PART 1

Tutorials

1

Theory of Information: Problems 1 to 15

1.1. Problem 1 – Entropy

We consider the information transmission channel of memoryless binary symmetrical type of Figure 1.1.

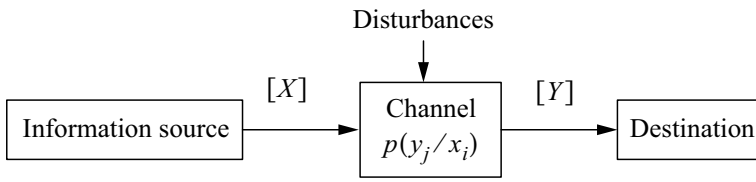


Figure 1.1. *Basic diagram of a digital communication*

It is assumed that the signal-to-noise ratio leads to the following values of conditional probabilities of errors:

$$p(y_j = 1/x_i = 0) = p(y_j = 0/x_i = 1) = p$$

$$p(y_i/x_i) = 1 - p$$

The source of binary information is considered to emit independent information with the following probabilities:

$$p(x_1) = p_1 \quad \text{and} \quad p(x_2) = p_2 = 1 - p_1$$

1) Calculate the source entropy $H(X)$.

2) Calculate the entropy $H(Y)$ at the receiver end.

3) Calculate the conditional entropy $H(Y/X)$ (entropy of transmission error).

4) Calculate the loss of information in the transmission channel $H(X/Y)$.

5) Deduce the average amount of information received by the recipient for each binary symbol sent $I(X,Y)$ (mutual information).

6) Determine the channel capacity C and show that it is obtained when $p_1 = 0.5$.

Solution of problem 1

1) By definition, we have:

$$H(X) = -\sum_{i=1}^{2} p(x_i) \log_2 p(x_i)$$

then:

$$H(X) = -\{p_1 \log_2 p_1 + (1-p_1) \log_2 (1-p_1)\} = H(p_1)$$

2) By definition, we have:

$$H(Y) = -\sum_{j=1}^{2} p(y_j) \log_2 p(y_j)$$

and:

$$p(y_j) = \sum_{i=1}^{2} p(x_i) \times p(y_j/x_i)$$

hence:

$$H(Y) = -\{[p_1(1-p) + (1-p_1)p] \times \log_2 [p_1(1-p) + (1-p_1)p]$$
$$+[p_1 p + (1-p_1)(1-p)]$$
$$\times \log_2 [p_1 p + (1-p_1)(1-p)]\}$$

3) In the same way, we have:

$$H(Y/X = x_i) = -\sum_{j=1}^{2} p(y_j/x_i) \times \log_2 p(y_j/x_i)$$

and:

$$H(Y/X) = \sum_{i=1}^{2} p(x_i)\, H(Y/X = x_i)$$

Since we are dealing with a binary symmetric communication channel, it turns out that:

$$H(Y/X) = H(Y/X = x_i) = -\{(1-p)\log_2(1-p) + p\log_2 p\} = H(p)$$

4) We have**:**

$$H(X/Y) = -\sum_{i=1}^{2}\sum_{j=1}^{2} p(x_i) \times p\left(y_j/x_i\right) \log_2 \left[\frac{p(x_i) \times p\left(y_j/x_i\right)}{p\left(y_j\right)}\right]$$

That is:

$$H(X/Y) = -\left\{p_1(1-p)\log_2\left[\frac{p_1(1-p)}{p_1(1-p)+(1-p_1)p}\right]\right.$$
$$+ p_1 p \log_2\left[\frac{p_1 p}{p_1 p + (1-p_1)(1-p)}\right]$$
$$+(1-p_1)p\log_2\left[\frac{(1-p_1)p}{p_1(1-p)+(1-p_1)p}\right]$$
$$\left.+(1-p_1)(1-p)\log_2\left[\frac{(1-p_1)(1-p)}{p_1 p + (1-p_1)(1-p)}\right]\right\}$$

5) By definition, we have:

$$I(X,Y) = H(Y) - H(Y/X)$$

6) By definition, we have:

$$C = \underset{\{p(x_i)\}}{Max}\, I(X,Y) = \underset{\{p(x_i)\}}{Max}\, H(Y) - H(Y/X)$$

$\underset{\{p_1\}}{Max}\, H(Y)$ is got for p_1 such that $\dfrac{\partial H(Y)}{\partial p_1} = 0$

$$\frac{\partial H(Y)}{\partial p_1} = -\left\{(1-2p)\log_2\left[\frac{(1-p)p_1 + p(1-p_1)}{pp_1 + (1-p)(1-p_1)}\right]\right\} = 0$$

You need to have the numerator of the log function equal to the denominator, hence:

$2p_1(1-2p) = 1-2p$; hence $p_1 = 1/2$

Thus, the maximum defines the capacity C of the communication channel and is obtained for:

$p_1 = 1/2$, hence $Max\ H(Y) = 1$ and therefore: $C = 1 - H(p)$

1.2. Problem 2 – K-order extension of a transmission channel

A memoryless binary symmetric transmission channel is considered: whatever the binary information to be transmitted, the probability of the transmission error is constant, equal to p.

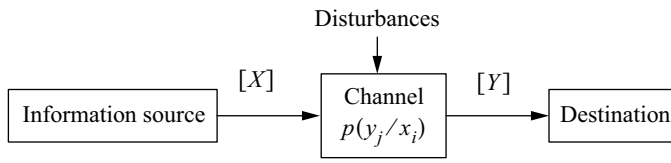


Figure 1.2. *Basic block diagram of a digital communication of a memoryless information source*

A. *K-order extension of a memoryless binary symmetric channel of error probability p*

The k-order extension channel has an input alphabet of 2^k binary words of length k and an output alphabet identical to that of the input alphabet. This channel is thus represented by a square matrix P_k of dimension $[2^k, 2^k]$ whose element p_{ij} corresponds to the probability of receiving y_j conditionally to have x_i transmitted $p(y_j/x_i)$.

1) If d is the Hamming distance between the two binary words of length k corresponding for one to the symbol x_i, and for the other to the symbol y_j, express the probability p_{ij} according to the three parameters: p, k, d.

B. *Second-order extension of a memoryless binary symmetric channel*

2) Write completely in literal form as a function of p the matrix P_2 representative of the second order extension of the binary symmetric channel.

3) The information source is considered to be transmitting equiprobable quaternary symbols x_i in the channel. Calculate the probability $p(y_j)$ to receive a symbol y_j.

4) Deduce the relationship which exists between the elements p_{ij} of the matrix P_2 representative of the second order extension of the binary symmetric channel and the probability $p(x_i/y_j)$ that the symbol x_i was emitted conditionally having received y_j.

5) Calculate the average amount of information $H(X/Y)$ lost in the channel due to transmission errors. You will express $H(X/Y)$ as a function of:

$$H(p) = -\{(1-p)\log_2(1-p) + p\log_2 p\}$$

C. *Fourth-order extension of a memoryless binary symmetric channel*

The size of the input alphabet of the source is then 16. The output alphabet is the same as that of the input alphabet.

The source is considered to emit equiprobable symbols x_i.

6) We extrapolate the result obtained in B-5 by considering that we have:

$$H(X/Y) = kH(p)$$

In the case $p = 0.03$, calculate the statistical mean of the information amount $H(X/Y)$ lost per symbol sent.

7) What is the entropy $H(X)$ of the source?

8) What is the maximum number of possible errors on a symbol received?

Solution of problem 2

A. K-order extension

1) The symbol x_i is made up of k bits. It is the same for the symbol y_j, so:

$$p_{ij} = p(y_j/x_i) = p(y_{j,1}, y_{j,2}, \cdots, y_{j,k}/x_{i,1}, x_{i,2}, \cdots, x_{i,k})$$

The communication channel is memoryless, so the probability of obtaining a given bit at the output depends only on the bit transmitted at the input (in addition to the intrinsic properties of the transmission channel itself), hence:

$$p(y_j/x_i) = p(y_{j,1}/x_{i,1}) \times p(y_{j,2}/x_{i,2}) \times \cdots \times p(y_{j,k}/x_{i,k})$$
$$= \prod_{n=1}^{k} p(y_{j,n}/x_{i,n})$$

because of the independence between the source of information and the communication channel.

The Hamming distance $d = d_H(y_j, x_i)$ is the number of bits of the same rank that are different between the symbol y_j and symbol x_i.

Then:

$$p(y_j/x_i) = p^d(1-p)^{k-d}$$

This law is close to the Binomial law because if p is the probability of a wrong decision on bit b, then $(1-p)$ is the probability of a right decision on bit b.

B. Second-order extension of the channel

2) We have:

$$k = 2 \rightarrow p(y_j/x_i) = p^d(1-p)^{2-d} \rightarrow \text{the matrix } P_2 \text{ (see Table 1.1)}$$

$$\sum_{j=1}^{4} p(y_j/x_i) = 1 = \sum_{i=1}^{4} p(y_j/x_i)$$

because of the symmetry.

		j **1**	**2**	**3**	**4**
		y_j 0 0	0 1	1 0	1 1
i	x_i				
1	0 0	$(1-p)^2$	$p(1-p)$	$p(1-p)$	p^2
2	0 1	$p(1-p)$	$(1-p)^2$	p^2	$p(1-p)$
3	1 0	$p(1-p)$	p^2	$(1-p)^2$	$p(1-p)$
4	1 1	p^2	$p(1-p)$	$p(1-p)$	$(1-p)^2$

Table 1.1. *Matrix P_2 representative of second-order extension of a binary symmetric channel*

3) We have:

$$p(x_i, y_j) = p(x_i) \times p(y_j/x_i) = p(y_j) \times p(x_i/y_j)$$

$$p(y_j) = \sum_{i=1}^{4} p(x_i, y_j) = \sum_{i=1}^{4} p(x_i) \times p(y_j/x_i)$$

Yet, the symbols are equiprobable:

$$p(x_i) = \frac{1}{4} \; \forall \; i = 1, \cdots, 4$$

Then, the symbols y_j are also equiprobable:

$$p(y_j) = \frac{1}{4}[(1-p)^2 + 2p(1-p) + p^2] = \frac{1}{4} \; \forall \; j = 1, \cdots, 4$$

4) We have:

$$p(x_i/y_j) = \frac{p(x_i) \times p(y_j/x_i)}{p(y_j)} = p_{ij}$$

because:

$$p(x_i) = p(y_j) = 1/4$$

5) Average amount of bit of information $H(X/Y)$ lost in the transmission channel.

