Pulse Code Modulation PCM

An alternative to analogue pulse modulation where we modulate the amplitude, width, or position of a pulse train, the samples are quantified (i.e. quantisation). This means that each sample is given a discrete level where these discrete values are finite in contrast to an analogue signal which can assume an infinite number of values.

This process is feasible since for instance for the human ear, it is difficult to differentiate between signals with small variations in intensity.

NOTE... quantisation $\Rightarrow$data are now discrete in time and in amplitude as in Figure 1 which illustrates linear quantisation. This is in contrast to Pulse Amplitude Modulation, PAM, which is only **discrete in** time. Note that quantisation can be linear or non-linear.

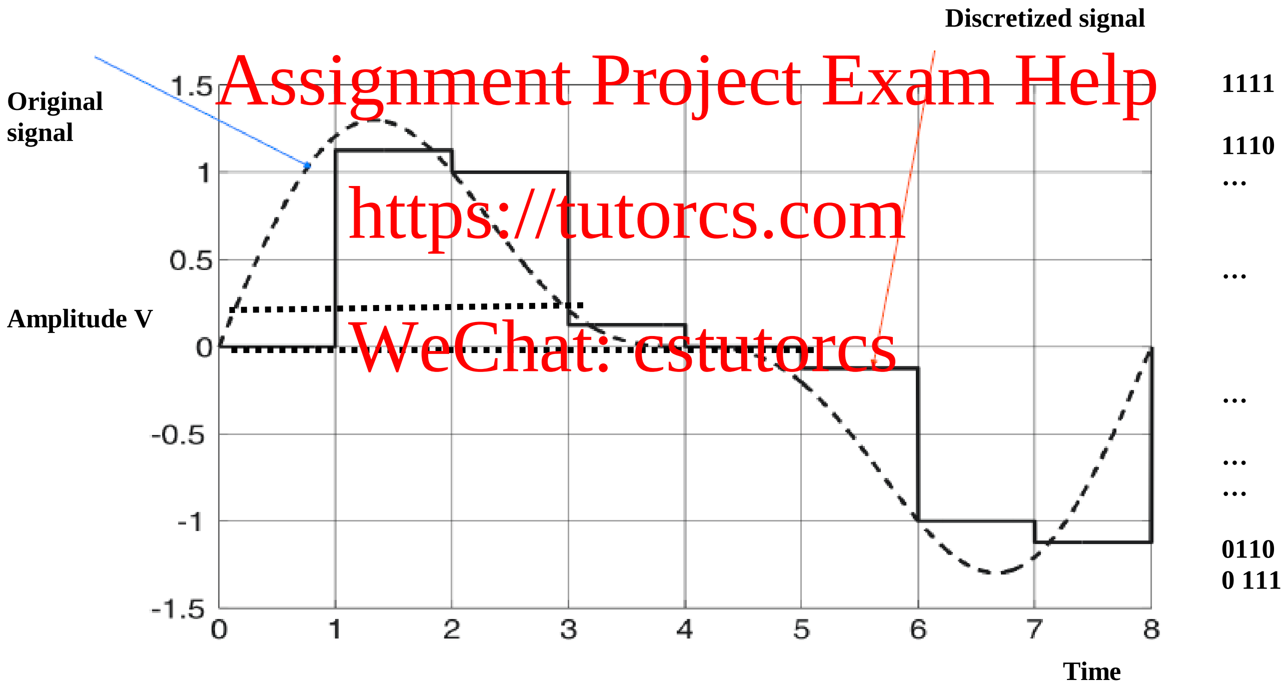


Figure 1. Illustration of the equal discretisation of an analogue signal

Non-linear quantisation: is used to accommodate wider variations in input signal. This permits better differentiation in small signal levels - for example in speech, variations in voltage are ~1: 1000 with high levels occurring less frequently. Thus it is more beneficial to allocate more levels for low signal voltages that occur more frequently in the signal. The difference between linear and non-linear quantisation is shown in Figure 2.a and 2.b, respectively. **NOTE that** Quantisation produces errors which are related to the difference between the actual signal level and the quantum as shown by the value of q(x) in Figure 2 and w(t)-h(t) in Figure 3.

