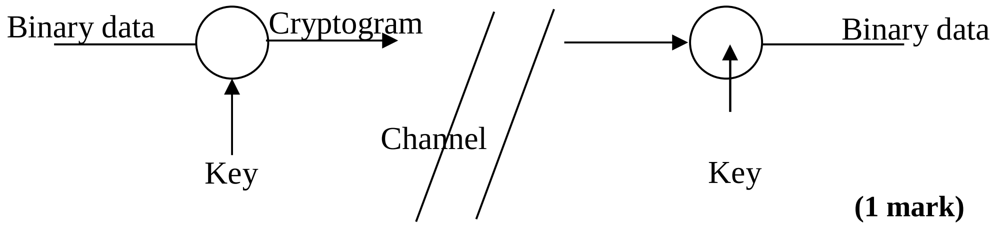
Solutions

<u>Q1a</u>

Digital data can be given an arbitrarily high level of security by encryption. Although in reality it doesn't work like that, as the National Security Agency (NSA) in the U.S. imposes an upper limit on security of transmissions, i.e. that any code is breakable by the NSA but not by anybody else. **(1 marks)**

E.g.



(1 mark)

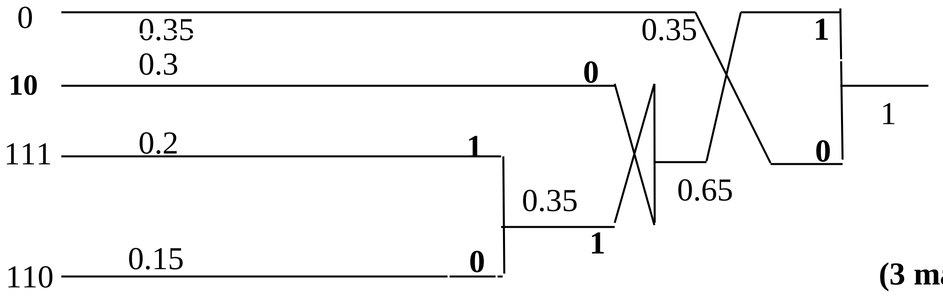
The properties of the key determine the system security. Possession of the key makes it very easy for the receiver to decode the cryptogram, whereas decoding is very time consuming/expensive for those who do not possess the key – this is the aim of cryptography. **(1 marks)**

<u>Q1b</u>

Source coding is the process whereby redundancy is removed from the message data in order to increase the rate at which information is transmitted across the channel. **(1.5 marks)**

Channel coding (or forward error correction) is traditionally applied to digital data before transmission, which adds redundant bits to help detect and correct errors that have occurred during transit at the receiver. **(1.5 marks)**

<u>Q1c</u>



(3 marks)

The entropy of the source: $H = \sum_{i=1}^{6} P_i \log_2 (\frac{1}{P_i})$ where P_i is probability of symbol i, so

H = 1.9 bits/symbol **(1 mark)**

The average message length is given by $\bar{n} = \sum_{i=1}^{6} n_i P_i$, where n_i is the bit length of the codeword for symbol i, so the average length = 2 bits /symbol **(1 mark)**

Therefore, $\text{efficiency} = \frac{H}{\bar{n}} = 0.95 = 95\%$ **(1 mark)**

<u>Q1d</u>

The input signal to the envelope detector is

$$s_i(t) = (\alpha + f(t))\cos(\omega_c t) = \text{Re}\{(\alpha + f(t))e^{j\omega_c t}\}$$ **(1 mark)**

Similarly, the input noise is given by,

$$n_i(t) = n_c(t)\cos(\omega_c t) - n_s \sin(\omega_c t) = \text{Re}\{(n_c(t) + jn_s(t))e^{j\omega_c t}\}$$ **(1 mark)**