We have:

$$H(X/Y = y_j) = -\sum_{i=1}^{4} p(x_i/y_j) \log_2 p(x_i/y_j)$$

$$H(X/Y) = E\{H(X/Y = y_j)\} = \sum_{j=1}^{4} p(y_j) H(X/Y = y_j)$$

$$H(X/Y) = -\frac{1}{4} \sum_{j=1}^{4} \sum_{i=1}^{4} p(y_j/x_i) \log_2 p(y_j/x_i)$$

because here we have:

$$p(x_i/y_j) = p(y_j/x_i)$$

$$H(X/Y) = -\frac{1}{4}[(1-p)^2 \log_2(1-p)^2 + 2p(1-p)\log_2 p(1-p) + p^2 \log_2 p^2] \times 4$$

$$H(X/Y) = -2\{(1-p)^2 \log_2(1-p) + p(1-p)[\log_2 p + \log_2(1-p)] + p^2 \log_2 p\}$$

$$H(X/Y) = -2\{(1-p)[(1-p)\log_2(1-p) + p\log_2 p] + p[(1-p)\log_2(1-p) + p\log_2 p]\}$$

$$H(X/Y) = 2[(1-p)H(p) + pH(p)] = 2H(p) = kH(p)$$

C. Fourth-order extension of the transmission channel

6) $p = 0.03$ and $H(X/Y) = 4H(p)$.

Average amount of information (in bit of information) lost per binary symbol sent?

We have:

$$H(X/Y) = -4[0.97 \times \log_2(0.97) + 0.03 \times \log_2(0.03)]$$
$$= 0.7777 \text{ bit of information/symbol}$$

7) Entropy of the source?

$$H(X) = H(S^4) = 4H(S)$$

and:

$$H(S) = -\sum_{i=1}^{2} p(b_i)\log_2 p(b_i) = 1 \quad \text{because} \quad p(b_1) = p(b_2) = \frac{1}{2}$$

hence:

$$H(X) = 4 \text{ bits of information/symbol}$$

8) Maximum number of possible errors?

$$d_{max} = 4$$

1.3. Problem 3 – Compressed speech digital transmission and Huffman coding

In the context of the transmission of the highly compressed speech signal over the telephone channel entirely in digital form, let us look at the problem of statistical source coding.

An information source S delivering elementary symbols s belonging to a symbol dictionary of size 6 is considered. The probabilities of transmission of this simple source of information are given in Table 1.2.

$\boldsymbol{s_i}$	s_1	s_2	s_3	s_4	s_5	s_6
$\boldsymbol{Pr\{s_i\}}$	0.05	0.20	0.22	0.33	0.15	0.05

Table 1.2. *Probabilities of emitting symbols s by the information source*

The symbols are delivered by the source S every $T = 10^{-3}\ s$.

1) Determine the entropy $H(S)$ of the source. Deduce the entropy bitrate D_s.

2) Construct the statistical Huffman coding, called code C_1, which generates a binary code associated with each symbol s_i.

3) Deduce the average length $\overline{l_1}$ of code C_1 and the bitrate D_1 per second.

4) What are the efficiency η_1 and redundancy ρ_1 of code C_1?

5) If we chose a fixed-length code (code C_2), what would be its efficiency η_2? What do you conclude?

6) Would it be possible to transmit this source of information over a transmission channel having a bitrate capacity of 2,400 bit/second?

Solution of problem 3

1) The entropy of the source is:

$$H(S) = -\sum_{i=1}^{6} p(s_i) \log_2 p(s_i)$$

Recall:

$$\log_2(Z) = \frac{\log_e(Z)}{\log_e(2)} \quad \text{and} \quad \frac{1}{\log_e(2)} \cong 1.44$$

$$H(S) \cong -1.44[0.05\log_e(0.05) + 0.2\log_e(0.2) + 0.22\log_e(0.22)$$
$$+0.33\log_e(0.33) + 0.15\log_e(0.15) + 0.05\log_e(0.05)]$$
$$\cong 2.31 \text{ bits of information/symbol}$$

The entropy bitrate of the source is:

$$D_s = \frac{H(S)}{T} = 2.31 \times 10^3 = 2.31 \text{ Kbits of information/s}$$

2) Construction of the Huffman code.

Symbol s_i	$p(s_i)_0$	$p(s_i)_1$	$p(s_i)_2$	$p(s_i)_3$	$p(s_i)_4$	Code C_1
s_4	0.33 →	0.33 →	0.33	0.42	0.58 0	0 0
s_3	0.22 →	0.22	0.25	0.33 0	0.42 1	1 0
s_2	0.20 →	0.20	0.22 0	0.25 1		1 1
s_5	0.15 →	0.15 0	0.20 1			0 1 0
s_1	0.05 0	0.10 1				0 1 1 0
s_6	0.05 1					0 1 1 1

Table 1.3. *Construction of the Huffman code* C_1

3) Average length of codewords:

$$\overline{l_1} = \sum_{i=1}^{6} p(s_i) \times l_i = 0.05 \times 4 + 0.20 \times 2 + 0.22 \times 2 + 0.33 \times 2$$
$$+ 0.15 \times 3 + 0.05 \times 4 = 2.35 \text{ bit/symbol}$$

Bitrate per second:

$$D_1 = \frac{\overline{l_1}}{T} = 2.35 \text{ Kbit/second}$$

4) Efficiency and redundancy of the Huffman code:

$$\eta_1 = \frac{H(S)}{\overline{l_1}} = \frac{2.31}{2.35} \cong 98.3\ \%$$

$$\rho_1 = 1 - \eta_1 = 0.017$$

5) Fixed-length code C_2.

Since we have 6 messages, we need 3 bits as: $2^2 < 6 < 2^3$, then:

$$\eta_2 = \frac{H(S)}{3} = \frac{2.31}{3} = 77\ \%$$

The fixed-length code C_2 is less efficient than the Huffman code C_1.

The bitrate per second with code C_2 is: $3 \times 1{,}000 = 3\ \text{Kbit/s}$.

6) The capacity of the channel is 2.4 Kbit/s, so we can transmit the code C_1 but not the code C_2 because the bitrate of C_2 is more important than the capacity of the channel.

1.4. Problem 4 – Coding without and with information compression

We consider a digital communication system, designed for the transmission of a signal $s(t)$ in digital form on a 34 Mbit/s transmission channel. Subsequently, we are only interested in a part of the transmitter, composed of a device for digitization and serialization (sampling, linear quantization on 8 bits, parallel to serial bytes transformation) represented in Figure 1.3. The sampling frequency is 10 MHz.

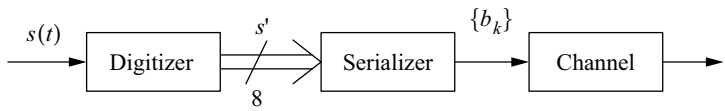


Figure 1.3. *Block diagram of a digital transmission system for analog signal*

1) With the system in Figure 1.3, is it possible to transmit this signal on the channel?

The bitrate D_1 is important, so we try to reduce it. For this purpose, a coding system with information compression of DPCM type (Differential Pulse Code

Modulation) is interposed between the digitization and serialization blocks. The DPCM coding system transforms the 256-level representation $s'(t_k)$ into a 9-level representation $s(t_k)$. The symbol s (corresponding to the encoded amplitude of the sample $s(t_k)$), is represented according to a natural binary code.

2) What is the bitrate D_2 at the output of the serialization unit?

To further reduce the bitrate, a block coding C which groups two consecutive symbols to form bijectively a single code symbol $S(t_k)$ is inserted after the encoding system DPCM (thus it has a frequency half that of s) : $\{s(t_{2k}), s(t_{2k+1})\} \leftrightarrow S(t_k)$.

3) The coding C does not using any statistical properties of s, what is its bitrate D_3?

To further reduce the bitrate, a Huffman code C_4 is used as the code C but without grouping by two the symbols s. The probabilities of realization of s are the following:

$$Pr(s = s_1) = Pr(s = s_3) = Pr(s = s_4) = 0.0625$$

$$Pr(s = s_2) = Pr(s = s_5) = 0.125$$

$$Pr(s = s_6) = Pr(s = s_9) = 0.03125$$

$$Pr(s = s_7) = Pr(s = s_8) = 0.25$$

4) Construct the Huffman code C_4. You will explicitly determine the codewords associated with each of the possible realizations of s.

5) Determine the average length $\overline{l_4}$ of the codewords of C_4 and the entropy $H(s)$.

6) What is the bitrate D_4 of the code C_4? What is its efficiency η_4? Can the signal be transmitted on the transmission channel?

7) We want to protect the binary information transmitted against transmission errors. The block encoding technique is used. This technique adds 15 bits of protection (packet error detection code) to a packet of 240 useful bits. What is the new average bitrate D_5 and is it compatible with the transmission channel capacity?

Solution of problem 4

1) We have:

$$D_1 = 8 \times 10^7 = 80 \text{ Mbit/s}$$

The rate D_1 is greater than the channel capacity, thus we cannot transmit the signal on this channel.

