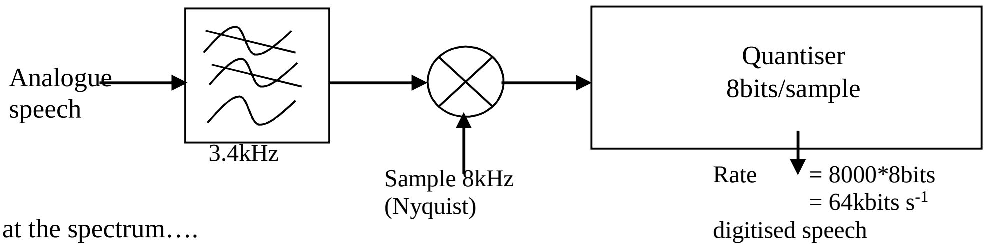
<u>**Solutions**</u>

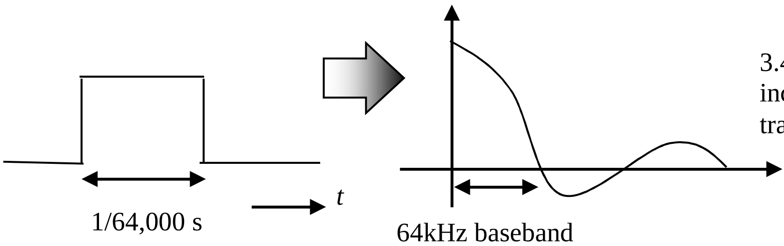
Q1a:
In fixed length coding, the same number of bits is used to encode each message. Howevere, some messages carry little information so it is a waste of resources to transmit them using same number of bits as those with higher informations. **(1 mark)**
A significantly more efficient approach is to use shorter codewords for those messages which do not carry much information, i.e. those with higher probabilities, so as to increase the information rate. **(1 mark)**
In variable length encoding, each message is encoded with a number of bits that depends on its information contents, which is very important for data compression. **(1 mark)**

Q1b: An analogue transmission system will always occupy less bandwidth than a digital system sending the same message. For example let's consider the transmission of speech:



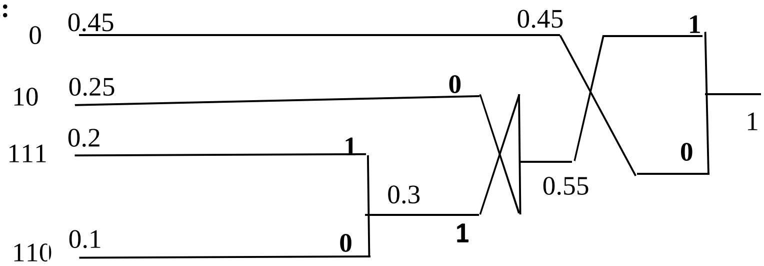
(2 marks)

Look at the spectrum….



3.4kHz analogue signal increased to 64kHz for digital transmission **(2 marks)**

Q1c:



(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 = 1.82 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 code-word for symbol i, so the average length = 1.85 bits /symbol **(1 mark)**

Therefore, $\text{efficiency} = \frac{H}{\bar{n}} = 0.98 = 98\%$ **(1 mark)**

Q1d: If a noise voltage input into our demodulator, $n_i(t)$, then the mean noise power is given by

$$N_i = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} \frac{n_i(t)^2}{R} dt$$ **(1 mark)**

where R is the resistance the power is dissipated in and T is the period of integration. If the noise is in the form of a current, then the mean noise power is now

$$N_i = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} n_i(t)^2 R dt \cdot$$ **(1 mark)**

To avoid the need to make the distinction between noise voltage and noise current sources, we'll assume for simplicity that all the noise power flows into $R = 1\Omega$. Note that in most cases R is matched to 50Ω.