Hence we can write the input to the envelope detector as follows,

$$s_i(t)+n_i(t)=\mathrm{Re}\left\{\left(\alpha+f(t)+n_c(t)+jn_s(t)\right)e^{j\omega_c t}\right\}$$ **(0.5 mark)**

The modulus of the input waveform is given by

$$r(t)=\left|s_i(t)+n_i(t)\right|=\sqrt{\left(\alpha+f(t)+n_c(t)\right)^2+n_s(t)^2}$$ **(0.5 mark)**

r(t) is the output from the envelope detector. Rewriting *r(t)* as,

$$r(t)=\left(\alpha+f(t)+n_c(t)\right)\sqrt{1+\frac{n_s(t)^2}{\left(\alpha+f(t)+n_c(t)\right)^2}}$$ **(0.5 mark)**

If the signal to noise ratio is large then we can approximate the square root term as unity.

$$r(t)\approx\left(\alpha+f(t)+n_c(t)\right).$$ **(0.5 mark)**

The right hand side shows the DC term, the signal term and the noise term respectively. Thus the mean output signal and noise power is

$S_o=\overline{f(t)^2}$ and $N_o=\overline{n_c(t)^2}=\overline{n_i(t)^2}=N_i$. **(1 mark)**

The mean input signal power is

$$S_i=\overline{\left(\alpha+f(t)\right)\cos(\omega_c t)^2}=\frac{1}{2}\left(\alpha^2+\overline{f(t)^2}\right)$$ **(1 mark)**

Thus $\dfrac{S_i}{N_i}=\dfrac{\alpha^2+\overline{f(t)^2}}{2\overline{n_i(t)^2}}$ and $\dfrac{S_o}{N_o}=\dfrac{\overline{f(t)^2}}{\overline{n_i(t)^2}}$ **(1 mark)**

Substitute for $\dfrac{S_o}{N_o}$ in the $\dfrac{S_i}{N_i}$ equation gives $\dfrac{S_i}{N_i}=\dfrac{\alpha^2}{2\overline{n_i(t)^2}}+\dfrac{S_o}{N_o}$, which can be re-arranged to get $\dfrac{S_o}{N_o}=2\dfrac{S_i}{N_i}-\dfrac{\alpha^2}{\overline{n_i(t)^2}}$ **(1 mark)**

<u>Q2a</u>

● Variable data rate required - the channel bandwidth will be constant so buffers are needed. **(1 mark)**

●A bit error can cause large areas of picture to be incorrect - a low error rate is needed. **(1 mark)**

<u>Q2b</u>

In linear block encoding a set of message bits k is mapped into a group of n bits. We assume in this section that bit errors occur randomly. If k = 3 then we can form 8 possible messages:

000	001	010	011
100	101	110	111

The **distance** between two code words is the number of bits you need to change to go from one code word to another. For example, the code word 000 is a distance of 2 away from 011 and 000 is 3 away from 111. **(2 marks)**

The problem with this coding system is that if we get an error then you get another valid message and the error can go undetected. It is possible to map these $2^{k=3}$ to $2^{n=4}$ codes and make each new code word have a minimum distance of 2:

0000	1001	1010	0011
1100	0101	0110	1111

Now if we transmit 0011 and get a one bit error we will get any of these codes: 0010 0001 0111 1011 which are not valid code words so we detected the error. This type of code is called a linear block code. **(2 marks)**

We can add more redundancy by mapping to a larger alphabet of code words, i.e. 5 bit words which have a minimum distance of 3:

01111	10011	01000

Now a 1-bit error in the code word 01111 to 01110 produces a code, which is 1 bit away from 01111 and 2 or more bits away from the other codes. Not only we have detected the 1 bit error, but we can correct it as well. **(2 marks)**