2) It is a 9-level encoding, therefore it takes 4 bits per sample (fixed-length code), hence:

$$D_2 = 4 \times 10^7 = 40 \text{ Mbit/s}$$

3) We have:

$$\{s(t_{2k}), s(t_{2k+1})\} \quad \overset{Code\ C}{\leftrightarrow} \quad S(t_k)$$

The pair $\{s(t_{2k}), s(t_{2k+1})\}$ has $9 \times 9 = 81$ different configurations possible, and since: $2^6 < 81 < 2^7$, it takes 7 bits to encode a pair of samples, hence:

$$D_3 = 7 \times \frac{1}{2} \times 10^7 = 35 \text{ Mbit/s}$$

4) Huffman coding.

s_i	$p(s_i)_0$	$p(s_i)_1$	$p(s_i)_2$	$p(s_i)_3$	$p(s_i)_4$	$p(s_i)_5$	$p(s_i)_6$	$p(s_i)_7$	C_4
s_8	0.25	0.25	0.25	0.25	0.25	0.25	0.5	0.5 | 0	1 0
s_7	0.25	0.25	0.25	0.25	0.25	0.25	0.25 | 0	0.5 | 1	1 1
s_5	0.125	0.125	0.125	0.125	0.25	0.25 | 0	0.25 | 1		0 1 0
s_2	0.125	0.125	0.125	0.125	0.125 | 0	0.25 | 1			0 1 1
s_1	0.0625	0.0625	0.125	0.125 | 0	0.125 | 1				0010
s_3	0.0625	0.0625	0.625 | 0	0.125 | 1					0011
s_4	0.0625	0.0625 | 0	0.625 | 1						0000
s_6	0.03125 | 0	0.0625 | 1							00010
s_9	0.03125 | 1								00011

Table 1.4. *Construction of the Huffman code* C_4

5) By definition, we have for the average length:

$$\overline{l_4} = \sum_{i=1}^{9} p(s_i) \times l_i$$

$$\overline{l_4} = 2 \times \left(\frac{1}{4} \times 2\right) + 2 \times \left(\frac{1}{8} \times 3\right) + 3 \times \left(\frac{1}{16} \times 4\right) + 2 \times \left(\frac{1}{32} \times 5\right)$$
$$= 2.8125 \text{ bit/codeword}$$

and for the entropy :

$$H(s) = -\sum_{i=1}^{9} p(s_i) \log_2 p(s_i)$$

Thus, by replacing:

$$H(s) = -\{2 \times 2^{-2} \log_2 2^{-2} + 2 \times 2^{-3} \log_2 2^{-3} + 3$$
$$\times 2^{-4} \log_2 2^{-4} + 2 \times 2^{-5} \log_2 2^{-5}\}$$

$$H(s) = 4 \times \frac{1}{4} + 6 \times \frac{1}{8} + 12 \times \frac{1}{16} + 10 \times \frac{1}{32}$$
$$= 2.8125 \text{ bits of information/codeword}$$

6) Bitrate D_4 of the code C_4 and its efficiency η_4 for this source:

$$D_4 = \overline{l_4} \times 10^7 = 28.125 \text{ Mbit/s}$$

$$\eta_4 = \frac{H(s)}{\overline{l_4}} = 1$$

The code C_4 is optimal absolute, because the probabilities are of the form: $p_i = 2^{-l_i}$.

Since the bitrate D_4 is smaller than the capacity of the channel, it turns out that the signal can be transmitted on the channel.

7) Block coding for protection against transmission errors:

$$D_5 = D_4 \times \frac{255}{240} = 29.883 \text{ Mbit/s}$$

In the same way, since the bitrate D_5 is smaller than the capacity of the channel, it turns out this signal can also be transmitted in a protected manner on the communication channel.

1.5. Problem 5 – Digital transmission of a TV signal (luminance component only) with information compression and Huffman coding

An information encoding and transmitting system for transmitting a monochrome television signal $s(t)$ in digital form is considered. The general scheme of the preliminary part of this system is given in Figure 1.4.

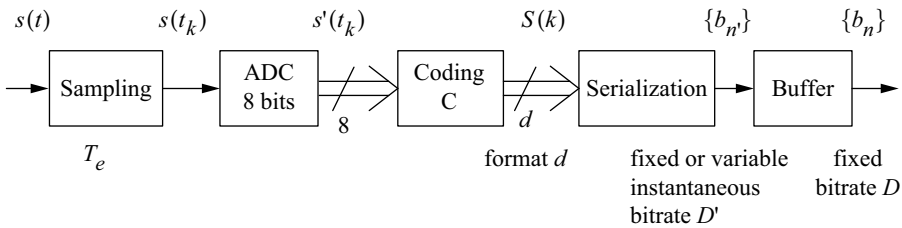


Figure 1.4. *General scheme of a digital transmission of a TV signal with information compression*

The analog signal (luminance component) is sampled with a sampling period $T_e = 100$ ns. In an analog/digital converter, each sample is then quantized linearly and converted to an integer $s'$of 8 bits (natural binary code). A coding block C converts this number of 8 bits into another binary codeword S of fixed or variable length d depending on the cases that we will examine. The codeword S of format d is then serialized and thus generates a bit stream with a fixed or variable bitrate D', depending on the case selected. A buffer is used to output a fixed bitrate D sequence such that D can be considered equal to $E[D']$ (E is the expected value). The transmission channel has a capacity of 34 Mbit/s of which only 32 Mbit/s can be used for the transmission of the video signal itself.

1) We first consider a very simplified version where the coding block C does not exist: the word $S(k)$ is strictly identical to the binary representation $s'(t_k)$ of the sampled signal $s(t_k)$.

What is the bitrate D' (in bit/s) at the output of the serialization block and the fixed bitrate D at the output of the buffer?

The bitrate D being considered too significant one seeks to reduce it. A Huffman encoding C_2 is used, constructed from the knowledge (by estimation) of the

amplitude probability law represented by the discrete random variable associated with s' is used. The entropy $H(s')$ is equal to 6.65 bit of information per amplitude and the efficiency η_2 of the code C_2 is 0.95.

2) What is the average length $\overline{l_2} = E(d)$ of the codewords S? Deduce the fixed bitrate D_2.

Since the bitrate is still too big, a differential pulse code modulation (DPCM) coding system with information compression type, shown in Figure 1.5, is used.

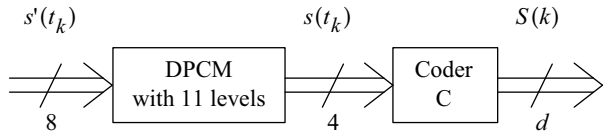


Figure 1.5. *Information compression using a DPCM system and a Huffman code* C_4

From a 256-level representation, the DPCM system generates a representation of $s(t_k)$ with 11 levels. The number s is represented according to a natural binary code.

3) We first consider in Figure 1.5 that the coding C does not exist. What are the bitrate D_3' at the output of the serialization block and the fixed bitrate D_3? Is it too large?

An alternative is now considered to further reduce the bitrate D_3. The coding C (called coding C_3) groups two consecutive symbols s to form bijectively a single codeword S (thus this one has a frequency half of that of s) : $\{s(t_{2k}), s(t_{2k+1})\} \leftrightarrow S(t_k)$.

4) Since the code C_3 does not use any statistical properties of s, show that the minimum length l_3 of the codeword s is 7 bits. What is the fixed bitrate D_{31}? Are we able to transmit the image on the transmission channel?

To further reduce the bitrate, a Huffman code C_4 is used as code C but without grouping the s symbols by two. The probabilities of realization of s are as follows:

$$Pr(s = s_1) = Pr(s = s_2) = Pr(s = s_{10}) = Pr(s = s_{11}) = 0.03125$$

$$Pr(s = s_3) = Pr(s = s_4) = Pr(s = s_8) = Pr(s = s_9) = 0.0625$$

$$Pr(s = s_5) = Pr(s = s_7) = 0.125$$

$$Pr(s = s_6) = 0.375$$

5) Design the Huffman code C_4. You will determine explicitly the codewords associated with each of the possible realizations of s.

6) What is the average length $\bar{l}_4$ of the codewords S. What is the efficiency η_4 of code C_4 since the entropy $H(s)$ is 2.905 bit/amplitude?

7) What is the fixed bitrate D_4? Are we able to transmit the image on the channel?

We want to protect the binary information transmitted against transmission errors. A coding block is used which adds a 16-bit protection to a useful 256-bit packet (packet error detector code).

8) What is the new average bitrate D_{41} and is it compatible with the capacity of the transmission channel?

Solution of problem 5

1) Bitrate at the output?

$$T_e = 100 \text{ ns} \rightarrow f_e = 10 \text{ MHz}$$

$$D' \text{ is fixed} \rightarrow D = D' = 8 \times 10^7 = 80 \text{ Mbit/s}$$

2) Average length of codewords and fixed bitrate?

$$H(s') = -\sum_{i=1}^{256} p(s_i') \log_2 p(s_i') = 6.65 \text{ bits of information/amplitude}$$

$$s \xrightarrow{Code\ C_2} S$$

$$\eta_2 = \frac{H(s')}{\bar{l}_2} \rightarrow \bar{l}_2 = \frac{H(s')}{\eta_2} = \frac{6.65}{0.95} = 7 \text{ bit/amplitude}$$

$$D_2 = \bar{l}_2 \times f_e = 7 \times 10^7 = 70 \text{ Mbit/s}$$

3) 11-level DPCM coding, thus 4 bits per sample are needed because $2^3 < 11 < 2^4$.

The bitrate D_3' is fixed, hence:

$$D_3' = D_3 = 4 \times 10^7 = 40 \text{ Mbit/s}$$

Yes, the bitrate D_3 is too large because it is greater than the capacity of the channel.

4) Coding C_3:

$$\{s_{2k}, s_{2k+1}\} \xrightarrow{Code\ C_3} S_k$$

The pair $\{s_{2k}, s_{2k+1}\}$ has $11 \times 11 = 121$ different configurations possible and since: $2^6 < 121 < 2^7$, 7 bits are necessary to encode a pair of samples, so:

$$l_3 = 7 \text{ bit/pair of samples.}$$

D_{31}' is fixed, hence:

$$D_{31} = D_{31}' = 7 \times \frac{f_e}{2} = 7 \times \frac{1}{2} \times 10^7 = 35 \text{ Mbit/s}$$

It is not possible to transmit the image on the channel because the bitrate D_{31} is greater than the capacity of the channel.

5) Huffman coding C_4.

6) Average length $\overline{l_4}$ and efficiency η_4:

$$\overline{l_4} = \sum_{i=1}^{11} p(s_i) \times l_i$$

$$\overline{l_4} = 0.375 \times 1 + 0.125 \times 3 + 0.125 \times 4 + 4 \times 0.0625 \times 4 + 4 \times 0.03125 \times 6 = 3 \text{ bit/codeword}$$

$$\eta_4 = \frac{H(s)}{\overline{l_4}} = 96.83\ \%$$

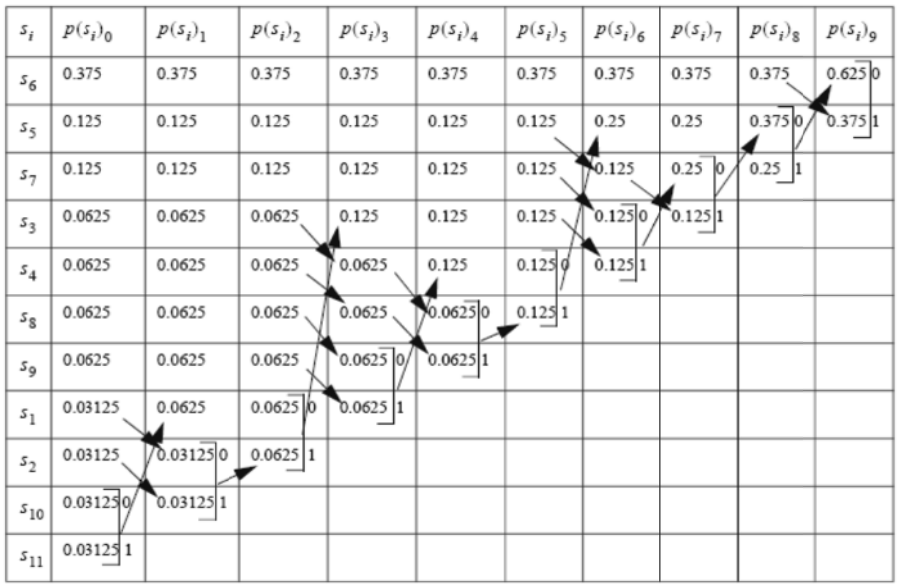
7) Fixed bitrate: $D_4 = \overline{l_4} \times f_e = 3 \times 10^7 = 30$ Mbit/s.