(1 mark)

The component $n_i(t)$ can be defined in terms on in-phase and out-of-phase components in the baseband which we are interested in (i.e. $\omega = \omega_c$)

$$n_i(t) = n_c(t)\cos(\omega_c t) - n_s(t)\sin(\omega_c t).$$ **(1 mark)**

N_i can be calculated from these noise components as

$$N_i = \frac{1}{2T}\int_{-T/2}^{T/2} n_c(t)^2[1+\cos(2\omega_c t)] + n_s(t)^2[1-\cos(2\omega_c t)] - 2n_c(t)n_s(t)\sin(2\omega_c t)dt,$$ **(1 mark)**

the cos and sin terms average to zero to give,

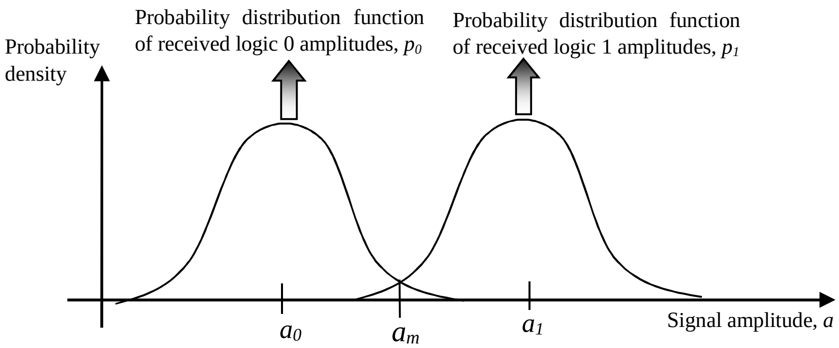
$$N_i = \frac{1}{2T}\int_{-T/2}^{T/2} n_c(t)^2 + n_s(t)^2\, dt = \frac{1}{2}\overline{n_c(t)^2} + \frac{1}{2}\overline{n_s(t)^2}.$$ **(1 mark)**

If the noise is random, then noise spikes will occur in-phase and out-of-phase with equal frequency and so we can see

$$\overline{n_c(t)^2} = \overline{n_s(t)^2} = \overline{n_i(t)^2}.$$ **(1 mark)**

Q2a: Information content is inextricably linked with the probability of that message happening. **(1 mark)**
A simple way of thinking about it is that the information content goes up as the chances of you guessing or working out the message content goes down. **(1 mark)**
We define information content, I of a statement or message, m as follows $I = -\log_2(p\{m\})$ bits, where $p\{m\}$ is the probability of m being sent. **(1 mark)**

Q2b: AWGN adds a random noise vector to the signal, the magnitude of which is determined by a probability density function that is Gaussian in nature. **(1 mark)**



(2 marks)

The degree to which AWGN spreads out the probability density function (pdf) for the amplitude of the received symbol depends upon the channel (the worse the channel the greater the spread in the pdf). **(1 mark)**
The functional form of the pdf is as follows,

$$p_{0(1)}(a) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(\frac{-\{a-a_{0(1)}\}^2}{2\sigma^2}\right)$$

where σ is the standard deviation and $a_{0(1)}$ is the mean of the distribution for logic 0 (1). **(1 mark)**

