTL pulses. That on CH2-A is the modulator output.

T5 *vary the amplitude of the DC message, and demonstrate that the signal on CH2-A is indeed a PWM signal. Note and record the properties of this modulator.*

You are now going to use a periodic message. Obviously (?) some thought must be given to the message frequency. For example, what should be its relationship to the clock frequency ? Be aware of this when making your observations. But remember this is the *modulator*, and you have only an oscilloscope. Unwanted effects may not show up in the time domain (or until *demodulation* is attempted - see later).

T6 *use an AUDIO OSCILLATOR for the message. Use the two BUFFER AMPLIFIERS in cascade to set two amplitudes V_1 and V_2; one for what you would define as 'high' depth of modulation [1], the other for 'low'. You can then change between these two later without need for constant re-measurement (see Figure 1 below). Determine amplitude limits for the message at the modulator input (I&D 1).*

[1] before the modulated edge runs into an adjacent pulse. Make sure you record how *you* have defined the 'depth of modulation'.

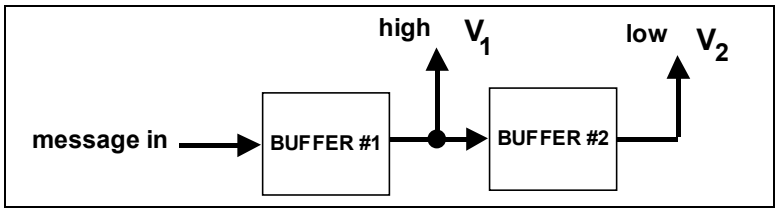


Figure 1: setting up for high and low depths of modulation

T7 *experiment with the oscilloscope trigger settings, the sweep speed, and the frequency of the AUDIO OSCILLATOR (a low frequency near a sub-multiple of the clock is a good start). With care, interesting, stable displays can be obtained, showing clearly the maximum degree of modulation allowed.*

T8 *is it possible to detect any limitations to the message frequency by looking at the PWM signal in the time domain ?*

demodulation

If the modulator performance seemed to be independent of message frequency, your experience should probably warn you that there *will* be limitations on the message frequency when attempting to demodulate.

Likewise, the bandwidth of the reconstruction filter will play a part. As a first step, set the filter bandwidth no wider than say *80% of half the clock frequency* [2].

warning: it can be shown that the spectrum of PWM has components ('sidebands') around the clock frequency more like FM than AM. As the depth of modulation increases these increase in amplitude and *spread out*. Depending upon the ratio of clock frequency ('carrier ω') and message frequency ('μ') they will eventually fall within the reconstruction filter`s passband. This is aliasing distortion. It can thus increase with depth of modulation. Be aware of this phenomenon during your investigations below.

T9 *try message reconstruction by lowpass filtering of the PWM signal. Use a TUNEABLE LPF module. Test at low and hight depths of modulation. Record your findings (for comparison with later measurements).*

You should have observed that this simple demodulator gave increased distortion with increased depth of modulation.

The preferred method of demodulation is first to convert the PWM signal to PAM, and then reconstruct with a lowpass filter. Conversion to PAM can be effected with an integrate-and-hold function.

Unlike PWM, PAM does *not* exhibit multiple sidebands around the clock frequency. Thus (aliasing) distortion is *not* going to increase with depth of modulation.

[2] what is the basis of this suggestion ?

T10 *perform an integrate and hold operation on the PWM signal. Use the* INTEGRATE-&-HOLD *sub-system* [3] *in the INTEGRATE & DUMP module (at the I&D 2 sockets; you have already set the I&H 2 function with the on-board rotary switch SW2 and J1. Confirm the conversion to PAM.*

Experiment ! it is important to experiment with message frequency and oscilloscope adjustments to obtain useful, stable displays, which reveal important system relationships and phenomena.

T11 *reconstruct the message by lowpass filtering of the PAM signal. Show that now the distortion does not increase as the depth of modulation is increased as it did in Task T9.*

T12 *determine the allowable frequency range of the message. It might be necessary to use a two-tone message - say a variable AUDIO OSCILLATOR and the fixed 2.083 kHz* MESSAGE *available from the MASTER SIGNALS module. See Tutorial Question Q3.*

PPM

One method of generation of PPM is to start with PWM, and then convert this to PPM. Before reading on consider how this might be done with available TIMS modules, and especially with the facilities of the INTEGRATE & DUMP module.

Now read on .....

There is already a modulated *position* on a PWM signal. This, as generated by the INTEGRATE & DUMP module, is the *rising* edge of the PWM signal (already examined above). This edge moves with respect to the falling edge, which is *fixed* in relation to the clock pulse. All that is then needed is to initiate the generation of a fixed-width pulse with this modulated rising edge.

The TWIN PULSE GENERATOR will do this. So will the DIGITAL DELAY sub-system in the INTEGRATE & DUMP module.

The DIGITAL DELAY sub-system is described briefly in the Chapter entitled *Digital utility sub-systems* (this Volume). By its name it is fairly obvious what it does, and closer investigation shows that it produces a fixed width pulse from each rising edge of a TTL input (the position of the pulse can be varied - delayed - but this is of no immediate interest).

[3] this sub-system is examined in the Chapter entitled *Digital Utility sub-Systems* of this Volume.

generation method #1

T13 *convert your PWM signal to PPM using the DIGITAL DELAY sub-system in the INTEGRATE & DUMP module. Use a DC message. Display both the clock and the PPM signal, and confirm the modulation.*

T14 *determine all parameters of interest. Change to a periodic message. Check that the arrangement for a quick change from low to high depth of modulation (used previously for PWM) is still useful. Observe.*

generation method #2

T15 *convert your PWM signal to PPM using one pulse of a TWIN PULSE GENERATOR module. Use a DC message. Display both the clock and the PPM signal, and confirm the modulation. See Tutorial Question Q2.*

T16 *determine all parameters of interest. Change to a periodic message. Check that the arrangement for a quick change from low to high depth of modulation (used previously for PWM) is still useful. Observe.*

demodulation

T17 *how can the message be recovered from the PPM signal ? Try reconstruction with a lowpass filter (both low and high depths of modulation). Report results.*

Since this is an FM-like signal, why not try a phase locked loop (PLL) for demodulation ? This technique is described in the experiment entitled *FM and the PLL*, in Volume A2. A block diagram for such an arrangement is shown in Figure 2.

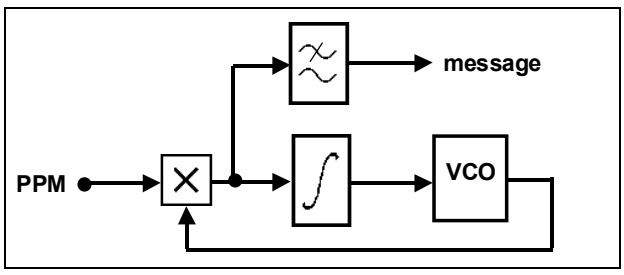


Figure 2: phase locked loop PPM demodulator

T18 *model the arrangement of Figure 2. The integrator can be the loop filter of the BIT CLOCK REGEN module. Ensure you do not overmodulate.*

Note that the PPM signal is a TTL signal, so it will be necessary to check that it is not overloading the MULTIPLIER. Perhaps a sufficient precaution would be to switch the MULTIPLIER to 'AC' mode. Otherwise it will need level shifting and scaling - use an ADDER and VARIABLE DC module.

A common method of demodulation in practice requires that the received PPM signal is first converted to PWM. See Tutorial Question Q2.

Another is lowpass filtering followed by integration. Why the integrator ?

TUTORIAL QUESTIONS

Q1 *why change PAM to PWM, then, after transmission, change it back again ? Indeed, why modulate ? In the answer to this question lie the reasons for most of communications engineering !*

Q2 *draw a block diagram illustrating a method of converting PPM to PWM.*

Q3 *explain the reasoning behind the suggestion to use a two-tone test signal when checking the allowable frequency range of the PWM system (Task T12).*

Q4 *for each of the modulators, explain how you decided when the output was 'fully modulated'.*

Q5 *why does the demodulator of Figure 2 show an INTEGRATOR and a LPF ?*

QAM AND 4-PSK

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, as well as the experiment entitled* ***Phase division multiplex*** *in Volume A2.*

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

TRUNKS: *see your Laboratory Manager about the QPSK signal(s) at TRUNKS*

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.