The bitrate D_4 is lower than the capacity of the channel, therefore we can transmit it on this communication channel.

8) Block coding protection:

$$D_{41} = D_4 \times \frac{272}{256} = 31.875 \text{ Mbit/s}$$

The D_{41} bitrate is less than the channel capacity, so it is compatible with the channel transmission.



s_i	s_6	s_5	s_7	s_3	s_4	s_8
Code C_4	1	001	0000	0110	0111	0100
s_i	s_9	s_1	s_2	s_{10}	s_{11}	
Code C_4	0101	000110	000111	000100	000101	

Table 1.5. *Construction of Huffman code C_4*

1.6. Problem 6 – Information, entropy, codes (1)

A color image coding system is considered for both storage and efficient transmission over a transmission channel. A bank of still images considered as an $S1$ source of information, are in VGA format (video graphics array) 640 × 480 pixels with only 16 color levels per pixel (luminance and chrominance jointly).

The statistics on this bank of images show that out of the 16 colors:

– 4 are used 60% of the time, with equal frequency;

– 4 others are used 30% of the time, with equal frequency;

– the others are used 10% of the time, also with equal frequency.

1) What is the amount Q_1 of binary information required to store an image with a fixed format binary code (code C_1)?

We want to reduce this amount by using a variable length code like a Huffman code.

2) Construct the code associated with this type of information (code C_2). For that, you can use a simple technique of grouping words to encode a class of words (important gain of time).

Deduce the average length $\overline{l_2}$, the amount Q_2 of binary information needed to store an image and the compression rate τ given by this code.

3) What is the entropy H of this source of information (per pixel)?

Deduce the efficiency η_2 of code C_2.

One wants to transmit the coded images with code C_2 to a recipient through a memoryless binary symmetric channel (BSC) having a fixed bitrate D. Let $S2$ be the binary information source that is at the serial output of the Huffman coding.

4) What are for $S2$ the probability p_0 to issue $x_i = 0$ and the probability p_1 to issue $x_i = 1$?

The transmission channel is a memoryless binary symmetric channel (BSC). It introduces transmission errors with an error probability p (the numerical application will be $p = 10^{-4}$).

5) Determine the entropies $H(X)$, $H(Y)$ and $H(Y/X)$.

6) Determine the amount of information received by the recipient for each binary symbol sent $I(X,Y)$, as well as the entropy $H(X/Y)$ (called ambiguity).

7) What is the average loss of information per image transmitted?

8) Determine the average number of received pixels per image, whose value is wrong.

9) Would it be possible to add a protection code after the coding C_2?

What do you suggest and justify your proposal?

Does it work at a codeword level or a block-code level?

Solution of problem 6

1) VGA image: $640 \times 480 = 307\,200$ pixels, 16 colors per pixel.

4 bit/pixel (because $16 = 2^4$) are necessary. Thus it needs:

$$Q_1 = 307{,}200 \times 4 = 1{,}228{,}800 \text{ bits} = 153{,}600 \text{ bytes}$$

2) The 16 colors are divided into 3 groups:

$$g_1 \leftrightarrow (c_0, \cdots, c_3)$$

$$g_2 \leftrightarrow (c_4, \cdots, c_7)$$

$$g_3 \leftrightarrow (c_8, \cdots, c_{15})$$

Construction of Huffman's code on groups: code C_2.

Group g	$p(g)_0$	$p(g)_1$	Code
g_1	0.6	0.6 ⌐ 0	0
g_2	0.3 ⌐ 0	0.4 ⌙ 1	1 0
g_3	0.1 ⌙ 1		1 1

Group g_1	Code
c_0	0 0 0
c_1	0 0 1
c_2	0 1 0
c_3	0 1 1

Group g_2	Code
c_4	1 0 0 0
c_5	1 0 0 1
c_6	1 0 1 0
c_7	1 0 1 1

Group g_3	Code
c_8	1 1 0 0 0
c_9	1 1 0 0 1
c_{10}	1 1 0 1 0
c_{11}	1 1 0 1 1
c_{12}	1 1 1 0 0
c_{13}	1 1 1 0 1
c_{14}	1 1 1 1 0
c_{15}	1 1 1 1 1

Table 1.6. *Construction of Huffman code C_2. For a color version of this table, see www.iste.co.uk/assad/digital2.zip*

So, the average length of this coding is:

$$\overline{l_2} = \sum_{i=0}^{15} p_i \times l_i = 4 \times \left(\frac{0.6}{4} \times 3\right) + 4 \times \left(\frac{0.3}{4} \times 4\right) + 8 \times \left(\frac{0.1}{8} \times 5\right)$$

$$= 3.5 \text{ bit/color } = 3.5 \text{ bit/pixel}$$

NOTE.– This average length would actually be equal to 3.45 bit/color (or pixel) for direct Huffman coding on colors c_0 to c_{15}.

Thus, with this code, one needs:

$$Q_2 = 307{,}200 \times \overline{l_2} = 307{,}200 \times 3.5 = 1{,}075{,}200 \text{ bits} = 134{,}400 \text{ bytes}$$

The compression rate τ is given by:

$$\tau = \frac{Q_1}{Q_2} = \frac{4}{3.5} = 1.142857$$

3) The entropy of this code is:

$$H(c) = -\sum_{i=1}^{15} p(c_i) \log_2 p(c_i)$$

$$H(c) = -\left\{4 \times \frac{0.6}{4} \log_2\left(\frac{0.6}{4}\right) + 4 \times \frac{0.3}{4} \log_2\left(\frac{0.3}{4}\right) + 8 \times \frac{0.1}{8} \log_2\left(\frac{0.1}{8}\right)\right\}$$

$$= 3.395462 \text{ bits of information/color}$$

$$= 3.395462 \text{ bits of information/pixel}$$

Its efficiency is:

$$\eta_2 = \frac{H(c)}{\overline{l_2}} = \frac{3.395462}{3.5} \cong 97\ \%$$

4) In the group g_1, there are 8 bits at 0 out of 12 bits.

In the group g_2, there are 8 bits at 0 out of 16 bits.

In the group g_3, there are 12 bits at 0 out of 40 bits.

So the probability of having a bit at 0 is:

$$p_0 = Pr\{x_i = 0\} = p(x_1) = 0.6 \times \frac{8}{12} + 0.3 \times \frac{8}{16} + 0.1 \times \frac{12}{40} = 0.58$$

and a bit at 1 is:

$$p_1 = Pr\{x_i = 1\} = p(x_2) = 1 - p_0 = 0.42$$

5) Recall that the source of information considered here is the binary source $S2$.

The three entropies are given successively by:

$$H(X) = -\sum_{i=1}^{2} p(x_i) \log_2 p(x_i) = -\{p_0 \log_2 p_0 + p_1 \log_2 p_1\}$$

$$H(X) \cong -1.44\{0.58 \times \log_e 0.58 + 0.42 \times \log_e 0.42\}$$
$$= 0.981454 \text{ bits of information/binary symbol}$$

$$H(Y) = -\sum_{j=1}^{2} p(y_j) \log_2 p(y_j)$$

with:

$$p(y_j) = \sum_{i=1}^{2} p(x_i) \times p(y_j/x_i)$$

$$p(y_1) = Pr\{y_j = 0\} = 0.58 \times (1 - p) + 0.42 \times p = 0.58 - 0.16p$$
$$= 0.579984$$

$$p(y_2) = Pr\{y_j = 1\} = 1 - p(y_1) = 0.420016$$

$$H(Y) \cong -1.44\{p(y_1) \times \log_e p(y_1) + p(y_2) \times \log_e p(y_2)\}$$
$$= 0.981461 \text{ bits of information/binary symbol}$$

$$H(Y/X = x_i) = -\sum_{j=1}^{2} p(y_j/x_i) \times \log_2 p(y_j/x_i)$$

and:

$$H(Y/X) = \sum_{i=1}^{2} p(x_i)\, H(Y/X = x_i)$$

Since we are dealing with a binary symmetric communication channel, we get:

$$H(Y/X) \cong -1.44\{(1-p)\log_e(1-p) + p\log_e p\} = H(p)$$

$$H(Y/X) \cong 1.4730335 \times 10^{-3} \text{ bits of information/binary symbol}$$

6) The amount of information transmitted is:

$$I(X,Y) = H(Y) - H(Y/X) = 0.9799879 \text{ bits of information/binary symbol}$$

$$H(X/Y) = H(X) - I(X,Y) = 1.46661 \times 10^{-3} \text{ bits of information/binary symbol}$$

7) The average loss of information per image is:

$$H(X/Y) \times Q_2 = 1{,}576.35 \text{ bits of information/image}$$

8) Average number of wrong pixels received:

– if transmission of group g_1: coding on 3 bits, $p(g_1) = 0.6$, then the probability of error-free transmission of the group's codewords g_1 is: $(1-p)^3$;

– if transmission of group g_2: coding on 4 bits, $p(g_2) = 0.3$, then the probability of error-free transmission of the group's codewords g_2 is: $(1-p)^4$;

– if transmission of group g_3: coding on 5 bits, $p(g_3) = 0.1$, then the probability of error-free transmission of the group's codewords g_3 is: $(1-p)^5$.

Then the probability of an error-free transmission of a pixel is:

$$Pr\{error - free\ pixel\} = 0.6 \times (1-p)^3 + 0.3 \times (1-p)^4 + 0.1 \times (1-p)^5$$

And if $p \ll 1 \rightarrow (1-p)^n \cong 1 - np$, hence:

$$Pr\{error - free\ pixel\} \cong 0.6 \times (1-3p) + 0.3 \times (1-4p) + 0.1 \times (1-5p)$$

$$\cong 1 - 3.5p$$

The probability of a transmission error of a pixel is then:

$$Pr\{\text{error of a pixel}\} \cong 3.5\, p$$

This result is quite logical since the average length of a codeword is:

$$\overline{l_2} = 3.5 \text{ bit/pixel}$$

The average number of erroneous pixels received per image is then:

$$307{,}200 \times 3.5\, p \cong 108 \text{ pixels}$$

9) The codewords $\in$ to the code C_2 are of variable lengths (3 or 4 or 5 bits), but the protection codes studied in this course are dependent on the length of the codewords, so the error correction will be difficult at the level of each codeword. This is why the protection (error correction) will be built at the level of blocks of bits instead of at the level of codewords.