Q2c

$$P_{(y=0)} = \gamma P_{(x=0)} + \chi P_{(x=1)}$$ **(1 mark)**

When x=0, there are two possibilities of receiving 0 at the output: The first is when there are no errors in the channels, and the second is when there are tow errors that change 0 to 1 and 1 to 0. This can be expressed as

$$\gamma = \left(\alpha^2 + (1-\alpha)(1-\beta)\right) = 0.48$$ **(1.5 marks)**

Similarly, when x=1, it is possible to achieve y=0 when there is a single error in the transmission that change 1 to 0 either in the first hop or in the second hop, which can be expressed as

$$\chi = (1-\beta)(\alpha+\beta) = 0.39$$ **(1.5 marks)**

$$P_{(y=0)} = 0.48 \times P_{(x=0)} + 0.39 \times P_{(x=1)} = 0.42$$ **(0.5 marks)**

Similarly, it can be shown that

$$P_{(y=1)} = 0.61 \times P_{(x=1)} + 0.52 \times P_{(x=0)} = 0.58$$ **(1 mark)**

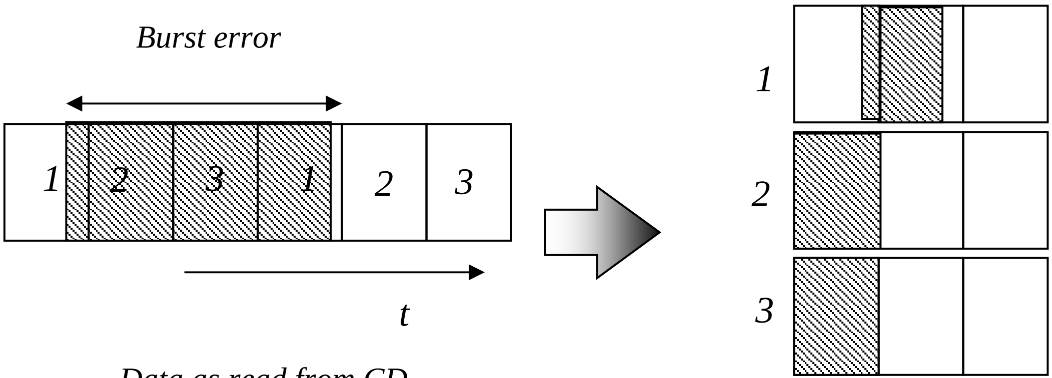
The answer can be check using

$$P_{(y=0)} + P_{(y=1)} = 1$$ **(0.5 marks)**

Q2d

Sometimes the bit errors will clump together, for example errors due to a sustained period of electrical interference, which are known as burst errors. Reed-Solomon codes are in essence parity check codes, although the method by which the additional bits are derived is significantly more complex in order that the code is efficient at correcting burst errors. R-S codes, of which there are many, are characterised by three quantities; *n* – The final length of the code, *k* – The initial length of the code and *t* – The number of errors which you can correct with the code. **(2 marks)**

Hence, like parity checks, R-S codes have a pre-defined limit as to the number of errors which can be corrected. In order to increase the effective number of errors an R-S code can correct, it is usual that R-S code blocks are interleaved (a process known as Forney interleaving). **(1 mark)**