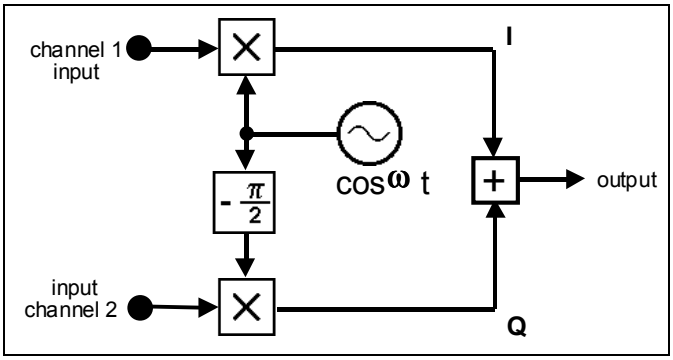


Figure 1: a quadrature modulator

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.

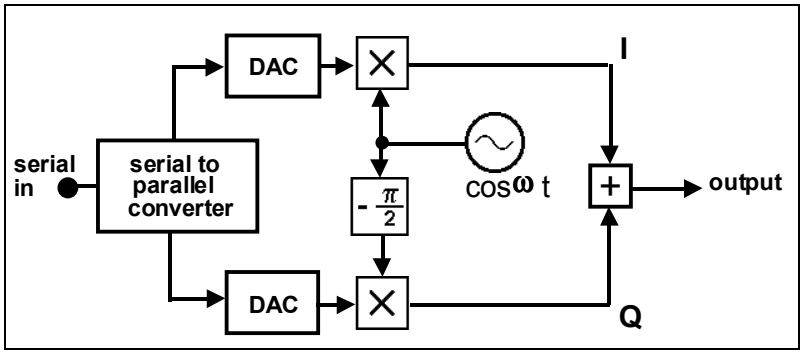


Figure 2: a QPSK modulator

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 describe 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 a 180^0 phase shift, given to the DSBSC.

Thus the signal in each arm is said to be undergoing a 180^0 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 idealised from in the block diagram of Figure 3.

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

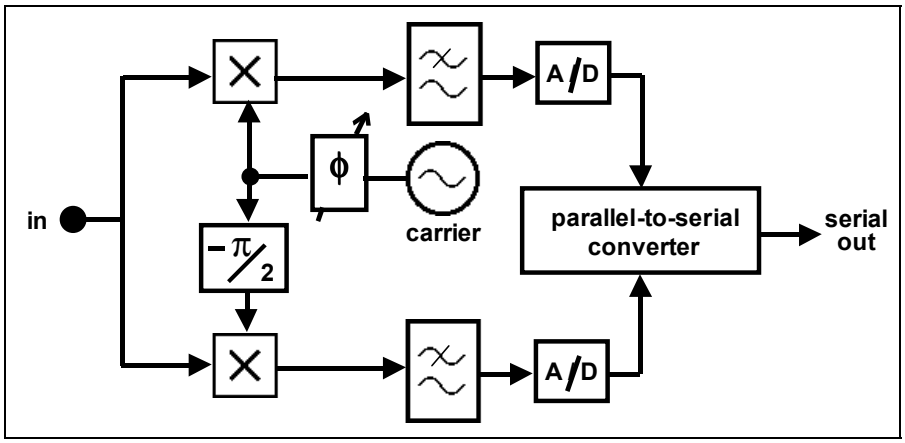


Figure 3: the QAM demodulator for QPSK

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.

For an experiment explicitly concerned with the acquisition of a carrier from such a signal, see the experiment entitled *The Costas loop* (Volume A2), and *Carrier acquisition* (in Volume D2).

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.

Lowpass filter bandlimiting 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.

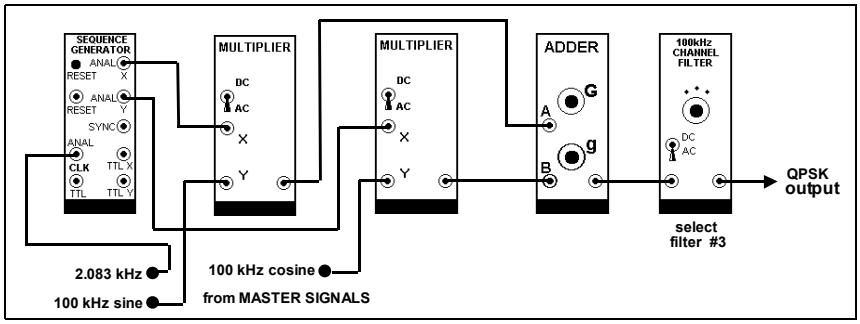


Figure 4: the QAM modulator for QPSK

The QAM modulator involves analog circuitry. Overload must be avoided, to prevent crosstalk between channels when they share a common path - the ADDER and output filter. In practice there would probably be a filter in the message path to each multiplier. Although these filters would be included for pulse shaping and/or band limiting, a secondary purpose is to eliminate as many unwanted components at the multiplier (modulator) input as possible. See Tutorial Question Q7.

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. See Tutorial Question Q5.*

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 centre of the screen; then connect the two data streams entering the QAM to the scope X and Y inputs.*

How would the display change if each of the data streams, presently non-bandlimited, was first passed through a bandlimiting filter ? Try this with the LPF in the HEADPHONE AMPLIFIER and a TUNEABLE LPF.

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.

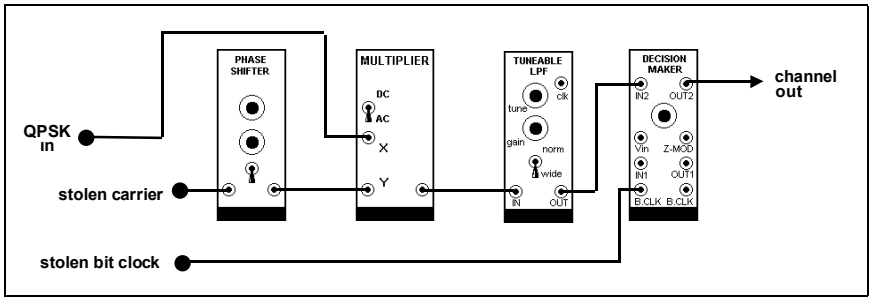


Figure 5: one channel of the demodulator

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, including the z-mod facility of the DECISION MAKER. Use an eye pattern to locate the optimum decision point.*

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.*

TUTORIAL QUESTIONS

Q1 *explain how a QAM system conserves bandwidth.*

Q2 *how would you measure the phase between two DSBSC ? Would a basic PHASE METER, which is used for indicating the phase between two sinewaves, be of any help ?*

Q3 *the modulator used the quadrature 100 kHz outputs from the MASTER SIGNALS module. Did it matter if these were not* ***precisely*** *in quadrature ? Explain.*

Q4 *the demodulator did not rely on the phasing of the 100 kHz quadrature outputs from the MASTER SIGNALS module, but instead required some means of phase adjustment of the carriers into* ***both*** *MULTIPLIER modules. Explain.*

Q5 *in the modulator, if each signal at the ADDER output is 1 volt peak, what will be the peak amplitude of their sum ?*

Q6 *name one advantage of making the bit rate a sub-multiple of the carrier frequency.*

Q7 *why is there a need to eliminate as many unwanted components as possible into the modulator ?*

MULTI-LEVEL QAM & PSK

MULTI-LEVEL QAM & PSK

ACHIEVEMENTS: *demodulation of various m-QAM signals, including 16-PSK and 16-QAM. Detection of errors.*

PREREQUISITES: *completion of the experiment entitled* ***Signal constellations*** *in volume D1. Completion of the experiment entitled* ***QAM and 4-PSK*** *(this Volume) would be an advantage.*

ADVANCED MODULES: *M-LEVEL DECODER, ERROR COUNTING UTILITIES, NOISE GENERATOR, WIDEBAND TRUE RMS METER.*

EXTRA MODULES*: a total of two MULTIPLIER, two TUNEABLE LPF, and two ADDER modules.*

TRUNKS*: see your Laboratory Manager for details of signals at TRUNKS*

PREPARATION

This experiment is about the use of the quadrature amplitude demodulator for the demodulation of m-QAM and m-PSK signals.

A modulator which produces quadrature amplitude modulated (QAM) signals is examined in the analog experiment entitled *Phase division multiplex* (Volume A2), and a digital experiment entitled *QAM and 4-PSK* (this Volume). The more general m-QAM and m-PSK signals will be examined in this experiment.

the m-QAM generator

Figure 1 shows the block diagram of a multi-level encoder and a quadrature modulator. The first of these has been examined in the experiment entitled *Signal constellations* (Volume D1), which, it is assumed, you have completed.