1.7. Problem 7 – Information, entropy, codes (2)

Let us take a facsimile-type digitized image coding system, images with black parts on a white background (handwritten or printed text, diagram, graphic, etc.), for storage and efficient transmission on a communication channel. The scanned images are in 1,600 x 2,400 pixels format with 2 grey levels per pixel. Pixels here are considered to be independent in terms of random variables (it is a great simplification).

The statistics made on the facsimile images show that the 0 label pixels associated with white color are observed with a frequency equal to 0.9 and that 1 label pixels associated with black color are therefore observed with a frequency equal to 0.1.

1) What is the quantity Q_1 of binary information needed to store an image with a fixed format binary code (code C_1)? Can a Huffman coding of this 2-symbol information source reduce this quantity Q_1 and why?

NOTE.– In the Huffman codes that will be constructed later, the suffix 1 will always be used for the lowest probability element and the suffix 0 for the highest probability.

2) What is the entropy $H(S_1)$ of the source of information per pixel? Deduce the efficiency η_1 of the code C_1.

We want to increase efficiency by using a code associated not with each pixel but associated with each group of 2 pixels (second-order extension code).

3) Construct the Huffman code associated with this new source S_2 of information (code C_2). Deduce the average length $\overline{l_2}$, the quantity Q_2 of coding bits necessary for the storage of an image, the compression ratio τ_2 obtained by this code C_2 (with respect to the code C_1) and its efficiency η_2.

It is still necessary to increase the efficiency of the coding by using a code associated with each group of 3 pixels (code with an extension of order 3).

4) Construct the Huffman code associated with this new source S_3 of information (code C_3). Deduce the average length $\overline{l_3}$, the quantity Q_3 of binary information necessary for storing an image, the compression ratio τ_3 obtained by this code C_3 (still with respect to the code C_1) and its efficiency η_3.

One could thus go on increasing the number of grouped pixels to increase the efficiency of the coding.

5) What would be the compression ratio τ obtained by an almost infinite order extension code (very large in practice) with respect to the code C_1 and its efficiency η?

We want to transmit the coded images with the code C_3 to a recipient on a transmission line having a fixed bitrate D. Let S_3 be the source of binary information that we have at the serial output of the Huffman code C_3.

6) What is for code C_3, the probability p_0 to issue $x_i = 0$ and the probability p_1 to issue $x_i = 1$?

The transmission channel is a memoryless binary symmetric channel. It introduces transmission errors with a probability p (the numerical application will be: $p = 10^{-6}$).

7) Determine the entropies $H(X)$, $H(Y)$ and $H(Y/X)$.

8) Determine the amount of information received by the recipient for each binary symbol sent $I(X,Y)$, as well as the entropy $H(X/Y)$.

9) What is the average loss of information per image transmitted?

10) Determine the average number of pixels received per image, whose value is wrong.

Solution of problem 7

1) Image size: $N = 1{,}600 \times 2{,}400 = 3{,}840{,}000$ pixels.

In order to memorize an image with code C_1: 1 bit/pixel, is needed:

$$Q_1 = 1 \times N = 3{,}840{,}000 \text{ bits} = 480{,}000 \text{ bytes}$$

The use of a Huffman coding of the source S_1 with 2 symbols $\{s_1 = 0, s_2 = 1\}$, gives an average length of $\overline{l_1}$ = 1 bit/symbol, so there is no compression at all.

2) The entropy is given by:

$$H(S_1) = -\sum_{i=1}^{2} p(s_i) \log_2 p(s_i) \cong -1.44\{0.9 \times \log_e 0.9 + 0.1 \times \log_e 0.1\}$$

$$\cong 0.46812 \text{ bits of information/binary symbol}$$

and its efficiency is:

$$\eta_1 = \frac{H(S_1)}{\overline{l_1}} = 46.81\ \%$$

3) Second-order extension code: grouping 2 pixels together (symbol $s_{ij} = s_i s_j$) hence there are four possible events.

Huffman coding: code C_2

Symbols s_{ij}	$p(s_{ij})_0$	$p(s_{ij})_1$	$p(s_{ij})_2$	Code C_2
$s_{11} = s_1 s_1$	0.81	0.81	0.81 0	0
$s_{12} = s_1 s_2$	0.09	0.1 0	0.19 1	1 1
$s_{21} = s_2 s_1$	0.09 0	0.09 1		1 0 0
$s_{22} = s_2 s_2$	0.01 1			1 0 1

Table 1.7. *Construction of Huffman code C_2*

The average length of the codewords is:

$$\overline{l_2} = \sum_{i=1}^{4} p_i \times l_i = 0.81 \times 1 + 0.09 \times 2 + 0.09 \times 3 + 0.01 \times 3$$

$$= 1.29 \text{ bits/symbol}$$

The amount of coding bits needed to store an image is:

$$Q_2 = \frac{N}{2} \times \overline{l_2} = \frac{3{,}840{,}000}{2} \times 1.29 = 2{,}476{,}800 \text{ bits}$$

The compression rate τ_2 relative to code C_1 and its efficiency η_2 are respectively:

$$\tau_2 = \frac{Q_1}{Q_2} = \frac{\overline{l_1}}{\overline{l_2}/2} \cong 1.550387579$$

$$\eta_2 = \frac{H(S_2)}{\overline{l_2}}$$

and:

$$H(S_2) = H(S_1^2) = 2H(S_1) = 0.93624 \text{ bit of information/symbol}$$

hence:

$$\eta_2 = \frac{H(S_2)}{\overline{l_2}} = \frac{0.93624}{1.29} \cong 72.57\ \%$$

4) Third-order extension code C_3: grouping of 3 pixels together (symbol $s_{ijk} = s_i s_j s_k$), so there are eight possible events.

The average length of the codewords is:

$$\overline{l_3} = \sum_{i=1}^{8} p_i \times l_i = 0.729 \times 1 + 3 \times 0.081 \times 3 + 3 \times 0.009 \times 5 + 0.001 \times 5$$

$$= 1.598 \text{ bits/symbol}$$

$Sl = s_{ijk}$	$p(Sl)_0$	$p(Sl)_1$	$p(Sl)_2$	$p(Sl)_3$	$p(Sl)_4$	$p(Sl)_5$	$p(Sl)_6$	Code C_3
$S1 = s_{000}$	0.729	0.729	0.729	0.729	0.729	0.729	0.729 0	0
$S2 = s_{001}$	0.081	0.081	0.081	0.081	0.109	0.162 0	0.271 1	1 0 0
$S3 = s_{010}$	0.081	0.081	0.081	0.081	0.081 0	0.109 1		1 0 1
$S4 = s_{100}$	0.081	0.081	0.081	0.081 0	0.081 1			1 1 0
$S5 = s_{110}$	0.009	0.01	0.081 0	0.028 1				11100
$S6 = s_{101}$	0.009	0.009 0	0.01 1					11101
$S7 = s_{011}$	0.009 0	0.009 1						11110
$S8 = s_{111}$	0.001 1							11111

Table 1.8. *Construction of Huffman code* C_3

The amount of coding bits needed to store an image is:

$$Q_3 = \frac{N}{3} \times \overline{l_3} = \frac{3{,}840{,}000}{3} \times 1.598 = 2{,}045{,}440 \text{ bits}$$

The relative compression rate τ_3 and the efficiency η_3 compared to the code C_1 are respectively:

$$\tau_3 = \frac{Q_1}{Q_3} = \frac{\overline{l_1}}{\overline{l_3}/3} = \frac{1}{1.598/3} = 1.877346683$$

$$\eta_3 = \frac{H(S_3)}{\overline{l_3}} = \frac{3H(S_1)}{1.598} = 87.88\,\%$$

5) Quasi infinite-order extension code: $n \to \infty \to \overline{l} = \overline{l_{min}} = H(S_1)$ hence the compression rate and efficiency, respectively:

$$\tau = \frac{\overline{l_1}}{H(S_1)} = \frac{1}{0.46812} = 2.136204392$$

$$\eta = 1$$

6) Probability p_0 to issue 0 and p_1 to issue 1 for the code C_3:

$$p_0 = Pr\{x_i = 0\} = p(x_1)$$

$$= 1 \times 0.729 \times + \frac{2}{3} \times 0.081 + 2 \times \frac{1}{3} \times 0.081 + \frac{2}{5} \times 0.009 + 2 \times \frac{1}{5} \times 0.009$$

$$= 0.8442$$

$$p_1 = Pr\{x_i = 1\} = p(x_2) = 1 - p_0 = 0.1558$$

7) The three entropies are successively the following:

$$H(X) = -\sum_{i=1}^{2} p(x_i) \log_2 p(x_i) = -\{p_0 \log_2 p_0 + p_1 \log_2 p_1\}$$

$$\cong -1.44\{0.8442 \times \log_e 0.8442 + 0.1558 \times \log_e 0.1558\}$$

$$\cong 0.6230004 \text{ bits of information/binary symbol}$$

$$H(Y) = -\sum_{j=1}^{2} p(y_j) \log_2 p(y_j)$$

with:

$$p(y_j) = \sum_{i=1}^{2} p(x_i) \times p(y_j/x_i)$$

$$p(y_1) = Pr\{y_j = 0\} = 0.8442 \times (1 - p) + 0.1558p = 0.8442 - 0.6884p$$

$$= 0.8441993$$

$$p(y_2) = Pr\{y_j = 1\} = 1 - p(y_1) = 0.1558006$$

$$H(Y) \cong -1.44\{p(y_1) \times \log_e p(y_1) + p(y_2) \times \log_e p(y_2)\}$$

$$= 0.623002 \text{ bits of information/binary symbol}$$

$$H(Y/X = x_i) = -\sum_{j=1}^{2} p(y_j/x_i) \times \log_2 p(y_j/x_i)$$

and:

$$H(Y/X) = \sum_{i=1}^{2} p(x_i)\, H(Y/X = x_i)$$

Since we are dealing with a binary symmetric communication channel, we get:

$$H(Y/X) \cong -1.44\{(1-p)\log_e(1-p) + p\log_e p\} = H(p)$$

$$H(Y/X) \cong 2.1334334 \times 10^{-5} \text{ bits of information/binary symbol}$$

8) Amount of information received by the recipient and entropy (ambiguity):

$$I(X,Y) = H(Y) - H(Y/X) = 0.6229806 \text{ bits of information/binary symbol}$$

$$H(X/Y) = H(X) - I(X,Y) = 1.98 \times 10^{-5} \text{ bits of information/binary symbol}$$

9) The average loss of information per image is:

$$H(X/Y) \times Q_3 = 1.98 \times 10^{-5} \times 2{,}045{,}440 = 40.499712 \text{ bits of information/image}$$

10) Group g_1: transmission of symbol $S1$; coding on 1 bit, $p(g_1) = 0.729$.