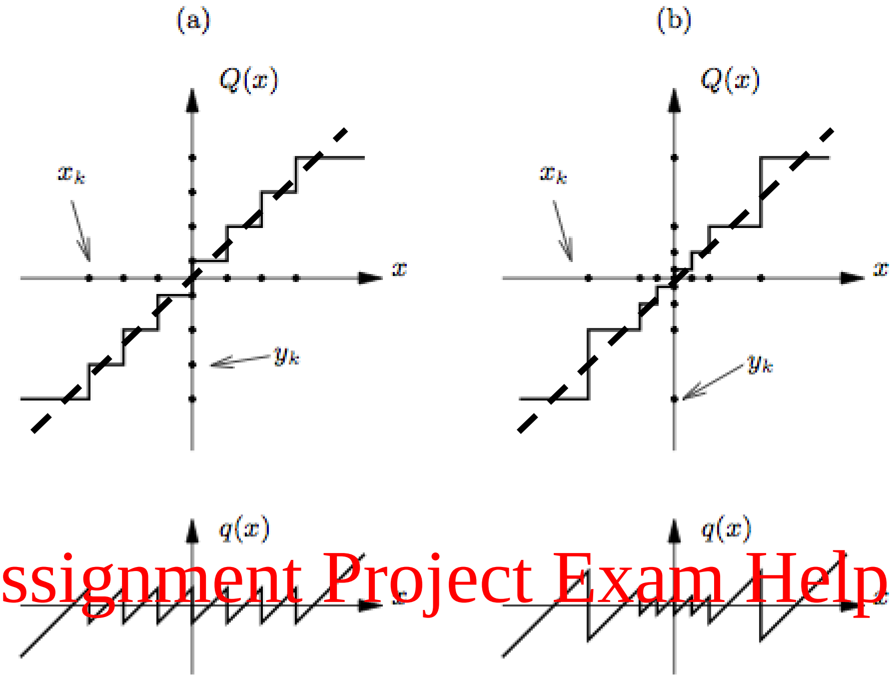


Figure 2. Quantisation levels (a) linear quantisation, (b) non-linear quantisation. Q(x) indicates quantisation noise which is the difference between the original signal (dashed line) and the quantisation level.

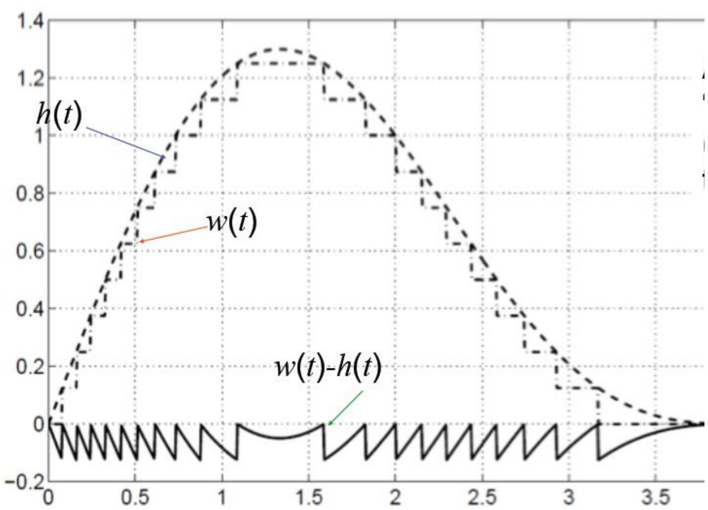


Figure 3 Quantisation error

Considering Figure 4 quantization of the signal is determined as follows.

If the input voltage Th2>$v_{in} \geq$ Th1, then v_{in} quantization level is level 2.

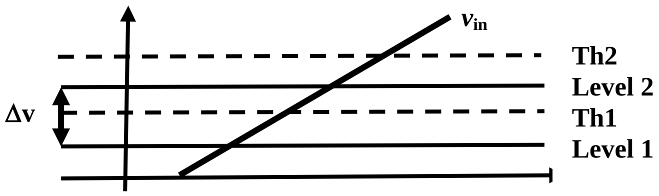


Figure 4. Illustration of linear quantization noise

Assuming linear quantisation the maximum instantaneous value of the error is equal to $\pm$ Δv/2. That is half the threshold.

The error is then linearly related to the voltage level.

Error=e(v_{in})= v_{in} $\quad -\Delta v/2 < v_{in} < \Delta v/2$

The mean error is then given by

$$\bar{e}(v_{in}) = \frac{1}{\Delta v}\int_{-\Delta v/2}^{\Delta v/2} v_{in}\,dv_{in} = 0$$

The mean square error which represents the power in a 1 Ω resistor is given by

$$mse = \frac{1}{\Delta v}\int_{-\Delta v/2}^{\Delta v/2} v_{in}^2\,dv_{in} = \left.\frac{v_{in}^3}{3}\right|_{-\Delta v/2}^{\Delta v/2} = \frac{\Delta v^2}{12}$$

The quality depends on the No. of levels. This means that we have to increase the number of quantisation levels N to a value where quantisation noise is acceptable - for example for colour **TV 512** levels give good TV while 64 levels give only fairly acceptable colour TV assuming a sinusoidal input.