That experiment also examined the multi-level decoder.

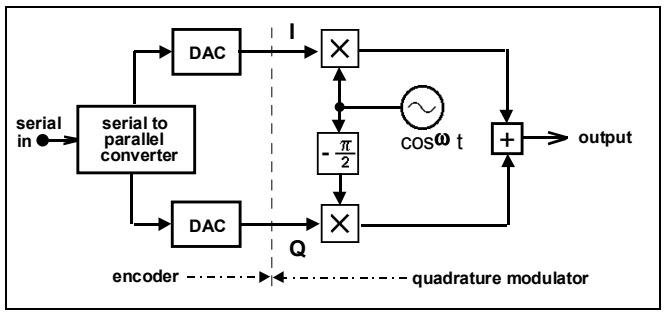


Figure 1: a multi-level QAM modulator

the m-QAM demodulator

Figure 2 shows the block diagram of a quadrature demodulator and a multi-level decoder.

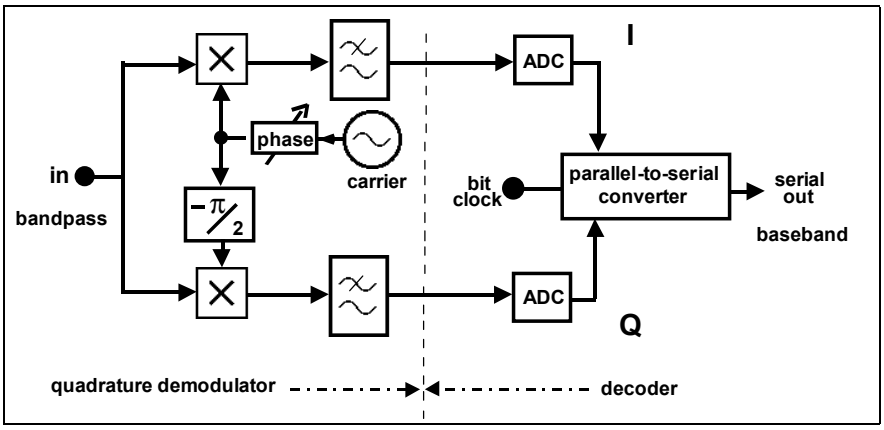


Figure 2: an m-QAM demodulator

carrier & bit clocks

Stolen carriers and bit clocks will be used at the demodulator.

As usual, while pursuing a major topic, one does not wish to become sidetracked by having to model carrier and bit clock recovery schemes. These have already been examined in the experiments entitled *Bit clock regeneration,* and *Carrier acquisition* (both in this Volume), and *Carrier acquisition and the PLL,* and *The Costas loop* (both in Volume A2).

carrier phasing

The local (quadrature) carriers to the two MULTIPLIER modules will be obtained from the MASTER SIGNALS module. These two will be connected *with the same relative phasing* as those at the transmitter. This will save much time in investigating the many phase combinations possible at the receiver, only one of which will satisfy the condition for error-free decoding.

The only phase adjustment will be introduced by a single PHASE SHIFTER inserted in series with the bandpass input to the modulator [1]. The PHASE SHIFTER module is a *narrow band* device, introducing different amounts of phase at different frequencies. However, the QAM signal itself is also narrow band, and so this use of the PHASE SHIFTER is not entirely inappropriate. See Tutorial Question Q2.

Once error free decoding has been achieved the correct conditions at the demodulator are known, and the (approximate) single PHASE CHANGER scheme could be replaced with a precise method, namely a PHASE CHANGER in each carrier path. These two schemes are illustrated in block diagram form in Figure 3 below.

[1] this differs from the method shown in Figure 2.

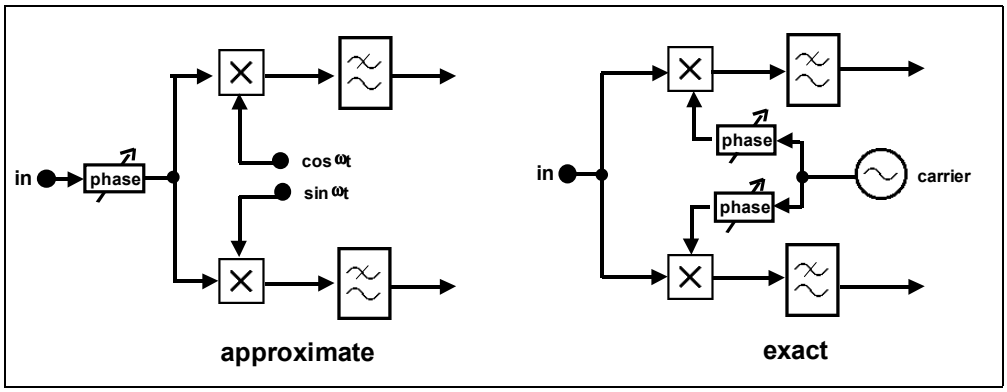


Figure 3: the two phasing schemes

decoder and error counting

The decoding is performed by the M-LEVEL DECODER module, acting on the **i** and **q** outputs from the demodulator. Correct alignment of the system, which requires carrier phase adjustment, is checked by counting errors.

The technique of error counting, using a reference sequence at the receiver, has been examined in detail in previous experiments [2]. In particular, and of direct relevance to *this* experiment, setting up for error free transmission using multi-level signalling and error detection instrumentation was described in the experiment entitled *Signal constellations* (Volume D1).

complexity

To build a complete system - encoder, modulator, noisy channel, demodulator, decoder and bit error rate measuring facilities - requires slots for more than 12 modules. The TIMS 301 rack provides 12 slots.

So in this experiment only the receiver will be modelled and examined. Several signals will be available at TRUNKS. These will be defined by your Laboratory Manager, and will be used for testing your receiver. All are based on 100 kHz carriers and 8.333 kHz bit clocks.

Their messages will be derived from SEQUENCE GENERATOR modules similar to your own, and of stated sequence lengths. So a reference message can be created at the receiver, and a sliding window correlator used for lining it up with the decoded sequence, as has been done in previous experiments.

It is unlikely that you could identify the signals at TRUNKS using only an oscilloscope. But you could identify them, using a lengthy trial and error process, by setting up the demodulator to be described and modelled in what follows. However, to save time, their descriptions, and locations at TRUNKS, will be defined by your Laboratory Manager.

[2] in detail in *BER measurement in the noisy channel* in this Volume.

the noisy channel

To save rack space it might be thought that the channel filtering could be included at the transmitter end of the system. But it is necessary to add noise at the *input* to the channel, so that the noise will suffer the same bandlimiting as the signal.

For an alternative method, see Tutorial Question Q3.

EXPERIMENT

The complete receiver is shown modelled in Figure 4 below.

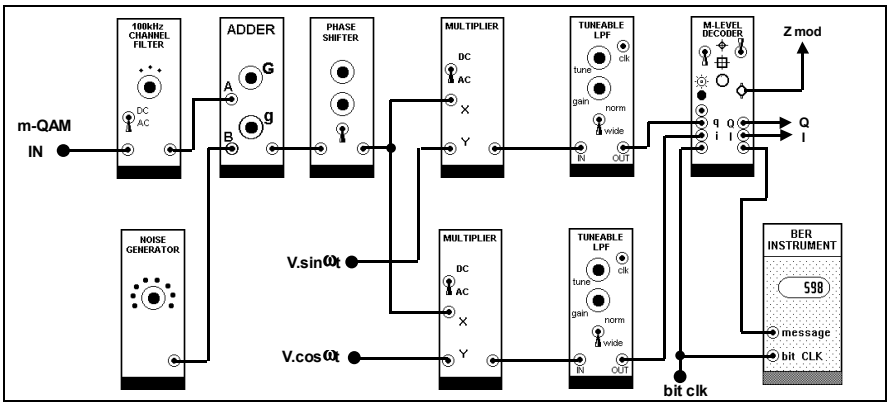


Figure 4: model of the m-QAM demodulator and decoder

The error counting instrumentation is shown as a BER INSTRUMENT module. This is described in the experiment entitled *BER instrumentation macro model* (this Volume).