Then, the probability of error-free transmission of the codeword $S1$ of group g_1 is: $(1-p)$.

Group g_2: transmission of symbols $S2$ or $S3$ or $S4$; coding on 3 bits, $p(g_2) = 3 \times 0.081 = 0.243$.

Then, the probability of error-free transmission of the codewords of group g_2 is: $(1-p)^3$.

Group g_3: transmission of symbols $S5$ or $S6$ or $S7$ or $S8$; coding on 5 bits, $p(g_3) = 3 \times 0.009 + 1 \times 0.001 = 0.028$.

Then, the probability of error-free transmission of the codewords of group g_3 is: $(1-p)^5$.

So the probability of error-free transmission of a 3-pixel packet is:

$$Pr\{3 \text{ error} - \text{free pixels}\} = 0.729 \times (1-p) + 0.243 \times (1-p)^3 + 0.028 \times (1-p)^5$$

And, if $p \ll 1 \rightarrow (1-p)^n \cong 1 - np$, hence:

$$Pr\{3 \text{ error} - \text{free pixels}\} \cong 0.729 \times (1-p) + 0.243 \times (1-3p) + 0.028 \times (1-5p) \cong 1 - 1.598\,p$$

The probability of error in the transmission of a packet of 3 pixels is then:

$$Pr\{3 \text{ wrong pixels}\} \cong 1.598\, p$$

This result is quite logical since the average length of the codewords is:

$$\overline{l_3} = 1.598 \text{ bit/3 pixels}$$

The average number of erroneous pixels received per image is then:

$$\frac{N}{3} \times 1.598\, p = \frac{3{,}840{,}000}{3} \times 1.598 \times 10^{-6} = 2.045 \text{ pixels} \cong 3 \text{ pixels}$$

1.8. Problem 8 – Coding and transmission of a television-type information source

Let us take a coding system of analog television signal. The analog color TV signals are digitized and encoded in (4:2:2) format. However, to simplify the problem discussed here, only the luminance component will be considered. This leads to having per frame:

– 576 useful lines, with 720 pixels of luminance per line (rectangular sampling structure);

– 25 frames per second (50 fields per second);

– with 256 gray levels per monochrome pixel (initial binary coding on 8 bits).

To simplify, we consider that the pixel levels are independent (in terms of random variables), denoted by: "U".

However, the probability law $Pr(U)$ of grey levels U is absolutely non-uniform.

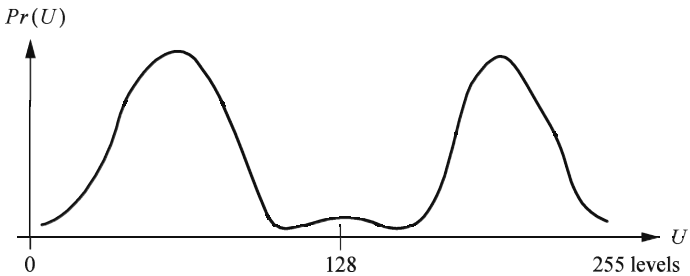


Figure 1.6. *Probability law $Pr(U)$ of TV frames grey levels U*

1) What is the amount Q_0 of binary symbols needed to store a second of monochrome TV frames with the initial fixed format binary code (code C_0)? We assume that the entropy $H(U) = 6$ bits of information/pixel. Deduce the efficiency η_0 of the code C_0.

2) Can a Huffman coding (called code C_1) of this source S_0 of information reduce this amount and why?

If we consider that the Huffman coding C_1 performs the absolute optimal coding, deduce the quantity Q_1 of binary symbols necessary to store a second of digital monochrome TV frames.

We are looking at increasing the efficiency by using an information compression of the source S_0. For this, an adaptive (and thus non-linear) re-quantization of the 256 grey levels U of each pixel is carried out on 8 levels, now denoted "Z". The probability law $Pr(Z)$ resulting from this new source of information (denoted S_2) and its binary code C_2 are shown in Table 1.9.

3) What is the amount Q_2 of binary symbols needed to store one second of digital monochrome TV frames (code C_2)?

Grey levels of Z	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
$\boldsymbol{Pr(Z)}$	0.0625	0.0625	0.15	0.21	0.14	0.0625	0.25	0.0625
Code C_2	000	001	010	011	100	101	110	111

Table 1.9. *Probability law $Pr(Z)$ of S_2 and binary code C_2*

4) Can a Huffman coding (called code C_3) of this source of information reduce this amount? What is the entropy per pixel and the total entropy of one second of digital TV frames?

5) Design the Huffman code associated with this new source S_2 (code C_3).

NOTE.– In the Huffman code that will be designed, the suffix 1 will always be used for the element with the lowest probability and therefore the suffix 0 for the element with the highest probability.

6) From the Huffman code C_3, deduce:

– the average number of coding bits per pixel;

– the amount Q_3 of binary symbols needed to store one second of digital monochrome TV frames;

– the compression rate τ_3 obtained by this code C_3 (with respect to the code C_2);

– its efficiency η_3 and redundancy ρ_3.

It is desired to transmit the coded frames with the code C_3 to a recipient over a digital transmission line, having a given capacity, denoted "*Cap*". Let S_3 be the source of binary information X that we have at the serial output of the Huffman coding.

7) What is the probability p_0 to issue $x_i = 0$ and the probability p_1 to issue $x_i = 1$ for S_3? Deduce its entropy $H(X)$.

The transmission channel is a memoryless binary symmetric channel. It introduces transmission errors with a probability p (the numerical application will be $p = 10^{-2}$). The output of the binary transmission channel is called Y when its input is X (the binary output of the source S_3).

8) What is the entropy of Y. Deduce the amount of information $I(X,Y)$ received by the recipient for each binary symbol sent by S_3.

9) What is the average loss of information in the channel per binary symbol sent $H(X/Y)$ and the average loss of information per second of transmitted TV frames?

10) Determine the average number of received pixels per second of TV frames whose value is wrong.

11) What is the capacity Cap of the binary transmission channel and the capacity Cap_s per second of TV frames?

We model $Pr(U)$ by a weighted sum (factors λ_a and λ_b respectively, with $\lambda_a = 0.6225$ and so $\lambda_b = 1 - \lambda_a$) of two discrete Gaussian probability laws G_a and G_b, with : $G_a(U) = \text{Gauss}(64, 8)$ and $G_b(U) = \text{Gauss}(160, 4)$ where, in $\text{Gauss}(m, \sigma)$, m is the mean value and σ is the standard deviation of the Gaussian probability law.

12) What is (with justification) among the eight following values: 8; 7; 6; 5; 4; 3; 2 and 1 bit/pixel the order of magnitude of the entropy H_0 of the source information S_0 per pixel?

Solution of problem 8

1) The image size is: $N = 576 \times 720 = 414{,}720$ pixels.

One second of digital TV frames has:

$Ns = N \times 25 = 10{,}368{,}000$ pixel/s

Monochrome TV: $\overline{l_0} = 8$ bits/pixel

Code C_0: $Q_0 = Ns \times \overline{l_0} = 82{,}944{,}000$ bits/s

Efficiency: $\eta_0 = \frac{H(U)}{\overline{l_0}} = \frac{6}{8} = 75\ \%$

2) The source of information generating U is non-uniform on [0, 255], so:

$H(U) < 8$ bits of information/pixel

Therefore, the entropy coding is quite interesting.

If the Huffman code C_1 performs an absolute optimal coding, then $\overline{l_1} = H(U)$, hence:

$Q_1 = Ns \times \overline{l_1} = 62{,}208{,}000$ bits/s

3) The amount of bits per second for the code C_2 is:

$Q_2 = Ns \times \overline{l_2} = Ns \times 3 = 31{,}104{,}000$ bit/s

4) Since Z has a non-uniform probability law, then the Huffman coding is efficient.

The entropy per pixel is:

$$H(Z) = -\sum_{i=0}^{7} p_i \log_2 p_i$$

$$H(Z) \cong -1.44 \left\{ \begin{matrix} 4 \times 0.0625 \times \log_2(0.0625) + 0.15 \times \log_2(0.15) \\ +0.21 \times \log_2(0.21) + 0.14 \times \log_2(0.14) + 0.25 \times \log_2(0.25) \end{matrix} \right\}$$

$\cong 2.7804781$ bits of information/pixel

The total entropy of one second of monochrome TV frames is:

$$H(TVs) = Ns \times H(Z) = 28{,}827{,}997 \text{ bits of information/s}$$

5) Huffman coding: code C_3.

Level Z	$p(Z)_0$	$p(Z)_1$	$p(Z)_2$	$p(Z)_3$	$p(Z)_4$	$p(Z)_5$	$p(Z)_6$	Code C_3
6	0.25	0.25	0.25	0.25	0.29	0.46	0.54 0	0 1
3	0.21	0.21	0.21	0.25	0.25	0.29 0	0.46 1	1 1
2	0.15	0.15	0.15	0.21	0.25 0	0.25 1		0 0 0
4	0.14	0.14	0.14	0.15 0	0.21 1			0 0 1
0	0.0625	0.125	0.125 0	0.14 1				1 0 1 0
1	0.0625	0.0625 0	0.125 1					1 0 1 1
5	0.0625 0	0.0625 1						1 0 0 0
7	0.0625 1							1 0 0 1

Table 1.10. *Construction of Huffman code* C_3

6) The average codewords length is:

$$\overline{l_3} = \sum_{i=1}^{8} p_i \times l_i = 0.25 \times 2 + 0.21 \times 2 + 0.15 \times 3 + 0.14 \times 3 + 4 \times 0.0625 \times 4$$

$$= 2.79 \text{ bit/grey level} = 2.79 \text{ bits/pixel}$$

The number of bits per second is:

$$Q_3 = Ns \times \overline{l_3} = 28{,}926{,}720 \text{ bits/s}$$

The compression ratio, efficiency and redundancy are respectively:

$$\tau_3 = \frac{Q_2}{Q_3} = \frac{\overline{l_2}}{\overline{l_3}} = \frac{3}{2.79} = 1.07527$$

$$\eta_3 = \frac{H(Z)}{\overline{l_3}} = \frac{2.7804781}{2.79} \cong 99.66\ \%$$

$$\rho_3 = 1 - \eta_3 = 0.0034128$$

7) Probability p_0 to issue $x_i = 0$ and probability p_1 to issue $x_i = 1$ for S_3?