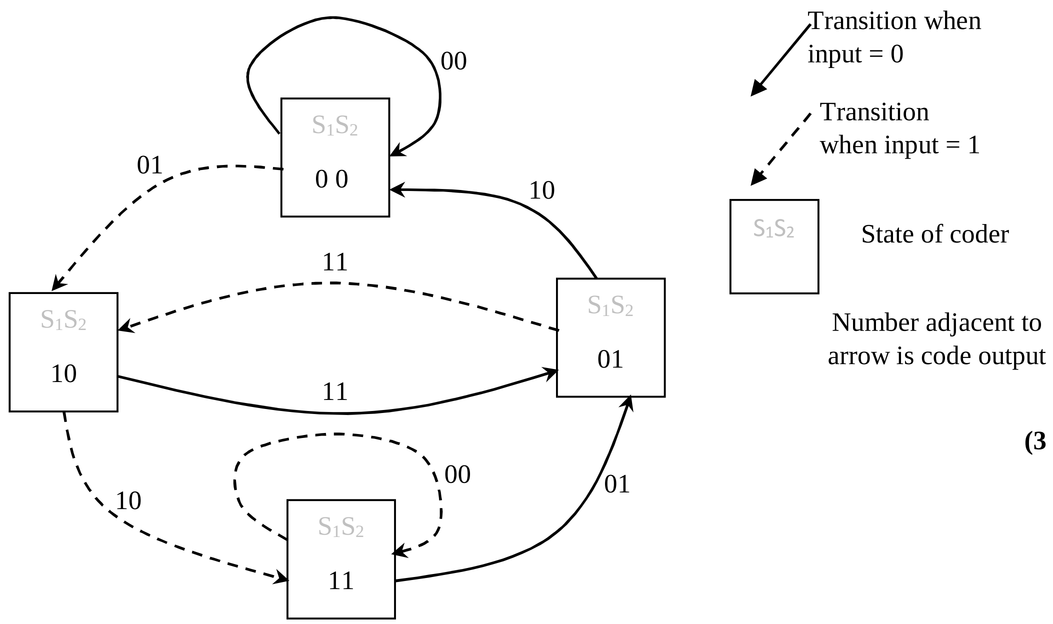
Q2c:
i.

Input S_1	Initial state S_2S_3	Next state	Output bits u_1	u_2
1	10	11	1	0
0	11	01	0	1
0	01	00	1	0
1	00	10	0	1

1	10	11	1	0
0	11	01	0	1
1	01	10	1	1
1	10	11	1	0
0	11	01	0	1

The output sequence is 1001100110011111001 **(3 marks)**

ii.



(3 marks)

Q2d: The probabilities of the first two messages are given as p(A)=p(B)=p. The probability of the 3[rd] message occurring can be obtained using

$p(C)=1- p(A)-p(B)=1-2p$ **(1 mark)**

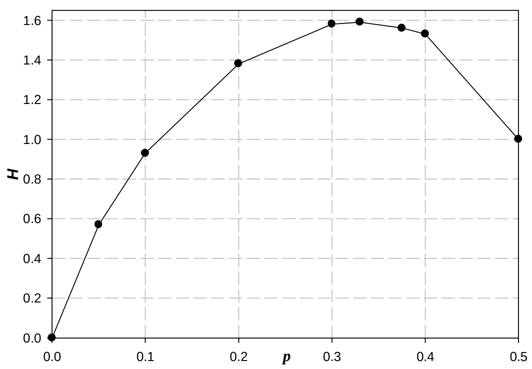
$$H=-\sum_{i=1}^{N}\mathrm{p}(i)\log_2\{\mathrm{p}(i)\}=-2p\log_2\{p\}-(1-2p)\log_2\{(1-2p)\}$$ **(1 mark)**

which gives

p	0	0.05	0.1	0.2	0.3	0.33	0.4	0.5
H	0	0.57	0.93	1.38	1.58	1.59	1.53	1

(1 mark)

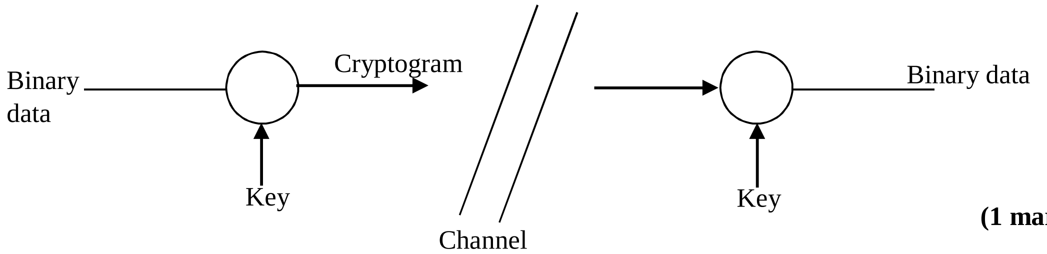
Therefore, the entropy as a function of p can be plotted as