T1 *patch up the system of Figure 4. Be sure to connect the cos and sin outputs from the MASTER SIGNALS module as shown, since this matches their usage at the transmitter, and so preserves relative carrier phasing.*

T2 *select the bandpass channel from the 100 kHz CHANNEL FILTERS module.*

T3 *select the 4-QAM signal from TRUNKS. Connect it to the input of your receiver.*

T4 *initially introduce no noise to the input ADDER. Adjust the gain through this ADDER to give a TIMS ANALOG REFERENCE LEVEL signal (4 volts peak-to-peak) into the quadrature demodulator.*

T5 *connect the Z-modulation output signal to the Z-modulation input of the oscilloscope.*

The eye pattern is first used to adjust the decision instant (displayed as a bright spot on the oscilloscope trace - vary the intensity of the display for best spot visibility). The front panel DECISION POINT control on the M-LEVEL DECODER allows adjustment over a little more than one clock period. If it cannot be located at the desired point, step it across by pressing the HUNT button. For 4-QAM the next press of the button will return to the previous condition (there are two clock periods per frame), but for 16-QAM (for example) there are four clock periods per frame, and so four accessible regions, each of which can be reached by repeated presses of the button.

Use the same techniques as are described in the experiment entitled *Detection with the DECISION MAKER* (Volume D1). See Tutorial Question Q1.

T6 *look at the eye pattern at either (LPF) output of the demodulator.*

Even though two carriers may be available in phase-quadrature, there are many combinations in which they may be connected to the two multipliers for desired results. Connection in the manner shown in the model of Figure 4 will eliminate many of these (these connections duplicate those at the transmitter).

Since the properties of the input signal are known, you already know how many levels to expect in the eye pattern. For 4-QAM there should be 2 levels (in each of the **i** and **q** paths). If there are more than 2 levels, what adjustments are available to you ?

Only one. Think about it, then proceed.

T7 *adjust the phasing of the demodulator PHASE CHANGER until there are the expected number of levels in the eye pattern (<u>two</u> levels for 4-QAM). There will be <u>four</u> phases which will give this result - two positions of the front panel variable COARSE control, and two with the 180^0 toggle switch. How many will be correct ? How do they differ ? This cannot be decided yet.*

T8 *accept any one of the four phases of the previous Task.*

T9 *using the gain controls of the TUNEABLE LPF modules, adjust the levels of the signals to the **i** and **q** inputs of the M-LEVEL DECODER to result in 0 to +5 volt signals at the **I** and **Q** outputs.*

T10 *set up the M-LEVEL DECODER front panel toggle switches to accept the coding scheme of the TRUNKS signal.*

T11 *attempt to align the DATA output from the M-LEVEL DECODER with the X-output from the reference SEQUENCE GENERATOR. If there are continuous errors, the only system error (?) must be the demodulator carrier phasing.*

T12 *if there are errors, adjust to another phase, and repeat the previous Task. One, and only one, of the four possible phases should give error free transmission.*

T13 *examine the signal constellation of the demodulated signal.*

So far only the 4-QAM format has been decoded.

T14 *examine the other signals at TRUNKS. In each case, before error free decoding can be assured, it will be necessary to:*

a) set the reference SEQUENCE GENERATOR to the same sequence length as that at the transmitter.

b) adjust the decision point on the eye diagram.

c) check for the correct number of levels in the eye diagram, and make a fine adjustment of the phasing if necessary.

d) repeat Tasks T7 to T13.

noise and BER measurement

The system is now ready for the addition of system impairments in the form of noise.

Follow the procedures described in detail in the experiment entitled *BER measurements in the noisy channel* (this Volume).

Before making these measurements, however, now is the time to fine tune the system. The effectiveness of these adjustments can be judged by aiming for an improvement of the bit error rate. They should be made with a 'moderate' amount of added noise.

Note that if only 12 slots are available (in a single TIMS 301 rack), and 13 are required, it is possible to share one slot between the WIDEBAND TRUE RMS METER and the ERROR COUNTING UTILITIES modules. These two are not required simultaneously.

T15 *trim the DC offsets at the* ***Q*** *and* ***I*** *outputs of the M-LEVEL DECODER module (first read the appropriate section of the* ***Advanced Modules User Manual*** *for details, and/or seek the advice of your Laboratory Manager. These signals are the actual inputs to the decoder circuitry).*

You are now ready for BER measurements. Detailed instructions are outside the scope of this introductory experiment.

further experiments

bandwidth

So far no mention has been made of the bandwidth of these signals. Further experiments, at the very least, should be devised to compare the inter-relationships between channel bandwidth, data rates, and error rates.

PCM

Instead of using a SEQUENCE GENERATOR as a message source, a PCM ENCODER could be used. This would allow the transmission of an analog message over the system.

TUTORIAL QUESTIONS

Q1 *there are two waveforms (**i** and **q**), and so two decision points, but the* DECISION POINT *control moves both points together. What does this say about the properties of the lowpass filters in the MULTIPLIER outputs ?*

Q2 *the method of carrier phase adjustment was made using a <u>single</u> PHASE CHANGER in the input signal path to the demodulator. Was this reasonable ? This method was implemented due to the shortage of free slots in a single TIMS 301 rack. The method illustrated in Figure 2 would require two slots, whereas the suggested method only one.*

Q3 *to save rack slots at the receiver the channel filter could be placed at the transmitter. Noise could still be added at the receiver input. This would save a single rack slot. If the channel filter bandwidth was wider than the lowpass filters at the demodulator output, would not this be an acceptable method of bandlimiting the noise ? Compare the two methods. What problems might there be in each ?*

SPREAD SPECTRUM - DSSS

SPREAD SPECTRUM - DSSS

ACHIEVEMENTS: *demonstration of some of the principles of a direct sequence spread spectrum (DSSS) system.*

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

ADVANCED MODULES: *DIGITAL UTILITIES, NOISE GENERATOR*

EXTRA MODULES*: a total of three SEQUENCE GENERATOR modules is required.*

RECOMMENDED INSTRUMENTATION: *some means of displaying the spectra of the signals to be examined would be an advantage; eg, the PICO Virtual Instrument, together with a PC.*

PREPARATION

the need

In some situations it is required that a communication signal be difficult to detect, and difficult to demodulate even when detected. Here the word 'detect' is used in the sense of 'to discover the presence of'. The signal is required to have a low probability of intercept - LPI.

In other situations a signal is required that is difficult to interfere with, or 'jam'.

The 'spread spectrum' signal has properties which help to achieve these ends.

Spread spectrum signals may be divided into two main groups - direct sequence spread spectrum (DSSS), and frequency hopping spread spectrum (FHSS). This experiment is concerned with demonstrating some of the principles of the first.

principle of DSSS

Consider the frequency translation of a baseband message (of bandwidth B Hz) to a higher part of the spectrum, using DSBSC modulation. The resulting signal occupies a bandwidth of 2B Hz, and would typically override the noise occupying the same

part of the spectrum. This makes it easy to find with a spectrum analyser (for example), and so the probability of intercept is high. A local carrier, synchronized with that at the transmitter, is required at the receiver for synchronous demodulation. The recovered signal-to-noise ratio is 3 dB better than that measured at its original location in the spectrum. This 3 dB improvement comes from the fact that the contributions from each sideband add coherently, whereas the noise does not. This can be called a 3 dB 'processing gain', and is related to the fact that the transmission bandwidth and message bandwidth are in the ratio of 2:1.

In a spread spectrum system literally thousands of different carriers are used, to generate thousands of DSBSC signals each derived from the same message. These carriers are spread over a wide bandwidth (much wider than 2B Hz), and so the resulting DSBSC signals will be spread over the same bandwidth.

If the total transmitted power is similar to that of the single DSBSC case, then the power of an individual DSBSC in the spread spectrum case is thousands of times less. In fact, over the bandwidth occupied by one of these DSBSC signals, it would be literally 'buried in the noise', and difficult to find with a spectrum analyser (for example).

Instead of the total transmitted power being concentrated in a band of width 2B Hz, the multiple carriers have spread it thinly over a very wide bandwidth.

The signal-to-noise ratio for each DSBSC is very low (well below 0 dB).

To recover the message from the transmitted spread spectrum signal all that a receiver requires is thousands of local carriers, at the same frequency and of the same relative phase, as all those at the transmitter !

question: where do all these carriers come from ?

answer: from a pseudo random binary sequence (PRBS) generator.