In pulse code modulation, PCM the quantised signal is then given a code hence PCM.

Why use PCM?

PCM have several advantages which include the following:

1. Because the signal is discrete, it is less susceptible to noise - i. e. the signal can be completely regenerated by repeaters assuming it has not been obliterated.

2. Can be stored as discrete data on numerous digital media e.g. computers, CD, and USB.

3. Numerical operations can be **performed** on data including error correcting codes and signal processing.

4. Data could be gathered at the necessary sample rate then retransmitted at a different rate suitable for the channel and then reconstructed at the receiver. Slow down (space probe) or speed up (satellite).

5. Can be time domain multiplexed like pulse modulation.

6. Apart from interfaces to the analogue world, all elements can be built with precision and reliability of digital electronics.

Disadvantages of PCM:
a. quantization noise   b. wide Bandwidth

Illustration of a Conceptual PCM System

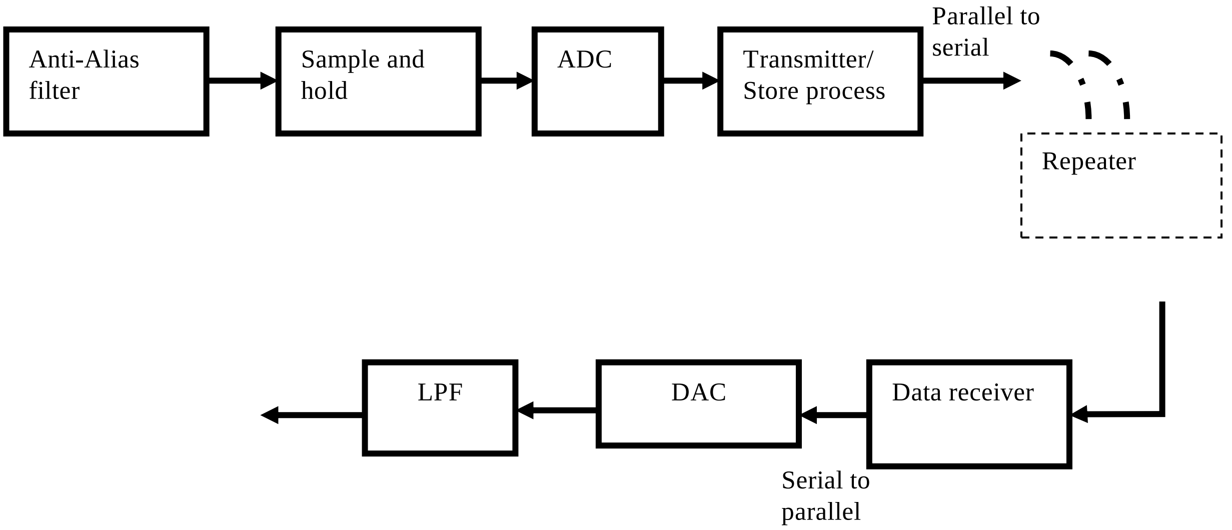


Figure 5 Illustration of a conceptual PCM system

Data Transmission

DATA
In this context we refer to Digital Binary Data - i. e. data in binary which could have been obtained by sampling, quantizing or encoding of an analogue signal, from a teletypewriter or from a computer etc.

Transmission
The operation of transferring the data from one place to another across a medium usually referred to as the communication channel.

Baseband signal: a signal which has most of its spectrum close to dc.

Proposed system for data transmission

Figure 6 shows the various blocks of a data transmission system. It consists of a digital data source. This is followed by an encoder which assigns codes for 1's and 0's. The modulator then modulates the signal to change it from baseband to band-pass. At the receiver the signal is demodulated, decoded to recover the 1's and 0's which are then used to recover the digital data

Figure 6. Block diagram of data transmission

In such a system the blocks depend on

a. the communication channel

b. ease of recovering the original signal at the R-X

Let us consider each separately:

1. <u>Communication channel</u>
Basically a channel can be

fiber Optics
light frequencies

free space
*troposphere
*ionosphere
*ground waves

wire
BW (dc-high frequency)

Depending on the **channel,** then a modulator/demodulator might be needed or not.