(2 marks)

The maximum entropy has been achieved when p=0.333 which is expected because the probabilities of all the three messages are equal at this value of p. **(1 mark)**

Q3a: Digital data can be given an arbitrarily high level of security by encryption. Although in reality it doesn't usually 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 – good enough for most circumstances though! e.g. **(1 mark)**

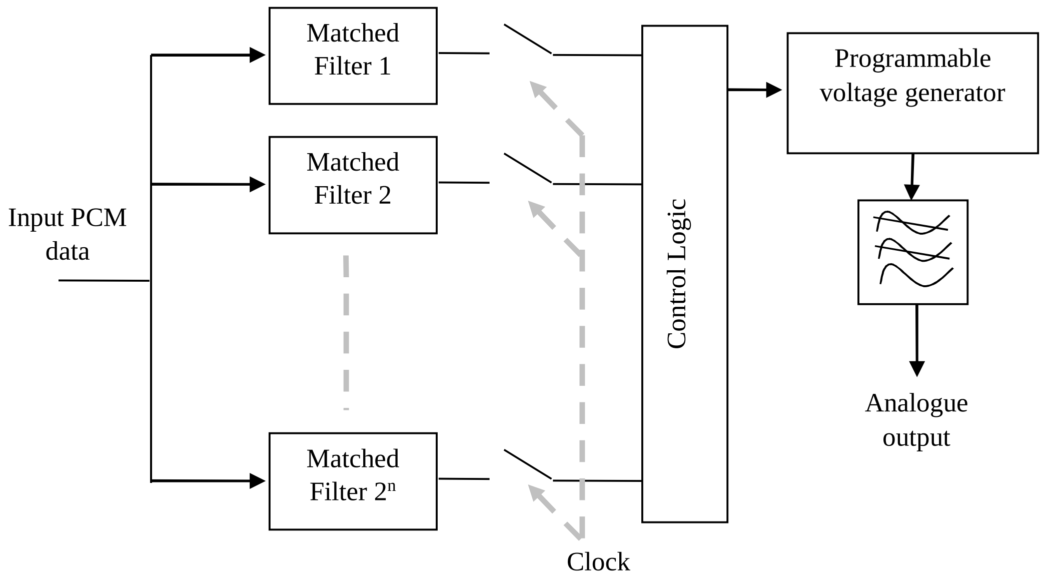


(1 mark)

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. Encryption is not possible in analogue systems. **(2 marks)**

Q3b: A fast frequency hopping (FFH) system improves the error performance in comparison to a slow frequency hopping system by sending the same data a number of times. **(1 mark)**
The rationale being that the probability of obtaining a string of errors on successive transmissions, which all carry the same information on different carrier frequencies, is very small indeed. **(1 mark)**
The improvement in error probability by employing FFH goes approximately linearly with increasing number of re-transmissions (i.e. transmitting the data 4 times will reduce the error probability an order of ~4 compared to a single transmission). **(1 mark)**
Whereas the probability of obtaining an error in the first place varies by orders of magnitude dependent upon the channel, hence if the error probability is small, then the relatively small extra improvement offered by FFH is negligible. **(1 mark)**

Q3c: In general, for an *n* bit PCM codeword system we have 2^n codewords, therefore we require 2^n matched filters. **(1 mark)**



(2 marks)