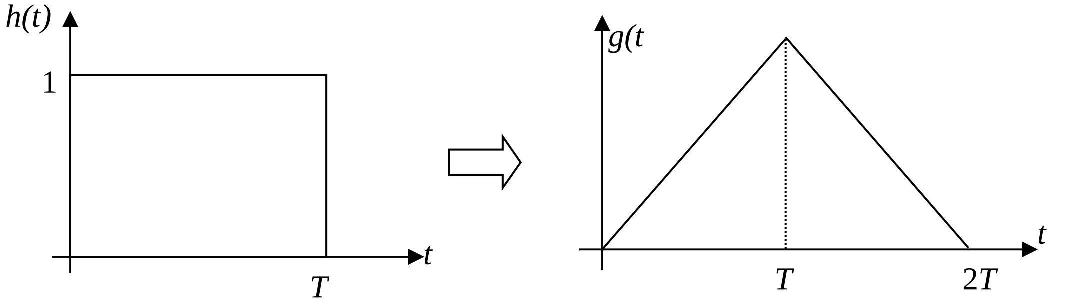
Data as read from CD

Data reconstructed from CD **(2 marks)**

For example, R-S codes are used to encode data on CDs and DVDs, this is because if you knock the player, the laser moves and you miss a chunk of data and so we have a burst error. To improve the capability of the R-S code, the bit stream is split up into a number of chunks, re-arranged and then coded on the CD. Now, when a burst error corrupts a large chunk of data read from the CD, we actually only corrupt a small portion of several messages, making it easier for the RS codes to cope. **(1 mark)**

Q3a

The matched filter output for a square wave is as follows.

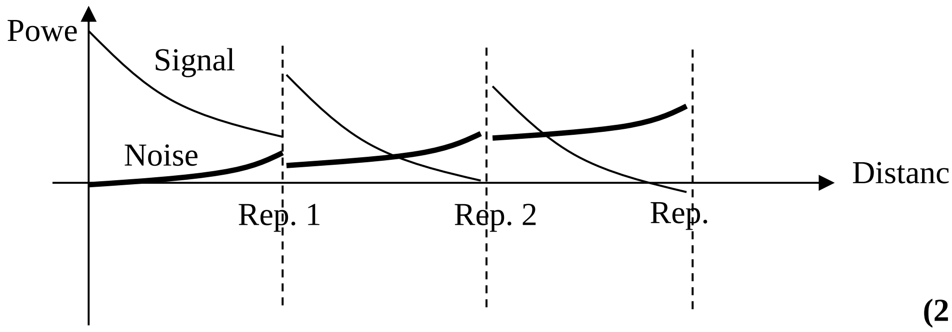


(1 mark)

The first half of the matched filter response is simply the integral of the message waveform, and hence in this region an integrator will give the same response as a matched filter. We take a sample of g(t) at *T* as this is the point where the integrator (& matched filter) gives the maximum output. **(1 mark)**

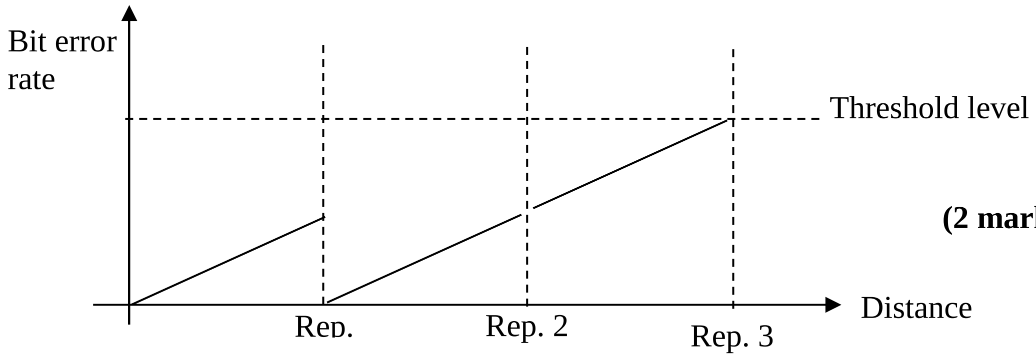
Q3b

In an analogue system a transmission repeater amplifies both the signal and the noise. Therefore, signal to noise ratio progressively reduces with distance. **(1 mark)**



(2 marks)

However, for a digital transmission which channel coding a regenerator can correct all errors below a certain threshold level.