Band Pass Transmission: if for example a radio channel or a fibre optical channel is chosen a modulator to change the energy content of the signal to the required frequency is needed. Since for a PCM signal the main frequency content of the signal is close to dc [i.**e.** at baseband]

Baseband Transmission: For a wire link or a direct link between Transmitter, and the Receiver, no such frequency translation is necessary.

2. <u>Signal Recovery</u> at the receiver [encoder/decoder]

Since we are transmitting Digital binary data where a group of bits known as words represent a voltage level or a computer command signal or a keyboard signal etc., the receiver has to perform synchronisation at two leveles:

1. to identify each bit individually.

2. to identify each word individually

<u>BIT identification</u>

Bit identification implies deciding when the bit starts or stops which is equal to the rate at which digital data are processed or clocked and when is the best time to decide whether the bit is 1 or 0.

Three methods of synchronisation

1. transmitter and receiver are slaved to a master source from which they derive the clock.

2. sending a separate signal - pilot clock.

3. derivation from the modulation itself [code]

**Consider derivation of synchronisation pulses from modulation

The main aim here is to derive a signal which is related to the rate of the data clock. To address this problem let us look at a number of baseband signals which are used to convey bit information {Note all applies equally well to BP]

PCM codes

Popular PCM codes are illustrated in Figure 7

Non Return to Zero, NRZ: usual binary signal where + V or 0 is generated for the whole of the bit duration.

Return to Zero, RZ, 1 is represented by + V for 1/2 clock period

Polar NRZ: as NRZ except for +V and -V

Polar RZ: Three levels [+V, 0] =1: [-V, O] for 0.

Bipolar NRZ: 3 levels no **dc,** 1 is **represented** by alternative $\pm$ V, 0=0V.

Bipolar RZ: 3 levels - no dc, 1 is represented by alternate [±V, 01 and 0=0V].

Manchester code 1 {+V for 1/2 clock -V for 1/2 clock} 0 {**-V** for 1/2 clock each +V for 1/2 clock} order is not important.