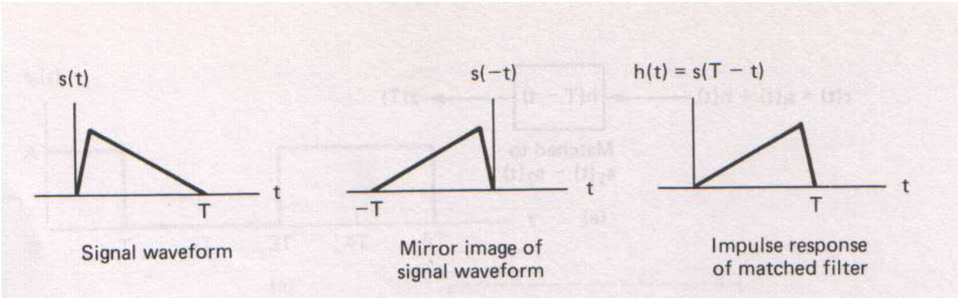
One consideration which is most important with this circuit is synchronisation. We must ensure that the point when the switches close corresponds to the point when the entire PCM code word has been fed into the matched filter. **(1 mark)**
Take for example the matched filter for the PCM word 13 (coefficients in order = 1101), now if we transmit 7 followed by 3, the bit stream is 00**1101**11. If our receiver is not synchronised and triggers when the bold section is loaded into the matched filters, we will falsely detect 13 rather than the intended messages 7 and 3. **(2 marks)**

Q3d: A filter that attempts to maximise the signal to noise ratio at the input to the detector. This can be achieved using a filter design that is *matched* to the specific transmitted signal waveform. **(1 mark)**

It can be shown that for a symbol with waveform, $s(t), 0 \le t \le T$, the filter that ensures maximum signal to noise ratio (SNR) at its output at time *T* will have an impulse response given by

$$h(t) = \begin{cases} ks(T-t), & 0 \le t \le T \\ 0, & \text{elsewhere} \end{cases}$$ **(2 marks)**

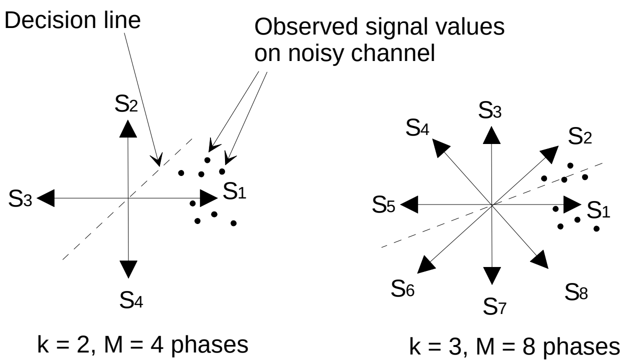
This is easily seen to be the mirror image of the original signal delayed by *T*. This delay is necessary to make the filter realisable (without the delay, the filter would have to produce an output before the input signal arrived!) **(1 mark)**



(2 marks)

Q4a: It is one that appears to be random, in other words it should have similar properties to true random sequences: **(1 mark)**
a) Balance: We want roughly the same number of 0s as we have 1s **(1 mark)**
b) Run sequence: A run sequence is a sequence of consecutive bits with the same value. To be noise-like we want 1/2 of run sequences to be of length 1, 1/4 of run sequences to be of length 2, 1/8 of run sequences to be of length 3… **(1 mark)**
c) Autocorrelation: We would like the PN sequence to have very low correlation with shifted copies of itself. **(1 mark)**

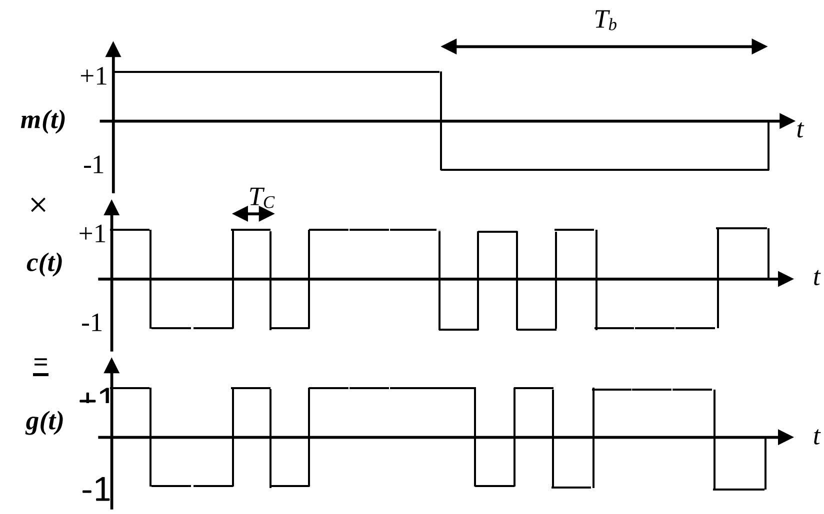
Q4b: Looking at the vector diagram:



(2 marks)

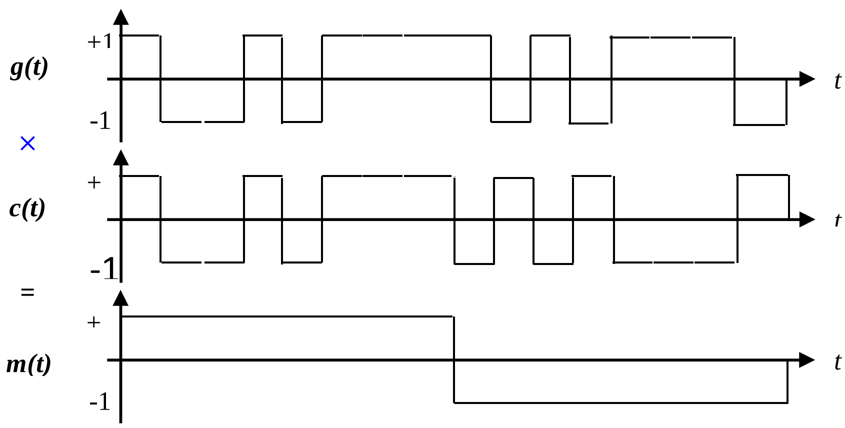
it is clear that as the number of phases increases, the angular separation between the different symbols becomes smaller and thus any phase error introduced by noise/distortion on the channel is more likely to result in the wrong symbol being detected. **(1 mark)**
Despite this, an engineer might choose to increase the number of phases (increase k) because of the increase in bandwidth efficiency that can be achieved (i.e. a higher data rate within the same bandwidth) **(1 mark)**

Q4c: Consider a data sequence of *m(t).* At the transmitter side, *m(t)* is multiplied by a spreading PN sequence, *c(t),* to obtain the spreaded signal *g(t)* as follows **(1 mark)**



(1.5 marks)

If an eavesdropper managed to demodulate the signal, what they would obtain would still look like noise and hence would be useless unless they had an exact replica of the PN code. **(1 mark)**
At the receiver we de-spread the S.S. signal by an exact replica of the PN code as follows,



(1.5 mark)

Q4d: Allows us to transmit twice the information as BPSK in the same bandwidth as it uses both in-phase and quadrature carriers. The message waveform is as follows,

In phase Quadrature

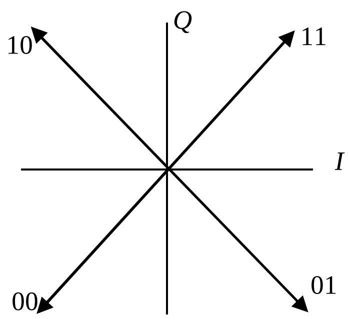
$$s(t)=\frac{A}{\sqrt{2}}d_1(t)\cos(\omega_c t)+\frac{A}{\sqrt{2}}d_2(t)\sin(\omega_c t).$$

Where $\begin{matrix} d_1(t)=\pm 1 \\ d_2(t)=\pm 1 \end{matrix}$, i.e. contains 1 bit of information. **(1 mark)**

If you look at the message waveform, we can see that the phase of the QPSK signal is given by