Given a stable clock, and a long sequence, it may be shown that the spectrum of a pseudo random binary sequence generator is a good source of these carriers [1]. A second PRBS generator, of the same type, clocked at the same rate, and appropriately *aligned*, is sufficient to regenerate all the required local carriers at the receiver demodulator.

In the spread spectrum context the PRBS signal is generally called a PN - pseudo noise - signal, since its spectrum approaches that of random noise.

Having the correct sequence at the receiver means that the message contributions from each of the thousands of minute DSBSC signals combine in phase - coherently - and add up to a finite message output. Otherwise they add with random phases, resulting in a (very) small, noise-like output.

The key to a successful message recovery is the knowledge of the PN sequence used at the transmitter.

[1] the PRBS is a periodic signal

processing gain

To achieve most of the claims made for the spread spectrum it is necessary that the bandwidth over which the message is spread be very much greater than the bandwidth of the message itself. Each DSBSC of the DSSS signal is at a level below the noise, but each is processed by the synchronous demodulator to give a 3 dB SNR improvement. The total improvement is proportional to the number of individual DSBSC components. In fact the *processing gain* of the system is equal to the ratio of DSSS bandwidth to message bandwidth.

a DSSS generator

To generate a spread spectrum signal one requires:

1. a modulated signal somewhere in the RF spectrum
2. a PN sequence to spread it

These two are combined as shown in Figure 1.

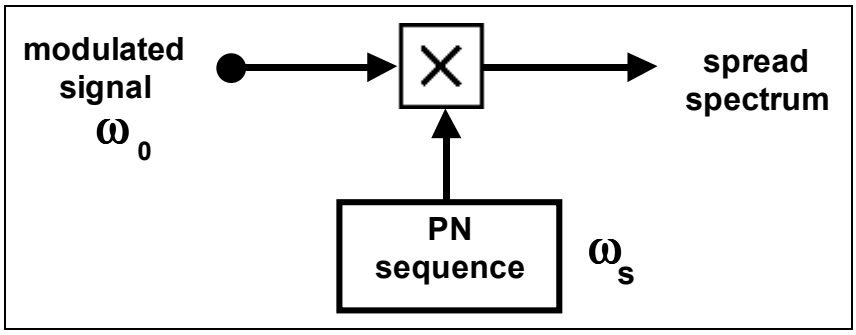


Figure 1: basis of spread spectrum

There are two bandwidths involved here: that of the modulated signal, and the spreading sequence. The first will be very much less than the second. The output spread spectrum signal will be spread either side of the original RF carrier (ω_0) by an amount equal to the bandwidth of the PN sequence.

Most of the energy of the sequence will lie in the range DC to ω_s, where ω_s is the sequence clock. The longer the sequence the more spectral components will lie in this range. It is necessary and usual that $\omega_0 >> \omega_s$, although in the experiment to follow the difference will not be large.

The modulated signal can be of any type, but typically digitally-derived, such as binary phase shift keyed - BPSK. In this case the arrangement of Figure 1 can be expanded to that of Figure 2.

A digital message is preferred in an operational spread spectrum system, since it makes the task of the eavesdropper even more difficult [2].

[2] it also offers some simplifications at the transmitter

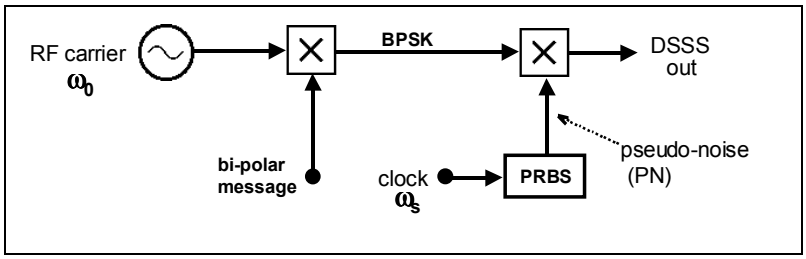


Figure 2: a spread BPSK signal

The arrangement of Figure 2 can be simplified by noting that, if the clock of the bi-polar message is a sub-multiple of the clock of the PN sequence, then the modulo-two sum of the message and the PN sequence can be used to multiply the RF carrier, generating a DSSS signal with a single multiplier. Such a simplification will not be implemented in this experiment.

a DSSS demodulator

A demodulator for the DSSS of Figure 1 is shown in block form in Figure 3.

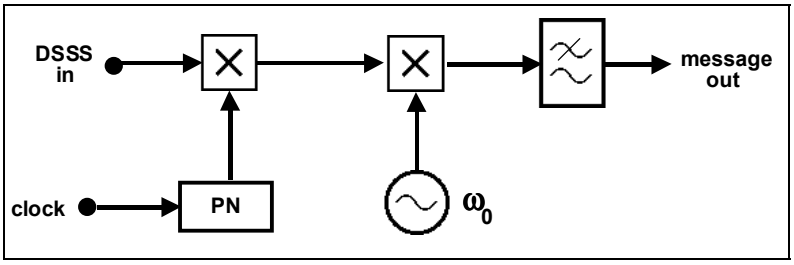


Figure 3: DSSS demodulator

The input multiplier performs the de-spreading of the received signal, and the second multiplier translates the modulated signal down to baseband. The filter output would probably require further processing - not shown - to 'clean up' the waveform to binary format.

The PN sequence at the receiver acts as a 'key' to the transmission. It must not only have the same clock and bit pattern; it must be *aligned* properly with the sequence at the transmitter.

sequence alignment

Sequence alignment is examined in the experiment entitled *PRBS Generation* (in Volume D1). Alignment of the two generators of Figures 2 and 3 is a simple matter if there is no delay introduced by the transmission path (not likely in practice, but simple in the laboratory). When delay is included, as in a practical system, alignment is a non-trivial exercise. It will not be examined in this experiment.

Also not shown are means of acquiring the carrier ω_0. A stolen carrier will be used in the experiment.

code division multiple access CDMA

When there is no DSSS signal present at the input to the demodulator there is none-the-less a noise output.

Should there be, in addition, a DSSS signal present, but derived from a PN sequence other than that being used by the demodulator, this will also produce a noise output, since such a signal (it can be shown) looks like random noise. However, if it was derived from the same sequence as that being used at the demodulator, and this sequence was correctly aligned, then instead of a noise output the message would appear.

Since the transmission medium - free space in practice - is linear (meaning that superposition applies) more than one DSSS can occupy the same spectrum space. The required message can be selected by choosing the appropriate sequence at the demodulator. All the others appear as noise.

Such an arrangement is an example of a code division multiple access (CDMA) spread spectrum system.

Currently under development is a pair of TIMS modules which will enable a demonstration of CDMA with fewer modules than would be required if the transmitter of this experiment was duplicated. Enquiries to tim@tims.com.au

the PN spectrum

The PN signal, being periodic, has a line spectrum. This spectrum is determined by the PN clock period T_c and the sequence length N (the number of bits, or clock periods, before the pattern repeats).

- the spectral lines are separated by $(1/NT_c)$ Hz.
- there is a DC component of amplitude (1/N).
- the amplitude of an individual line in the spectrum is weighted, where:

$$\text{weight} = \sqrt{\frac{2(N+1)}{N^2}\left(\frac{\sin(\pi n/N)}{\pi n/N}\right)}, \quad n \neq 0 \qquad \text{..... 1}$$

It is clear that a plot of these weights will show them lying within an envelope having a sync function shape. Most of the energy of the PN sequence lies below the first minimum (when n = N); that is, below the clock frequency.

For approximate analysis it is often assumed that the shape of the power spectral density is rectangular, extending from DC to $(1/T_c)$ Hz.

spectral analysis

Some important measurements to be made with a spread spectrum system are of the various spectra. A simple wave analyser, as described in the experiment entitled *Spectrum analysis - the WAVE ANALYSER*, in Volume A2, is not suitable for the purpose.

A *PICO Virtual Instrument* SPECTRUM ANALYSER is recommended.

EXPERIMENT

a DSSS model

This experiment will be concerned with modelling the systems of Figures 2 and 3.

the message

The message comes from a SEQUENCE GENERATOR.

To obtain a reasonable processing gain the message clock needs to be much slower than the PN clock. Being a sub-multiple of the PN clock is also an advantage (see Tutorial Question Q8). The2 kHz MESSAGE from the MASTER SIGNALS module has been used - 1/48 of the 100 kHz master clock.. You may prefer a larger division ratio. This can be achieved with further division using the DIGITAL UTILITIES module.