The probability of sending a bit at zero is given by:

$$p_0 = Pr\{x_i = 0\} = p(x_1) = 0.25 \times \frac{1}{2} + 0.21 \times \frac{0}{2} + 0.15 \times \frac{3}{3} + 0.14 \times \frac{2}{3} + 0.0625$$

$$\times \frac{2}{4} + 0.0625 \times \frac{1}{4} + 0.0625 \times \frac{3}{4} + 0.0625 \times \frac{2}{4} = 0.4933$$

and that of sending a bit at 1 is therefore:

$$p_1 = Pr\{x_i = 1\} = p(x_2) = 1 - p_0 = 0.5067$$

The entropy is given by:

$$H(X) = -\sum_{i=1}^{2} p(x_i) \log_2 p(x_i) = -\{p_0 \log_2 p_0 + p_1 \log_2 p_1\}$$

$$\cong -1.44\{0.4933 \times \log_e 0.4933 + 0.5067 \times \log_e 0.5067\}$$

$$= 0.99987 \text{ bits of information/binary symbol}$$

8) The entropy of Y at the output of the communication channel is:

$$H(Y) = -\sum_{j=1}^{2} p(y_j) \log_2 p(y_j)$$

with:

$$p(y_j) = \sum_{i=1}^{2} p(x_i) \times p(y_j/x_i)$$

$$p(y_1) = Pr\{y_j = 0\} = 0.4933 \times (1 - p) + 0.5067 \times p = 0.493434$$

$$p(y_2) = Pr\{y_j = 1\} = 1 - p(y_1) = 0.506566$$

$$H(Y) \cong -1.44\{p(y_1) \times \log_e p(y_1) + p(y_2) \times \log_e p(y_2)\}$$

$$= 0.99800773 \text{ bits of information/binary symbol}$$

The amount of information transmitted through the channel is:

$$I(X,Y) = H(Y) - H(Y/X) = H(X) - H(X/Y)$$

since we are dealing with a binary symmetric communication channel, we have:

$$H(Y/X) = H(Y/X = x_i) = -\sum_{j=1}^{2} p(y_j/x_i) \times \log_2 p(y_j/x_i)$$

$$H(Y/X) \cong -1.44\{(1-p)\log_e(1-p) + p\log_e p\} = H(p)$$

$$H(Y/X) \cong 0.080642209 \text{ bits of information/binary symbol}$$

hence:

$$I(X,Y) = H(Y) - H(Y/X) = 0.917365521 \text{ bits of information/binary symbol}$$

9) The average loss of information in the channel per binary symbol sent is given by:

$$H(X/Y) = H(X) - I(X,Y) = 0.082504479 \text{ bits of information/binary symbol}$$

The average loss of information per second of transmitted TV frames is:

$$Ns \times \overline{l_3} \times H(X/Y) = 2{,}386{,}583.963 \text{ bits of information/s}$$

10) We have to consider each of the codeword lengths:

– the group g_1 corresponds to the set of levels $\{0, 1, 5, 7\}$ each of which is coded on 4 bits and of probability equal to 0.0625, hence:

$$p(g_1) = p(0) + p(1) + p(5) + p(7) = 0.25$$

– the group g_2 corresponds to the set of levels $\{2, 4\}$ each of which is coded on 3 bits and of probability equal to 0.15 and 0.14 respectively, hence:

$$p(g_2) = p(2) + p(4) = 0.15 + 0.14 = 0.29$$

– the group g_3 corresponds to the set of levels $\{3, 6\}$ each of which is coded on 2 bits and of probability equal to 0.21 and 0.25 respectively, hence:

$$p(g_3) = p(3) + p(6) = 0.21 + 0.25 = 0{,}46$$

The probability of error-free transmission of:

g_1 is $(1-p)^4$; g_2 is $(1-p)^3$; g_3 is $(1-p)^2$

Thus, the probability of having no error with the code C_3 is:

$$Pr\{\text{no error}\} = 0.25 \times (1-p)^4 + 0.29 \times (1-p)^3 + 0.46 \times (1-p)^2$$

But if $p \ll 1 \rightarrow (1-p)^n \cong 1 - np$, then:

$$Pr\{\text{no error}\} \cong 0.25 \times (1-4\,p) + 0.29 \times (1-3\,p) + 0.46 \times (1-2\,p) \cong 1 - 2.79\,p$$

The probability of having at least one error with the code C_3 is then:

$$Pr\{\text{error}\} = 1 - Pr\{\text{no error}\} = 2.79\,p = 0.0279$$

The average number of wrong pixels received per second of frames is:

$$Ns \times 0.0279 = 289{,}267 \text{ pixels}$$

11) Since the transmission channel is binary symmetric:

$$Cap = Max\, I(X,Y) = 1 - H(p)$$
$$= 0.91935779 \text{ bits of information/binary symbol}$$

and:

$$Cap_s = Cap \times H(TVs) = 26{,}503{,}243 \text{ bits of information/s}$$

12) We have:

$$Pr(U) = \lambda_a G_a(U) + (1-\lambda_a) G_b(U)$$

If we consider that a Gaussian law has a practical range of $\pm 3\,\sigma$ around its mean value m, then:

$$H(U) = \lambda_a H(U_1) + (1-\lambda_a) H(U_2) + H(\lambda_a)$$

with U_1, the random variable associated with the Gaussian law is (64, 8) and U_2, the random variable associated with the Gaussian law is (160, 4). In addition, we have:

$H(U_1) < H(U_1')$: entropy of a uniform law on $[64 - (3 \times 8), 64 + (3 \times 8)] = [40, 88]$

$H(U_2) < H(U_2')$: entropy of a uniform law on $[160 - (3 \times 4), 160 + (3 \times 4)] = [148, 172]$

$H(U_1') = \log_2(88 - 40) = \log_2(48) = 5.585$

$H(U_2') = \log_2(172 - 148) = \log_2(24) = 4.585$

Consequently, we have:

$$H(U) < \lambda_a \times 5.585 + (1 - \lambda_a) \times 4.585 + H(\lambda_a)$$

with:

$$H(\lambda_a) = 0.9544$$

hence:

$H(U) < 6.1619$ bits of information/pixel $\rightarrow H(U) \leq 6$ bits of information/pixel

So, the order of magnitude of the entropy is:

$H(U) \cong 6$ bits of information/pixel.

1.9. Problem 9 – Entropy and motion information encoding of multimedia source

The context is that of the transmission of coded video. Several categories of information are represented and coded in a compressed form. One of these categories is motion information. Each frame I_t of a sequence SI of frames:

$$SI\{\cdots, I_{t-1}, I_t, I_{t+1}, \cdots\}$$

is divided into K macro-blocks MB_k of size 16 x 16 pixels (we have: $k = 1, \cdots, K$).

The sequence SI consists of L frames per second (typically in Europe L = 25). Each macro-block MB is associated with a displacement vector $\vec{D}$ which makes it possible to predict its content from previous frame(s). $\vec{D}$ is a vector with two components: dx and dy, taking their values on integers and half integers. For simplicity, it is assumed that in practice only seven values for dx and dy,

respectively, are of significant probabilities. Each value of dx and dy is associated with a symbol s. These values are given in Table 1.11 with their probabilities (for the sake of simplicity, it has been assumed that the components dx and dy of the displacement vector $\vec{D}$ have the same statistics. This is not really the case).

Value	-1.5	-1	-0.5	0	0.5	1	1.5
Symbol	s_1	s_2	s_3	s_4	s_5	s_6	s_7
Probability	0.014	0.024	0.117	0.701	0.101	0.027	0.016

Table 1.11. *Probabilities of a component d of motion vector $\vec{D}$*

1) Determine the entropy $H(d)$ of a component dx or dy of the displacement vector $\vec{D}$. Deduce the entropy $H(\vec{D})$ of the displacement vector $\vec{D}$ for a separate coding of dx and of dy. What would be the efficiency η_1 of a fixed length code C_1 (length L_1) coding $\vec{D}$ and the bitrate DB_1 per second for a number $K = 396$ macro-blocks per frame and $L = 25$ frame/second for coding the vectors $\vec{D}$?

2) Taking code C_1 the natural binary coding in the ascending order of the symbols s_i, determine the probability p_0 of having a bit at zero in the bitstream encoding the displacement vector $\vec{D}$. Deduce the probability p_1 of having a bit at one.

3) Construct the Huffman code C_2 giving the codeword S_i associated with each of the symbols s_i of a vector component $\vec{D}$.

NOTE.– In the construction of the code C_2, the coding suffix associated with the element of lowest probability will be systematically set to 0.

Deduce from this: the average length $\overline{l_2}$ of the codewords of C_2, the average length $\overline{L_2}$ of the codewords encoding the vector $\vec{D}$ (again with a separate coding of dx and dy), its efficiency η_2 and the average bitrate per second DB_2 for coding the vectors $\vec{D}$.

4) It is considered that the source S delivers the following SS time sequence of symbols s:

$$\cdots\cdots s_4 \quad s_2 \quad s_4 \quad s_4 \quad s_3 \quad s_5 \quad s_4 \quad s_6 \quad s_4 \quad s_7 \cdots\cdots \rightarrow \text{time}$$

Deduce the corresponding sequence SB of bits obtained at the output of the Huffman coding C_2. Sequence SB is of the form $\{\cdots, b_{k-1}, b_k, b_{k+1}, \cdots\}$.

What do you observe?

Taking code C_2, determine the probability p_0 to have a bit at zero in the bit stream encoding the displacement vector $\vec{D}$. Deduce the probability p_1 of having a bit at one.