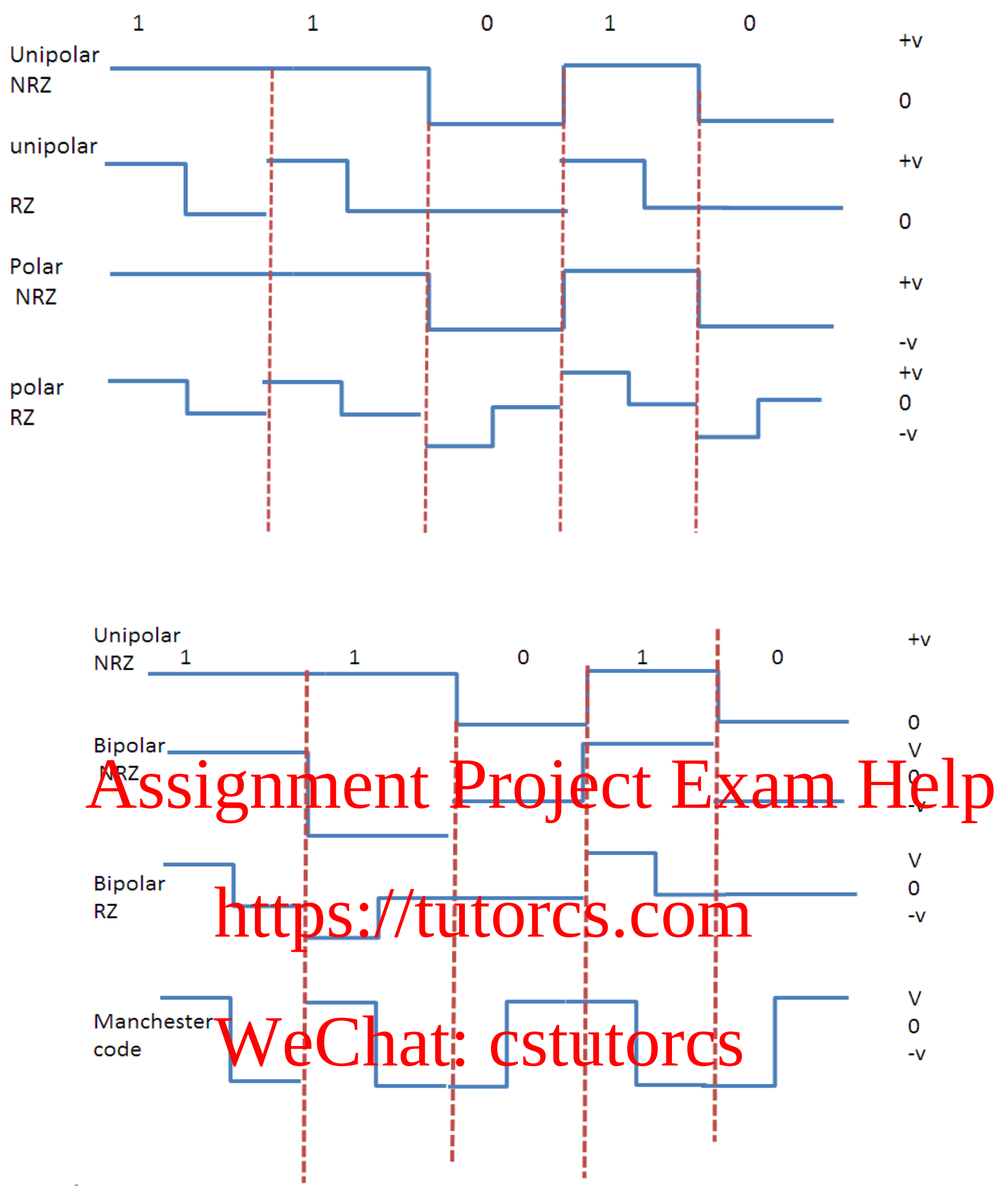


Figure 7. Popular bit formats

NOTE

1. NRZ – uni-polar, polar, or bipolar NRZ are difficult to synchronise to because there are not enough transitions in the data to give timing information. In the case of Bipolar it suffers from lack of edges to synchronise to when there is a large no. of consecutive 0's. However, NRZ is efficient in terms of its use of the BW.

2. **RZ –**

a. unipolar and bipolar - synchronisation can be derived from the 1 bits since there are transitions during the 1 bits.

b. Polar: transitions on all bits, hence synchronisation is possible by checking the level of the signal. This enables the derivation of a clock at the receiver.

Note: 1) it requires twice the BW of the NRZ 2) 3 levels are required for polar & bipolar transmission

3. Manchester code: transitions on all bits so synchronisation is possible. It does not have a dc component but again requires twice the BW of NRZ.

Clock Recovery Methods

(a) Edge Timing (b) Phase locked Loop, PLL

Edge Timing

For RZ and Manchester code there are edges to identify the half bit. This information can be used to extract the clock. Figure 8 shows a block diagram for a possible circuit.

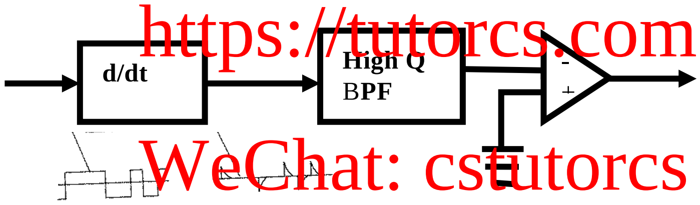
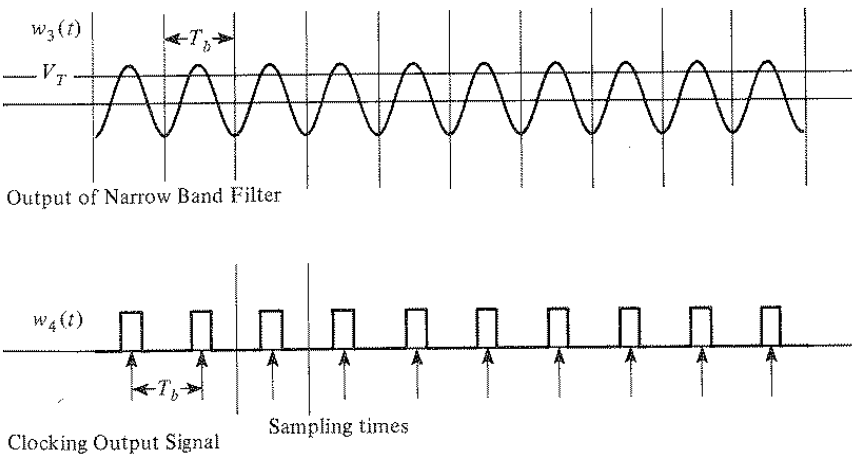


Figure 8. Block diagram of circuit for clock recovery



<u>Note</u> 1.   Since the Band Pass Filter, BPF has a high Q it takes a long time to lose its output due to a drop in input.