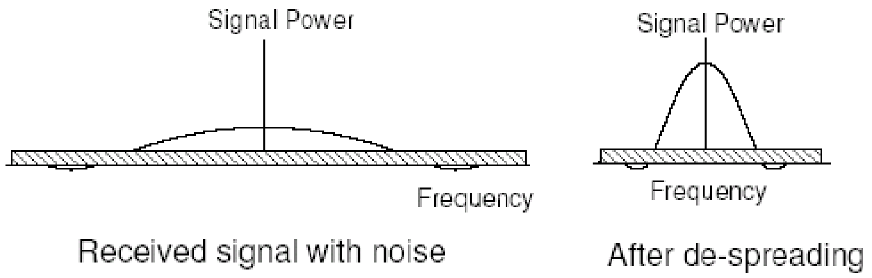


(2 marks)

Hence we have distance independent signal to noise ratio (very useful for optical fibre links – 100s of km) at the expense of circuit complexity. **(1 mark)**

Q3c

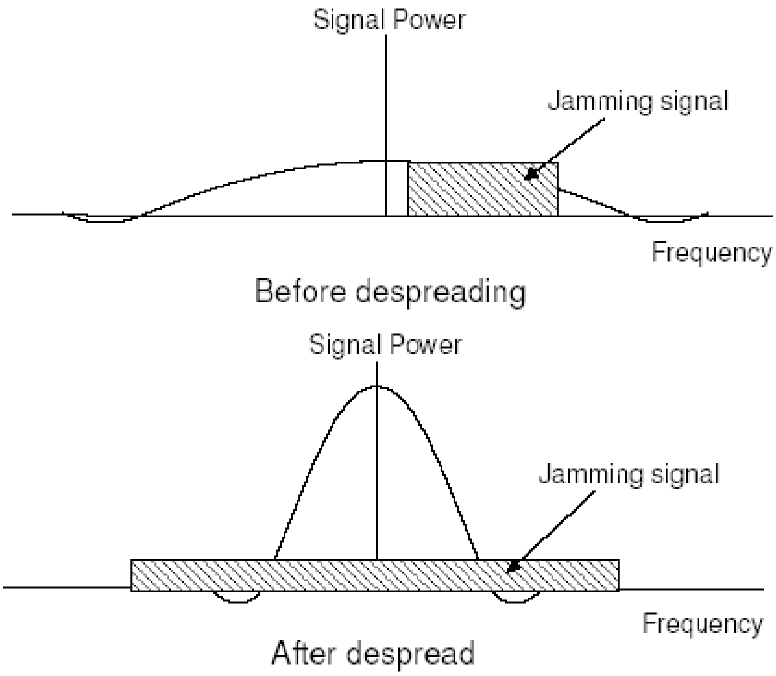
At the receiver the incoming signal is fed into a spreading circuit identical to that used at the transmitter. Because the original signal has already been spread using the PN code the effect of the spreading circuit is to de-spread the signal.



(1 mark)

(i) AWGN embedded in the received signal will not correlate with PN code, so the spreading circuit has no effect (multiplying wide band noise with wide band noise will result in wide band noise!). **(1 mark)**

(ii) A jammer only has a finite power so can choose to distribute a small amount of noise power over a large bandwidth or a large amount of noise power over a small bandwidth. **(1 mark)**



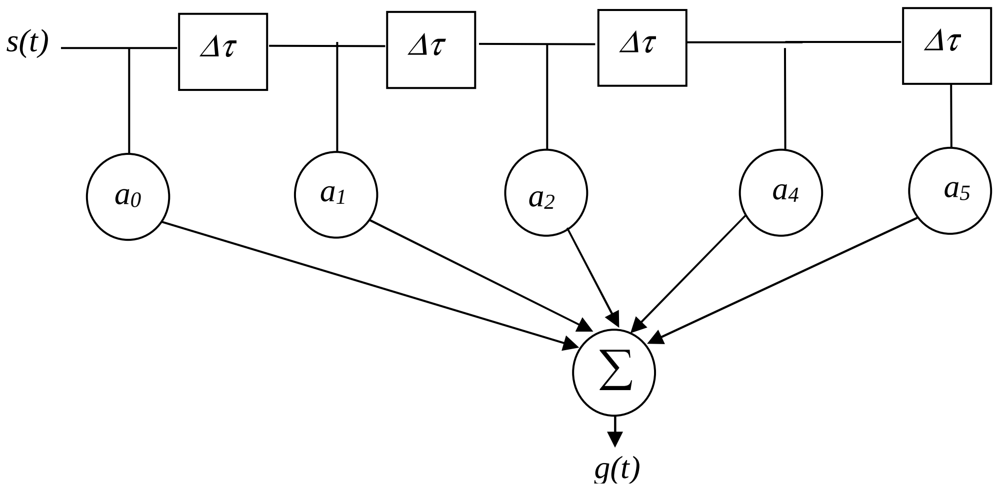
(1 mark)