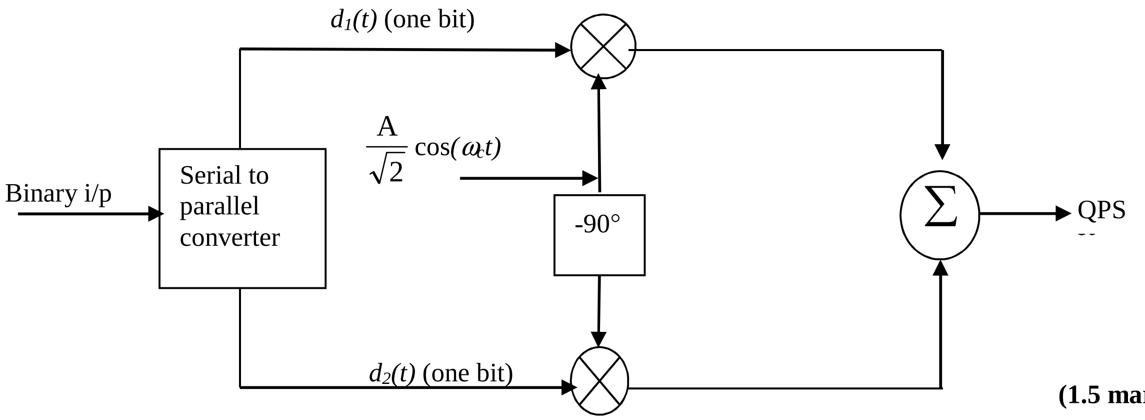
$\theta = \tan^{-1}\left(\frac{d_2(t)}{d_1(t)}\right).$ **(1 mark)**

Therefore we have four possible outcomes, $\pm 45^{\circ}$. $\pm 135^{\circ}$. Hence the phasor diagram is as follows,



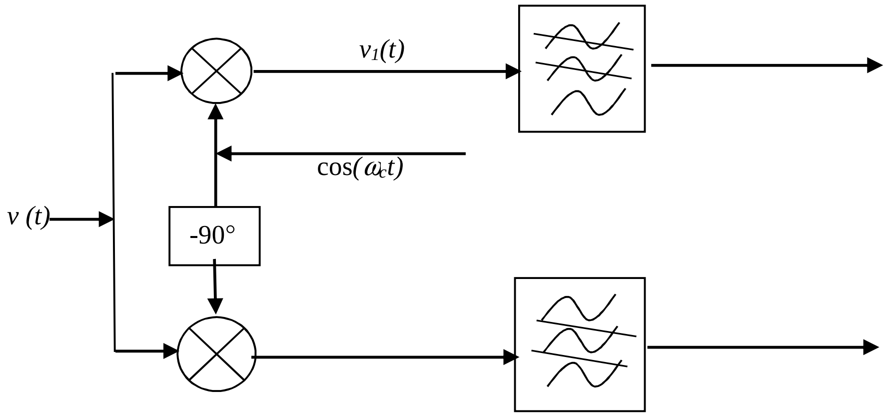
(1 mark)

One can generate QPSK by the following circuit.



(1.5 mark)

And we can detect QPSK as follows,



(1.5 mark)

Let us consider the top channel,

$$v(t)\cos(\omega_c t)$$

$$v_1(t) = \frac{A}{\sqrt{2}}d_1(t)\cos^2(\omega_c t)+\frac{A}{\sqrt{2}}d_2(t)\cos(\omega_c t)\sin(\omega_c t) \quad .$$

$$\frac{A}{\sqrt{2}}d_1(t)\frac{1}{2}[1+\cos(2\omega_c t)]+\frac{A}{\sqrt{2}}d_2(t)\frac{1}{2}[\sin(0)+\sin(2\omega_c t)]$$

Output from the low-pass filter is thus $Ad_1(t)/2\sqrt{2}$.

Similarly, the output from the bottom channel is $Ad_2(t)/2\sqrt{2}$. **(1 mark)**