<u>Note</u> 2. For Manchester coding a rectifier is needed following the differentiator. This gives an output at twice the frequency which can be divided using a JK flip flop.

2. the input signal might not be that clean, hence further processing might be needed i.e. limiting.

Phase Locked Loop (PLL)

PLL- Circuit shown in Figure 9 generates a clock signal which has the right frequency. The Circuit can be locked to an external source such that the generated clock has the same frequency and phase as the external source. The clock remains in synchronisation with the incoming signal by comparing their relative phases and continuously adjusting the clock to coincide with the external clock frequency and phase

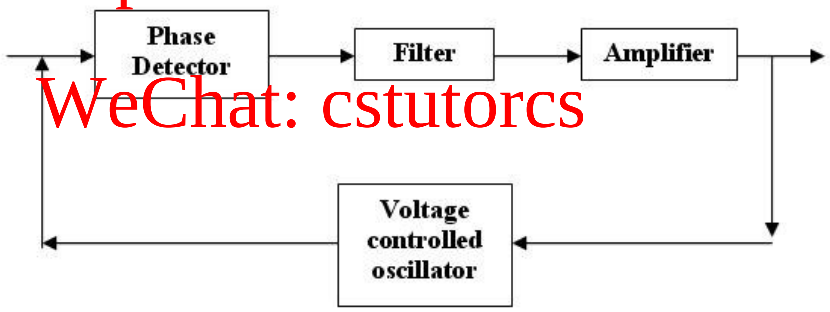


Figure 9. Basic block diagram of

VCO: voltage controlled oscillator whose frequency is controlled by the DC level at its input.

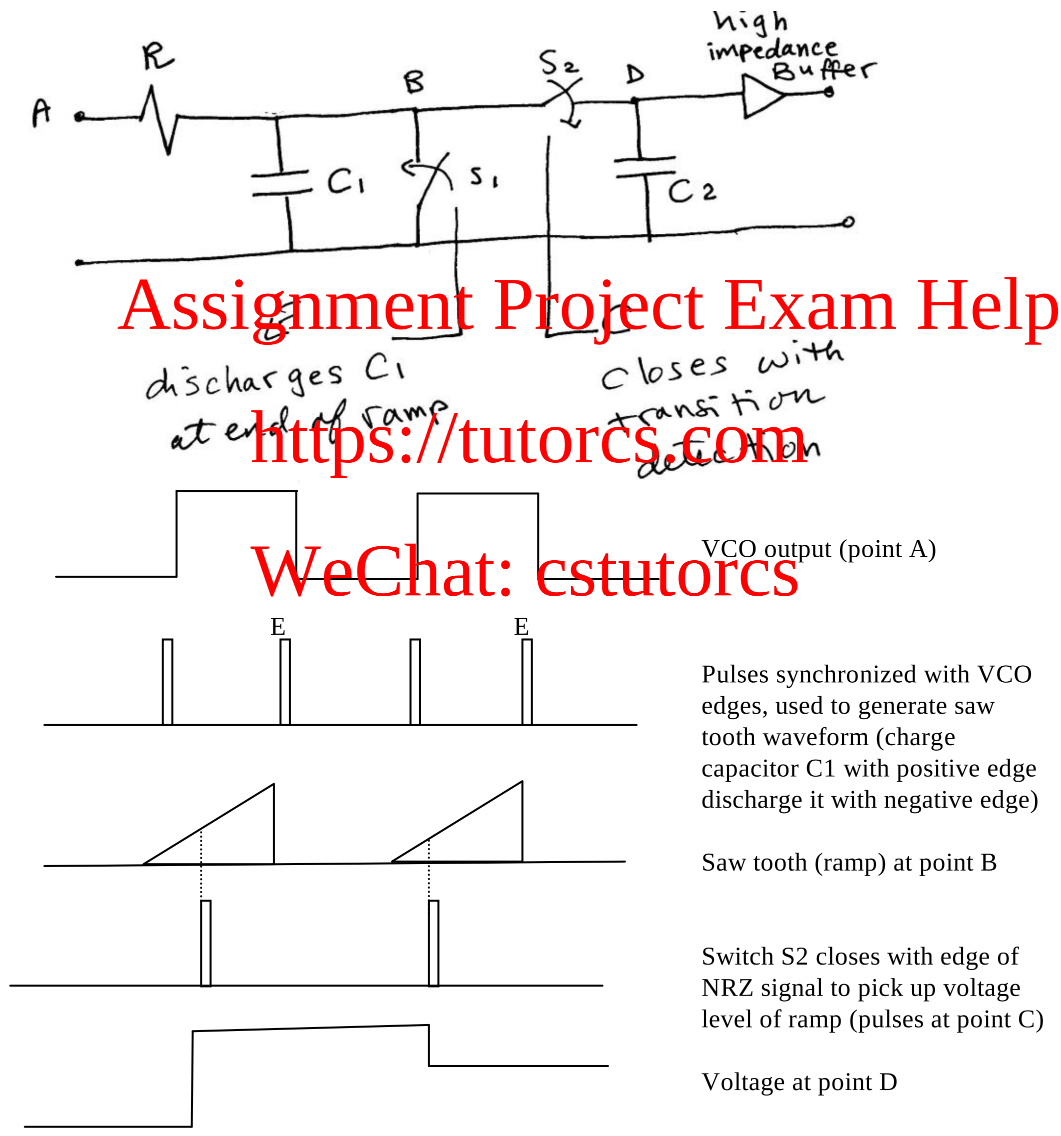
Phase detector: produces a voltage whose value depends on the phase difference between its inputs. The polarity and magnitude of the output of the phase detector are such as to bring the VCO to oscillate at a frequency equal to the input frequency.