The former will just look like AWGN and will not be very effective. In the latter case the received signal will consist of the spread signal which has high spectral density noise embedded over a narrow band. **(1 mark)**

The signal correlates with the PN code so will be de-spread as before. The noise, however, does not correlate with the PN code so the de-spreading circuit will look like a spreading circuit. The net result is that the signal spectral density is increased and the noise spectral density is reduced. We are assuming here the jamming transmitter is passive, frequency follower jammers can detect the transmission frequency as it changes. **(1 mark)**

Q3d

Five taps transversal filters is shown below,



The output is given by $g(t_o) = \sum_{k=0}^{n} a_k s(t_o - k\Delta\tau)$ **(1.5 marks)**

The output of a matched filter is given by $g(t_o) = \sum_{k=0}^{n} h(k\Delta\tau) s(t_o - k\Delta\tau) . \Delta\tau$ **(1 mark)**

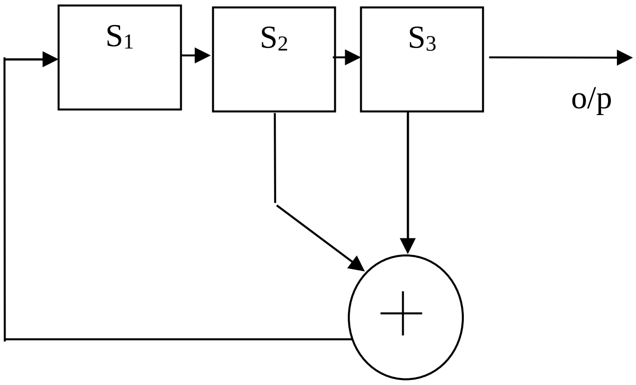
It can be seen that the two equations are equivalent, i.e. the transversal filter is a matched filter, when
$a_k = h(k\Delta\tau) . \Delta\tau$. **(1 mark)**
In this case; $h(t) = s(t_o - t) = -10\cos(500\pi t)$, since cos is an even function. Therefore,
$a_k = -10\cos(500\pi k\Delta\tau) . \Delta\tau = -5\times10^{-3}\cos(0.25\pi k)$. **(1 mark)**
The tap coefficients weights can then be calculated as
$a_0 = -5\times10^{-3}$, $a_1 = -3.5\times10^{-3}$, $a_2 = 0$, $a_3 = 3.5\times10^{-3}$, $a_4 = 5\times10^{-3}$ **(1.5 marks)**

Q4a

- Low probability of interception: The finite amount of power used to transmit the message waveform is spread over a very wide bandwidth, making the presence of a signal very difficult to detect (this energy density reduction is also often utilised by satellite communications to ensure that they do not interfere with terrestrial ones).
- Interference rejection: S.S. systems are good at rejecting signals maliciously intended to jam the transmission and also inadvertent jamming via multipath signals (i.e. the message signal which has bounced of an object or objects and hence has arrived late to the receiver).
- Higher data rates: Since a larger bandwidth is used, a higher data rate can be obtained whilst maintaining the same bit error probability.
- Multiple access: S.S. systems use unique PN codes to spread the data, and so many signals can occupy the same bandwidth without fear of interference between signals, since only each receiver has the correct PN code to unlock its signal. **(1 mark for each point)**