Select a short message sequence for stable oscilloscope displays (both toggles of the on-board switch UP).

the transmission medium

The transmitter is connected to the receiver via an ADDER, acting as a non-bandlimiting (and so no delay) transmission path. The second input to the ADDER will be used for inserting noise. The inclusion of a finite delay would introduce problems with aligning the receiver PN sequence.

clocks

Since the PN clock is a sub-multiple of the carrier, only one of these needs to be recovered by the receiver. In the experiment they are stolen from the transmitter.

generation

T1 *model the block diagram of Figure 2. This is shown in Figure 4. The ADDER is included for inserting noise from a NOISE GENERATOR (module not shown).*

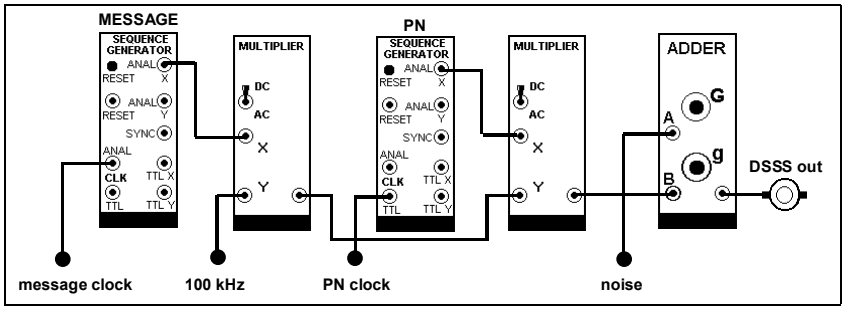


Figure 4: the transmitter model

T2 *before inserting the SEQUENCE GENERATOR modules, select a short sequence for the message (both toggles of the on-board switch SW2 UP), and the same long sequence for the PN generators (both toggles of the on-board switch SW2 DOWN).*

T3 *initially use the 100 kHz TTL available from MASTER SIGNALS, divided by 12, using a DIGITAL UTILITIES module (not shown), for the PN generator clock.*

T4 *initially reduce the noise output from the ADDER to zero.*

The next three Tasks could be omitted, but they confirm that both MULTIPLIER modules are operating as expected.

T5 *instead of connecting the bi-polar message sequence to the X input of the first MULTIPLIER, connect instead the 2 kHz MESSAGE (sinusoidal) signal. This makes the output from the first MULTIPLIER a DSBSC signal, easily recognisable on the oscilloscope. Check this.*

T6 *instead of connecting the PN sequence to the X input of the second MULTIPLIER, connect instead the VARIABLE DC module set to near +2 volt. This makes the second MULTIPLIER a voltage controlled amplifier with a gain of about unity. Thus the 'DSSS output' will be a well-recognisable DSBSC based on a 2 kHz message. Check your levels with this recognisable signal.*

T7 *using the PICO SPECTRUM ANALYSER, examine the output spectrum. Confirm it is a DSBSC.*

When satisfied that the MULTIPLIER modules are behaving as expected, return their inputs to the signals previously connected.

T8 *synchronize the oscilloscope to the SYNCH signal (START-OF-SEQUENCE) of the message generator. Examine signals throughout the system. Some will be familiar, others not. There are no adjustments to be made, except for the output amplitude from the ADDER.*

T9 *using the PICO SPECTRUM ANALYSER, examine the output spectrum. With an 8.333 kHz PN clock, confirm that the output spectrum - the DSSS signal - has its energy concentrated over about 8 kHz either side of the 100 kHz carrier.*

T10 *now add noise. Adjust the noise level so that, while observing the spectrum of the ADDER output, the DSSS signal can be seen above the noise level.*

T11 *while still observing the spectrum, increase the spread of the DSSS signal. This is done by increasing the PN sequence clock rate by choosing a lower division of the 100 kHz TTL - choose divide-by-2, for a 50 kHz clock.*

The increase of PN clock rate has widened the spectrum of the PN sequence to about 50 kHz (from 8 kHz). Since the DSSS signal contains the same energy as before, it has been spread more thinly over the spectrum, and it will have sunk deeper into, and got 'lost' in, the noise.

This is one of the main purposes of spread spectrum.

demodulation

T12 *model the receiver of Figure 3 as suggested in Figure 5 below. Both the 100 kHz carrier, and the PN sequence, are stolen from the transmitter. Not shown is a PHASE SHIFTER for the 100 kHz carrier. This is used to maximize the output amplitude (it will also change its polarity).*

T13 *the bandwidth of the output filter is chosen to suit the message. Use a TUNEABLE LPF (shown in Figure 5), or the 3 kHz LPF in the HEADPHONE AMPLIFIER. For restoration of the output to a TTL format a DECISION MAKER would be included, but this is not necessary for this experiment. Visual comparison of the sent and received sequences is adequate.*

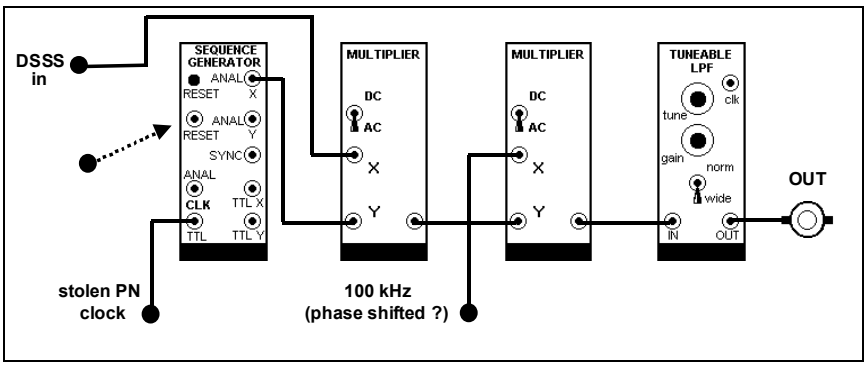


Figure 5: the receiver model

Although there are two stolen clocks shown, in practice it is often only necessary to acquire, by what ever means, a single clock. This is because one can be a known sub-multiple of the other.

T14 *observe the output, when the transmitter is connected to the input. Probably there will be 'nothing' - or nothing resembling the expected output sequence. Varying the phase of the 100 kHz carrier should not change things.*

The problem is that the receiver PN sequence, although synchronized with that at the transmitter, is not correctly aligned in time. With *no transmission delay* it is a simple matter to achieve this.

T15 *bring the two sequences into alignment by momentarily connecting the start-of-sequence SYNC output of the transmitter SEQUENCE GENERATOR to the RESET input of the receiver SEQUENCE GENERATOR.*

T16 *re-examine the output from the demodulator. The message should have been recovered (being a short sequence, this is easy to confirm visually). Adjust the bandwidth of the demodulator output filter for minimum bandwidth consistent with reasonable waveshape. Remember, a DECISION MAKER could be used to regenerate a perfect copy of the original, but this is not necessary for our present purpose.*

interference

T17 *with the system set up and showing the demodulated sequence at the receiver output, replace the noise with a 100 kHz sinusoid from a VCO. This represents an interfering signal (a very elementary form of jamming). Monitor the VCO with the FREQUENCY COUNTER.*

T18 *while watching the demodulator output, sweep the VCO frequency through its full frequency range.*

the wanted sequence will still be present at the demodulator output, but there will be negligible sign of the effects of the interfering signal. You have demonstrated another important property of spread spectrum.

CDMA

It was seen earlier that, when the 'incorrect' PN sequence was used at the receiver, there was negligible output from the receiver even when a DSSS signal was present at the input. In fact, it was the correct sequence, but it was mis-aligned. The same result would have been observed if the sequence had been changed.

In fact, many DSSS signals can be present at the same time, on the same carrier or otherwise, and will appear to the receiver as noise. Only when using the correct PN sequence (correctly aligned) will a message appear at the output.

The required message is recovered by using the correct code - so this is code division multiplex, or code division multiple access - CDMA.

CDMA can be demonstrated with TIMS by modelling two or more transmitters, with different PN codes for each, and adding their outputs. This becomes a big system, but it is possible.