<u>Note</u> the phase detector output is limited in range. i.e. only a certain range of frequencies can be accommodated.

<u>LPF</u> sets the loop dynamics and basically filters the output of the phase detector to produce dc voltage.

For clock recovery the PLL phase detector must remember the last voltage between transitions such as in NRZ

The figure below shows a possible phase detector with corresponding waveforms



The output (A) of the VCO is converted to a ramp **(**B**)** which is sampled when the transition detector output's a pulse [C]. The voltage on the ramp is stored in a capacitor whose contents are then transferred quickly to another smaller capacitor before it gets discharged at the end of the ramp by [E]. Hence, point **D** will **hold** the voltage necessary to correct for the VCO. Since the input impedance of the buffer is high, it will take a long time to discharge, thus compensating for lack of clock transitions in incoming signal.

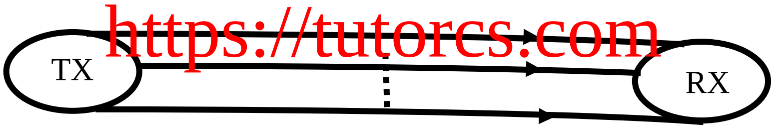
S1: open at the +ve transition of clock
C1: is charging through R.

When S2 closes with the transition detector- the voltage on **C1,** appears on C2; At the negative transition of the clock **S1** is closed and C1 is discharged.

Word synchronisation

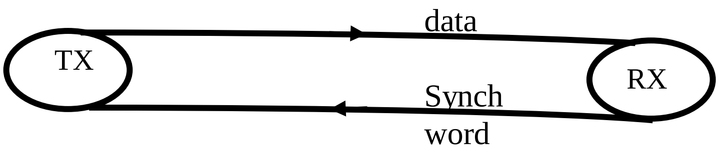
To ensure the data lists are correctly grouped one can have a number of solutions.

1. Parallel solution: for n bits send all the n bits simultaneously. This requires n links for n-bit word.



2. Transmit a different level for each word (2^n -levels): problems arise with discrimination between levels.

3. serial link plus a word synchronization link



4. Use one link for serial transmission and transmit synchronisation signals to indicate words.

There are 2 kinds of transmission which come under this category.

Synchronous serial data transmission and asynchronous serial data transmission.

I. Synchronous Serial Data transmission

Data are broken into lists which are transmitted serially in synchronisation with a highly stable clock according to a predetermined order [e.g. MSB .. LSB]. The receiver also has a highly stable clock to lock to the transmitted bit rate. A word sequence is recognised by transmitting a particular sequence every so many words. The receiver recognises this code and establishes synchronisation. Then it counts the bits according to the predetermined format to maintain word synchronisation.

Synch			- - - - - - - - -	Synch	

Note

1. If synchronisation is lost for some reason, it will not be established until a 2nd synchronisation word is received
2. Overhead - synchronisation sequence length.

The synchronisation sequence must have certain properties:
1. Un-likeliness - it should be chosen to be as unlikely as possible to occur by chance. Obviously the longer it is the better. Also the synchronisation sequence should not occur in the message i.e. codes should be chosen carefully.