Q4b

A PN code generated form an n-bit shift register has a maximal length of (2^n-1). A PN sequence with this length is known as maximal length sequence. **(1 mark)**
Example of generating a maximal length sequence is shown below. Consider the case when n = 3, S_1 = 1, S_2 =1 and S_3 = 1. **(1 mark)**



S_1	S_2	S_3	o/p
1	1	1	1
0	1	1	1
0	0	1	1
1	0	0	0
0	1	0	0
1	0	1	1
1	1	0	0

(3 marks)

Q4c

Clock	**S_1**	**S_2**	**S_3**	**S_4**
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	1	0	0	1
4	1	1	0	0
5	0	1	1	0
6	1	0	1	1
7	0	1	0	1
8	1	0	1	0
9	1	1	0	1
10	1	1	1	0
11	1	1	1	1
12	0	1	1	1
13	0	0	1	1
14	0	0	0	1
15	1	0	0	0

(2 marks)

The resulting PN code 111101011001000 has eight 1's and seven zero's so it meets the balance requirement. **(1 mark)**

Consider the zero runs, there are four of them. One half of length 1, and one fourth are of length 2. The same is true for the one runs. Therefore this PN code satisfies the run requirements. **(1 mark)**

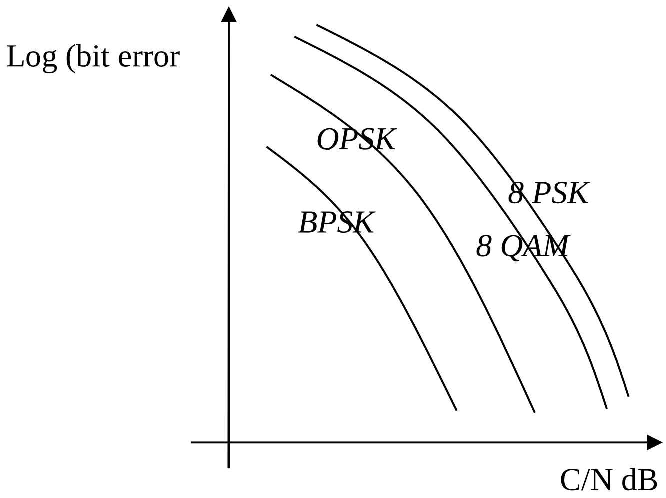
We will investigate the autocorrelation property by

$[b]\oplus[b]^{(1)}$ and $[b]\oplus[b]^{(2)}$

$[b] \oplus [b]^{(1)} =$
$[111101011001000]$
$\oplus$
$[011110101100100]$
$=$
$[100011110101100]$
eight 1's and seven 0's, i.e. autocorrelation of -1
$[b] \oplus [b]^{(2)} =$
$[111101011001000]$
$\oplus$
$[001111010110010]$
$=$
$[110010001111010]$
Again autocorrelation of -1.
So the 15 bit spreading code exhibits balance, run property and correlation. This is an example of a good PN code. **(2 marks)**

<u>Q4d</u>

Non-orthogonal M-Ary signalling is a scheme which improves the bandwidth efficiency at the expense of degraded noise performance or required signal to noise ratio. **(1 mark)**
These points are summarised by the schematic diagram shown below.



(2 marks)

The points to note about this schematic are as follows;

- Increasing M for a single type of M-Ary modulation (e.g. comparing the PSK curves on the schematic above) increases the bit error rate if the carrier to noise ratio stays the same. If M is to be increased with no effect on error probability, then the carrier to noise ratio must be increased. **(1 mark)**
- Optimised QAM will require less power to achieve the same error probability as a PSK system with the same M. **(1 mark)**

The bandwidth efficiency improves with increasing M (just a reminder, as this is not considered on the above figure) **(1 mark)**