As stated earlier, two new modules are being developed for TIMS which will simplify such a demonstration. Enquiries to tim@tims.com.au

TUTORIAL QUESTIONS

Q1 *sketch the amplitude spectrum of a PN sequence clocked at 50 kHz, and with a length of 2047 bits. Annotate with as much numerical information as you can.*

Q2 *consider a DSBSC signal derived from a single tone. How many lines would there be in the spectrum of the spread signal ? You will have to supply some data regarding the spreading sequence.*

Q3 *explain the principle difference between the channel selection process in phase division multiplex (PDM), and code division multiple access (CDMA). Thus explain why PDM can only support two channels, whereas this restriction is not present in CDMA.*

Q4 *unlike the two channel [3] DSBSC multiplexing system examined in the experiment entitled* ***Phase division multiplex*** *(in Volume A2), CDMA is capable of supporting more than two independent channels. What do you see as a limit to the number of channels which could be accommodated in a code division multiple access (CDMA) spread spectrum system ?*

Q5 *consider a DSBSC signal derived from a single tone. How many lines would there be in the de-spread spectrum ? You will have to supply some data regarding the spreading sequence.*

Q6 *the message bandwidth is typically much less than that of the spreading PN sequence. However there is an advantage in making it at least as wide as the spacing of the spectral lines of the latter. Why might this be ?*

Q7 *what have you read about the effects of multi-path fading on a DSSS signal ?*

Q8 *what advantage is there in making the message bit rate a sub-multiple of the PN bit rate ?*

[3] ie, two *message* channels

this page is blank

this page is blank

DIGITAL UTILITY SUB-SYSTEMS

DIGITAL UTILITY SUB-SYSTEMS

ACHIEVEMENTS: *an awareness of various sub-systems included within modules to which independent access may be made.*

PREREQUISITES: *none.*

ADVANCED MODULES: *various ! See below.*

INTRODUCTION

Many modules in the *Advanced Modules* set have sub-systems within them to help them perform their main function. Many of these modules allow direct access to these sub-systems. So they may be used independently in other models.

It is often forgotten that these sub-systems are available. This chapter serves as a reminder of their existence, and gives brief details of their properties.

bandpass filters

There are two identical but independent analog bandpass filters, BPF1 and BPF2, built into the BIT CLOCK REGEN module.

They are order 4 Chebyshev bandpass filters. They have mid-band gains of unity.

A clock signal to each filter may be supplied internally or externally (to the *EXT CLK* socket), as determined by an on-board switch SW1.

- when the **int**ernal clock is selected the response is centred on 2.083 kHz, with -3 dB bandwidths of about 100 Hz, and -40 dB bandwidths of about 700 Hz.
- when an **ext**ernal TTL clock is selected the centre frequency may be tuned over the range 1 kHz to 5 kHz. The clock frequency should be 50 times the desired centre frequency.

The clock source selections are made with SW1, according to the settings given in the Table below.

SW1-1	SW1-2	BPF1 clock	BPF2 clock
DOWN	DOWN	EXT	EXT
DOWN	UP	EXT	INT
UP	DOWN	INT	EXT
UP	UP	INT	INT

BPF operation in terms of clock source

There is only one EXT CLK input provided. Thus, when BPF1 and BPF2 both have external clock selected by SW1, they both receive the same clock.

reminder: the 2.083 kHz (which is 1/48 of the TIMS 100 kHz MASTER CLOCK), is a common bit rate for many experiments.

digital delay

This sub-system is built into the INTEGRATE & DUMP module.

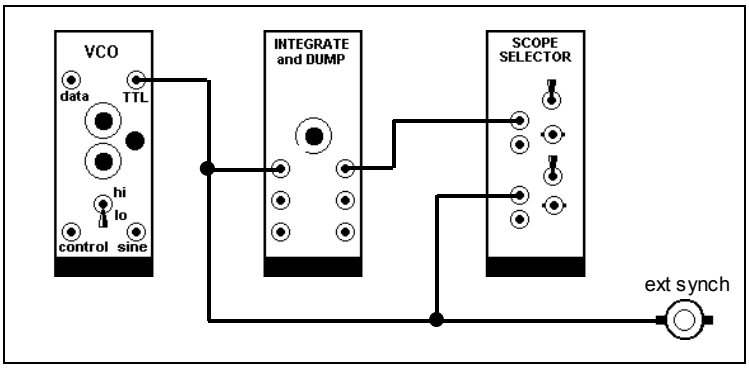
The input is a TTL clock signal. The output is a train of pulses of the same repetition rate, but of width about 10 μsec (fixed). The operating frequency of the clock should be below 15 kHz.

The delay is adjustable by a front panel control DELAY, in conjunction with a toggle switch SW3 mounted on the circuit board. The delays to be expected are shown in the table below.

SW3-upper toggle	*SW3-lower toggle*	*delay range from front panel, using DELAY*
RIGHT	RIGHT	10 μsec - 100 μsec
RIGHT	LEFT	60 μsec - 500 μsec
LEFT	RIGHT	100 μsec - 1 msec
LEFT	LEFT	150 μsec - 1.500 msec

on-board switch SW3 settings

You can check the delay between the input and output TTL pulseforms by a scheme such as shown in the Figure below.



demonstration of digital delay

The display on CH1-A will be of the input TTL clock, obtained from a VCO in the LO frequency range.

The display on CH2-A will be a train of 10μsec wide pulses, synchronized to the input clock. The relative position of these two TTL pulse trains may be varied with the front panel control of the INTEGRATE & DUMP module (in conjunction with the switch settings shown in the Table above).

digital divide-by-1, 2, 4, or 8

This sub-system is built into the BIT CLOCK REGEN module.

The frequency of the input TTL signal must be held below 15 kHz.

Division by a factor of 1, 2, 4, and 8 are selected by the on-board switch SW2, according to the scheme of the Table below.

Note that the 'divide by 1' option introduces an inversion.

SW2-A (left)	*SW2-B (right)*	*divide by*
DOWN	DOWN	8
DOWN	UP	4
UP	DOWN	2
UP	UP	-1

switch selectable division ratios

digital divide-by-2, 3, 4

The DIGITAL UTILITIES module has two each of TTL divide-by-2, divide-by-3, and divide-by 4 sub-systems.

It also has a TTL divide-by-minus 1, which offers a polarity inversion.

digital divide-by-4

This sub-system is built into the LINE-CODE ENCODER module.

The input must be TTL, as is the output. The output has a mark-space ratio of 1:1.

Accepts TTL signals anywhere within the TIMS range - from 100 kHz down.

digital divide-by-8

This sub-system is built into the CONVOLUT`L ENCODER module.

The input must be TTL, as is the output. The output has a mark-space ratio of 1:1 for any mark-space ratio of the input.

Accepts TTL signals anywhere within the TIMS range - from 100 kHz down.

digital inversion

This is available in the DIGITAL UTILITIES module.

exclusive-or

This sub-system is built into the ERROR COUNTING UTILITIES module.

It accepts two TTL input signals whose logical X-OR (EXCLUSIVE-OR) sum is required. The output depends on which of the two available modes is in use.

gated mode

Requires a TTL clock. The logic result is computed and presented at the output only during the HI of the clock pulse. For precise timing details it is best to make some measurements on the logic operation under the conditions in which you are interested ! When used in company with the pulse counting facility (a second sub-system within the module) these details are taken care of automatically.

normal mode

If no clock signal is provided, then the output is a continuous result of the X-OR operation.

frequency doubler

There is a transition detector in the BIT CLOCK REGEN module. It is optimised to work in the region of 2 kHz. For an input TTL signal it gives an adjustable width TTL output pulse for every input logic transition.

Thus, for a rectangular input at 2 kHz the output is rectangular at 4 kHz.

The width of the output pulse can be adjusted with the on-board variable resistor RV1 -VARY PULSE WIDTH - provided the jumper J1 is in the VARY position..

integrate and dump

Two identical integrate-and-dump (I&D) sub-systems are built into the INTEGRATE & DUMP module.

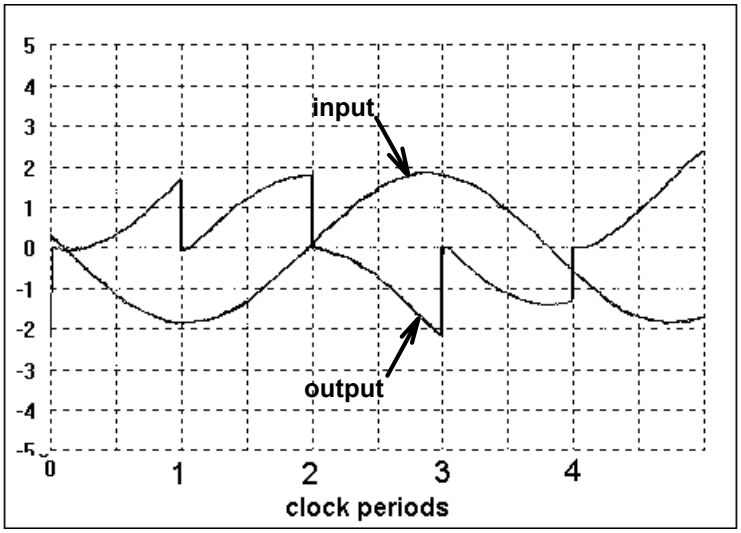
Each sub-system is driven by the same TTL clock, at a rate less than 15 kHz.