2. The synchronisation sequence should look as different as possible from itself in all shifts from its central position. That is the correlation has a peak when the transmitted sequence and received sequence are exactly in synchronisation and it should ideally then go to zero. This is due to the fact that to identify the synchronisation word, the receiver synchronisation word is shifted one bit at a time and compared with the received synchronisation word until they are coincident.

Un-normalised Correlation = (No. of agreements-No. of disagreements).
Normalised Correlation = (No. of agreements-No. of disagreements)/No. of bits.

e. g a 1111111 code will produce an error if the first bit following the code is 1. Good codes are Barker Codes which have a correlation that has a peak only when the two words are in synchronisation.

For a Barker sequence.

Normalised Correlation= 1 when the words are in synchronisation, or -1 or 0 when they are out of synchronisation. There are seven Barker codes with this property. An example of the correlation property of the 7 bit Barker code is given in Table 1 and illustrated in Figure 10.   Barker codes for different values of N equal to 3, 7, 11 and 13 are: 110, 1110010, 11100010010, and 1111100110101

k																										
	x[n]											1	1	1	0	0	1	0							φ[k]	
1						1	1	1	0	0	1	0													-1	
2							1	1	1	0	0	1	0												0	
3								1	1	1	0	0	1	0											-1	
4									1	1	1	0	0	1	0										0	
5										1	1	1	0	0	1	0									-1	
6											1	1	1	0	0	1	0								0	
7												1	1	1	0	0	1	0							7	
8													1	1	1	0	0	1	0						0	
9														1	1	1	0	0	1	0					-1	
10															1	1	1	0	0	1	0				0	
11																1	1	1	0	0	1	0			-1	
12																	1	1	1	0	0	1	0		0	
13																		1	1	1	0	0	1	0	-1	

Table 1. Correlation of 7-bit Barker code

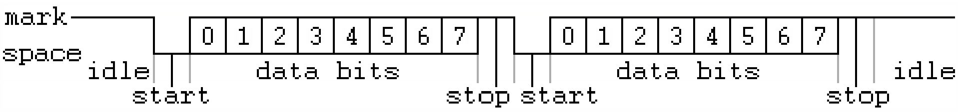
Figure 10: Un-normalised Auto-Correlation function of 7-bit Barker code

Asynchronous Serial data transmission

Each word is transmitted independently as and when it is generated. The duration of each word is fixed but the time between successive words depends on when the data are generated.

To get word synchronisation each word is preceded by a start bit and is terminated with a stop bit.

When a start bit occurs, the receiver changes from the idle state, synchronises itself with the incoming word and then goes to the idle state with the stop bit.

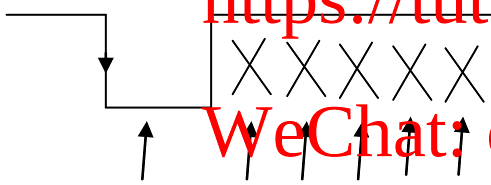


NOTE

1. Receiver resynchronises to each word

2. No need for highly stable clocks since the synchronisation is only required over short intervals of time duration equal to n bits.

3. It is important to detect the start bit clearly. Clock can drift by half a bit and still be synchronised.



To receive the data word, detect the leading edge of the start bit and then sample at the middle point of the bit.

4. This technique is mostly suitable for low noise, low distortion applications.

5. Overhead - 2 bits per word for synchronisation. This would be a great deal if data are frequent. However if data are not frequent such as keyboard transmission then it is highly suitable since data are sparse and much time can be lost for an asynchronous system

To summarise we use:
Synchronous transmission for large blocks of data
Asynchronous transmission for sporadic data

Protocols

From previous discussion, it can be concluded that there are many approaches to data transmission. Thus the transmitter and the receiver have to employ the same techniques i. e. speak the same language in order to communicate.
Elements to be agreed upon might include:

1. Data: Encoded symbols like keyboard and what code is being used? **For example** ASCII or if numerical data what format are they in? binary, BCD, Gray code etc

2. Words 1. No. of bits/word. 2. Which parity bits and if so what form of parity bits?

3. Mode: Parallel word rate? Serial: bit rate? Order of bits, synchronous or asynchronous.

4. Bit formats i.e. bit encoding such as NRZ, RZ etc.

5. Signals for Baseband define signal levels to represent (1, 0) or Bandpass: specify type of modulation.