The input to each sub-system is an analog waveform.

Although the sub-system inputs and outputs are labelled I&D 1 and I&D 2 (they identify with the name of the module), they represent the input and output sockets of sub-system #1 and sub-system #2. The integrate and dump operation is obtained with on-board rotary switches SW1 and SW2 respectively (position '5').

The integration starts after a small delay, on the rising edge of the clock pulse. It is continuously available at the output until the end of the clock pulse, when it is dumped and a new integration period commences. *Note that the integrator inverts.*

This is illustrated in the Figure below.



integrate and dump of a sine wave (the integrator inverts)

The time constants of the integrators are:

integrator	*R*	*C*	*comments*
1	R7 = 330 kohm	C4 = 470 pF	fixed
2 - option 1	R26 = 330 kohm	C34 = 470 pF	J1 open
2 - option 2	R26 = 330 kohm	C34+C44 = 940 pF (jumper at 'IN')	J1 shorted adds C44 to C34

The jumper J1 is on the circuit board.

integrate and hold

Two identical integrate-and-hold sub-systems are built into the INTEGRATE & DUMP module.

Each sub-system is driven by the same TTL clock, at a rate less than 15 kHz.

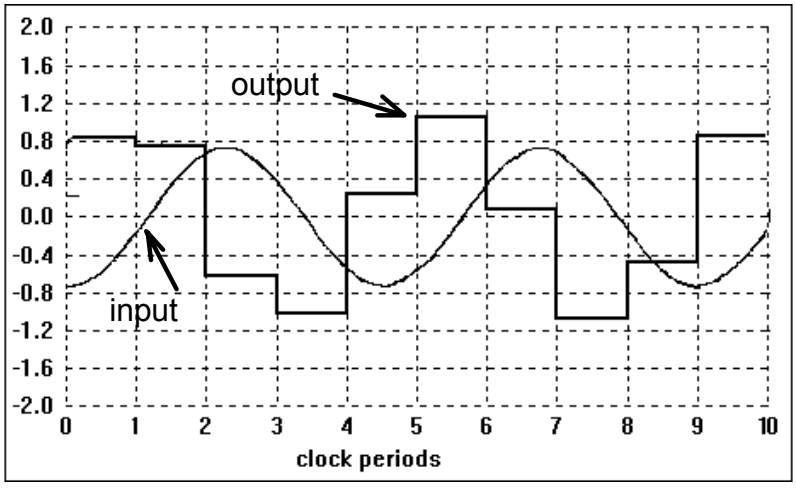
The input to each sub-system is an analog waveform.

Although the sub-system inputs and outputs are labelled I&D 1 and I&D 2 (they identify with the name of the module), they represent the input and output sockets of sub-system #1 and sub-system #2. The integrate and hold (I&H) operation is obtained with on-board rotary switches SW1 and SW2 respectively (position '2' or '3').

operation: the input is integrated over the period of the clock. Integration commences at a rising edge of the clock pulse, and ceases at the next rising edge. At the end of the period the result is presented at the output, and held there for the next clock period. At the same time the integrator is re-set, and the cycle is repeated.

A pulse of fixed width (about 10 µs) at the READY output occurs some time after the rising edge of the clock, to indicate that the output has been updated and has settled.

See the Figure below, for the case of a sinusoidal input, synchronous with the clock. This is the sort of display you can observe if you use the 2 kHz MESSAGE and the 8.333 kHz SAMPLE CLOCK signals from the MASTER SIGNALS module for the input and clock respectively. *Note that the integrator inverts.*



integrate and hold of a sine wave (the integrator inverts)

The time constants of the integrators are:

integrator	*R*	*C*	*comments*
1	R7 = 330 kohm	C4 = 470 pF	fixed
2 - option 1	R26 = 330 kohm	C34 = 470 pF	J1 open
2 - option 2	R26 = 330 kohm	C34+C44 = 940 pF (jumper at 'IN')	J1 shorted adds C44 to C34

The jumper J1 is on the circuit board.

limiter - 1

This sub-system is built into the DELTA MODULATION UTILITIES module.

It accepts analog input (it will saturate with inputs in excess of the TIMS ANALOG REFERENCE LEVEL of ±2 V).

The output is a TTL HI for inputs above 10 mV, and a TTL low otherwise.

This is a COMPARATOR, with respect to 10 mV. It makes a convenient 'analog-to-TTL converter' (eg, sine to TTL).

limiter - 2

There are two hard limiters in the FM UTILITIES module.

limiter - 3

This is part of the (analog) UTILITIES module. It may be set to soft limit as well.

pulse width modulator - PWM

This sub-system is built into the INTEGRATE & DUMP module. It is the subject of the experiment entitled *PWM and PPM* (in this Volume).

sample and hold

This sub-system is built into the INTEGRATE & DUMP module. It is the subject of the experiment entitled *Sampling with sample and hold* (in volume D1).

square TTL - 20 to 200 kHz

It is not always realised that the VCO can provide a TTL square wave anywhere in the range 500 Hz to 200 kHz. This is in the FSK mode.

By connecting either a TTL LO or a TTL HI to the DATA input socket, the output is either f_1 or f_2. The range of f_1 is changed with the on-board control RV7, and f_2 with RV8. Together with the front panel HI/LO toggle switch the VCO output (in FSK mode) can cover the range from below 500 Hz to well over 200 kHz.

An sinusoidal output is available simultaneously.

timed pulse

This sub-system is built into the ERROR COUNTING UTILITIES module.

The sub-system is driven by a TTL clock, from which it derives timing information.

default mode

On receipt of a momentary TTL HI at the TRIGGER input, or a push of the PUSH BUTTON, a TTL LO appears at the GATE output socket. It is otherwise HI.

This GATE LO remains for a preset number of clock cycles (10^3, 10^4, 10^5, or 10^6) as determined by the front panel four-position switch PULSE COUNT. But see 'gate time multiplier' below.

other modes

The above description of the TRIGGER and GATE pulses is what might be called the default mode. But an on-board toggle switch SW1 enables either or both of these states to be reversed.

usage

A typical usage is in an error counting situation, where the GATE pulse is used to initiate a counting operation. Hence the sub-system is part of the ERROR COUNTING UTILITIES module.

gate time multiplier

The front panel rotary switch indicates the number of clock cycles for which the GATE is open. But these can be altered by an on-board double pole switch SW2 and a jumper J1.

- ***normal mode:*** jumper J1 in norm position. Front panel multiplier is x1.
- ***extended mode*** jumper J1 in norm position. SW2 according to details written on the circuit board. Multipliers of x1 (default), x2, x4, and x8 are available.
- ***expanded mode*** jumper J1 in ÷12 position.. See the *TIMS Advanced Modules User Manual* for more details. This mode is applicable when using a 100 kHz bit rate.

TTL HI

Available from the DIGITAL UTILITIES and VARIABLE DC modules.

transition detector

This sub-system is built into the BIT CLOCK REGEN module.

It accepts as input any TTL signal likely to be found in the TIMS environment.

Its output is a positive TTL pulse for each transition (ie, in either direction) of the input.

The pulse width may be adjusted with the on-board VARY PULSE WIDTH variable resistor RV (with the on-board jumper J1 in the VARY position). To avoid anomalous operation the width must be less than that of the period of the bit clock.

The module has been optimized for operation at a bit clock of 2.083 kHz. This mode is selected with the on-board jumper J1 in the FIX position. In this case the pulse width is fixed at about half the period of the bit clock.

unit delay

An INTEGRATE & HOLD sub-system is available in the INTEGRATE & DUMP module.

This is a clocked sub-system.

When fed with a train of pulses, synchronous with the clock, this train is output one clock period later. A property of this particular sub-system is that the output is inverted in polarity.