Solution of problem 9

1) The entropy of a component of the motion vector $\vec{D}$ is given by:

$$H(d) = -\sum_{i=1}^{7} p(s_i) \times \log_2 p(s_i)$$

$$H(d) \cong -1.44 \times \{0.014 \times \log_e(0.014) + 0.024 \times \log_e(0.024) + 0.117$$
$$\times \log_e(0.117) + 0.701 \times \log_e(0.701) + 0.101$$
$$\times \log_e(0.101) + 0.027 \times \log_e(0.027) + 0.016 \times \log_e(0.016)\}$$
$$\cong 1.499 \text{ bits of information/component}$$

The components dx and dy have the same statistics and are coded separately, hence:

$$H(D) = 2 \times H(d) = 2.998 \text{ bits of information/vector } \vec{D}$$

There are seven values possible per dx or dy component, so for fixed length coding, it takes 3 bits to encode dx and 3 bits to encode dy. So, a total of:

$$L_1 = 6 \text{ bits}$$

to encode each displacement vector $\vec{D}$.

The efficiency of this simple encoding technique is then:

$$\eta_1 = \frac{H(D)}{L_1} = \frac{2.998}{6} \cong 50\ \%$$

and the bitrate for coding one second of $\vec{D}$ is:

$$DB_1 = L_1 \times K \times L = 6 \times 396 \times 25 = 59{,}400 \text{ bits}$$

2) Since dx and dy have the same statistics, it is sufficient to consider only one component to determine the probability p_0 of having a bit at zero.

s_i	s_1	s_2	s_3	s_4	s_5	s_6	s_7
Code C_1	000	001	010	011	100	101	110
Number of 0	3	2	2	1	2	1	1

Table 1.12. *Code C_1 and number of 0 in each codeword*

$$p_0 = \sum_{i=1}^{7} p(s_i) \times \frac{\text{Number of zeros in } s_i}{l_i}$$

$$p_0 = \left\{0.014 \times \frac{3}{3} + 0.024 \times \frac{2}{3} + 0.117 \times \frac{2}{3} + 0.701 \times \frac{1}{3} + 0.101 \times \frac{2}{3} + 0.027 \times \frac{1}{3} + 0.016 \times \frac{1}{3}\right\} = 0.4233$$

The probability p_1 of having a bit at 1 is then:

$$p_1 = 1 - p_0 = 0.5767$$

3) Huffman coding:

s_i	$p(s_i)_0$	$p(s_i)_1$	$p(s_i)_2$	$p(s_i)_3$	$p(s_i)_4$	$p(s_i)_5$	C_2
s_4	0.701	0.701	0.701	0.701	0.701	0.701 1	1
s_3	0.117	0.117	0.117	0.117	0.182 1	0.299 0	0 0
s_5	0.101	0.101	0.101	0.101 1	0.117 0		0 1 1
s_6	0.027	0.03	0.051 1	0.081 0			0 1 0 1 1
s_2	0.024	0.027 1	0.03 0				0 1 0 1 0
s_7	0.016 1	0.024 0					0 1 0 0 1
s_1	0.014 0						0 1 0 0 0

Table 1.13. *Construction of Huffman code C_2 of a component of vector $\vec{D}$*

The average length of the codeword (one component d of the motion vector only) is:

$$\overline{l_2} = \sum_{i=1}^{7} p(s_i) \times l_2(i)$$

$$\overline{l_2} = \{0.014 \times 5 + 0.024 \times 5 + 0.117 \times 2 + 0.701 \times 1 + 0.101 \times 3 + 0.027 \times 5 + 0.016 \times 5\} = 1.643 \text{ bits/codeword}$$

So, per a full motion vector:

$$\overline{L_2} = 2 \times \overline{l_2} = 3.286 \text{ bits/vector } \vec{D}$$

Its efficiency is:

$$\eta_2 = \frac{H(D)}{\overline{L_2}} = \frac{2.998}{3.286} = 91.23\ \%$$

And per second of TV frames, an average rate of:

$$DB_2 = \overline{L_2} \times K \times L = 3.286 \times 396 \times 25 = 32{,}531.2 \text{ bits/s} \cong 32{,}531 \text{ bits/s}$$

4) From the coded sequence (Table 1.14), there are rapid changes in the length of the codewords from one motion vector to the next.

SS	s_4	s_2	s_4	s_4	s_3	s_5	s_4	s_6	s_4	s_7
SB	1	01010	1	1	00	011	1	01011	1	01001

Table 1.14. *Coding of a sequence of a motion vector component*

The probability of having a bit at 0 is:

$$p_0 = \sum_{i=1}^{7} p(s_i) \times \frac{\text{Numbre of zeros in } s_i}{l_i}$$

$$p_0 = \left\{0.014 \times \frac{4}{5} + 0.024 \times \frac{3}{5} + 0.117 \times \frac{2}{2} + 0.701 \times \frac{0}{1} + 0.101 \times \frac{1}{3}\right.$$

$$+0.027 \times \frac{2}{5} + 0.016 \times \frac{3}{5}\Big\} = 0.1966$$

The probability of having a bit at 1 is then:

$$p_1 = 1 - p_0 = 0.8034$$

1.10. Problem 10 – Hamming coding

The problem of coding binary words for protection against transmission errors is tackled. The code C considered here is a single error correcting Hamming code. We successively call:

– i^t: binary word of information to be transmitted, of length m:

$$i^t = [i_1, i_2, \cdots, i_m]$$

– u^t: binary codeword resulting from the coding of i^t, of length n:

$$u^t = [u_1, u_2, \cdots, u_n]$$

– v^t: binary word obtained at the output of the transmission channel associated with u^t transmitted in the channel, also of length n:

$$v^t = [v_1, v_2, \cdots, v_n]$$

The construction of the codeword u^t from the information word i^t is done by using the generator matrix $[G]$ of the code C:

$$u^t = i^t \times [G]$$

The decoding of the word v^t received is carried out in two phases:

a) the detection of a possible transmission error and correction of the error;

b) the decoding by itself.

In the first phase, we calculate the syndrome s'^t on the word received:

$$s'^t = v^t \times [H']^t$$

Where $[H']$ is the parity matrix of the code C associated with the matrix $[G]$ *via* the matrix $[H]$. The latter is obtained from $[H']$ by permutation of columns to satisfy the form of $[H]$.

The syndrome makes it possible to detect the presence of a transmission error and to locate its position in the word received.

The characteristics imposed on the code C are as follows:

– correction of single errors;

– it is a systematic code;

– it is a Hamming code, with $m = 4$.

1) What is the minimum distance of this code?

2) Show that the length n of the codewords is 7.

3) Deduce that the generator matrix of the code is of the form:

$$[G] = \left[I_{4,4} \mid P_{4,3}\right]$$

where $I_{4,4}$ is the identity matrix.

4) Show that the parity matrix of the code is of the form:

$$[H] = \left[P_{4,3}^{t} \mid I_{3,3}\right]$$

5) Show that the presence of a transmission error on the j^{th} bit of u^t generates a syndrome s'^t equal to the j^{th} line h'_j of $[H']^t$ (h'_j is the natural binary representation of the number j).

6) Determine the matrices $[H']^t$ and $[H]$.

7) Determine the generator matrix $[G]$ of the code.

8) Construct the codewords u^t corresponding to the three information words:

$$i^t = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

We receive the three following words:

$$v^t = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

9) Check each of the words for a membership or non-membership to the code.

10) Make a block diagram of the encoder and the decoder.

Solution of problem 10

1) The minimum distance of the code is given by:

$$d_{min} = 2r + 1; \quad (r = 1)\text{: single errors, 1 wrong bit} \rightarrow d_{min} = 3$$

2) Length of Hamming code (corrector of one erroneous bit):

$$2^k \geq 1 + n \text{ ; with } n = m + k \rightarrow 2^k \geq 1 + m + k$$

$$\rightarrow 2^k \geq 5 + k \rightarrow k = 3 \text{ and } n = 4 + 3 = 7$$

with:

– n: number of bits of a codeword;

– m: number of bits of an information word;

– k: number of bits of a control word.

3) It is a systematic code:

$$u^t = i^t \times [G] = [i^t, i^t \times [P]\,] \rightarrow [G]_{4,7} = \left[I_{4,4} \mid P_{4,3}\right]$$

4) We should have:

$$[G]_{4,7} \times [H]^t_{3,7} = [0]_{4,3} \text{ ; and also : } [G]_{4,7} \times [H']^t_{3,7} = [0]_{4,3}$$

$$\rightarrow \left[I_{4,4} \mid P_{4,3}\right] \times [H]^t_{3,7} = [0]_{4,3}$$

Note that: $[M]^t_{3,7}$ means $\left[[M]_{3,7}\right]^t$.

If:

$$[H]^t_{3,7} = \begin{bmatrix} P_{4,3} \\ -- \\ I_{3,3} \end{bmatrix}$$

then:

$$\left[I_{4,4} \mid P_{4,3}\right] \times \begin{bmatrix} P_{4,3} \\ -- \\ I_{3,3} \end{bmatrix} = [P]_{4,3} \oplus [P]_{4,3} = [0]_{4,3}$$

$$\rightarrow [H] = \left[P^t_{4,3} \mid I_{3,3}\right]$$

5) If we receive:

$$v^t = u^t \oplus \varepsilon^t \text{ with } \varepsilon^t = [0, 0, \cdots, \underset{j^{th}\text{ position}}{1}, 0, \cdots, 0]$$

then:

$$v_{7,1}^t \times [H']_{3,7}^t = s_{3,1}'^t = [u^t \oplus \varepsilon^t] \times [H']^t = u^t \times [H']^t \oplus \varepsilon^t \times [H']^t$$
$$= \varepsilon_{7,1}^t \times [H']_{3,7}^t$$

$$\rightarrow s'^t = \varepsilon^t \times [H']^t = j^{th} \text{ row of } [H']^t$$

6) If s'^t gives the position of the error coded in a natural binary code, the form of the matrix $[H']^t$ is then:

$$[H']_{3,7}^t = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Hence:

$$[H']_{3,7} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \rightarrow [H]_{3,7} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

We get $[H]$ verifying the systematic code from $[H']$.

7) The generator matrix $[G]$ of the code is such that:

$$[H]_{3,7}^t = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ - & - & - \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} P_{4,3} \\ -- \\ I_{3,3} \end{bmatrix} \rightarrow [G]_{4,7} = \left[I_{4,4} \mid P_{4,3} \right]$$

$$= \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right]$$

8) Construction of the codeword u^t corresponding to the information word:

$$i_{4,1}^t \times [G]_{4,7} = u_{7,1}^t$$

$$[u_1 \quad u_2 \quad u_3 \quad u_4] \times [G] = [u_1 \quad u_2 \quad u_3 \quad u_4 \quad u_5 \quad u_6 \quad u_7]$$

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Here:

– $[u_1 \quad u_2 \quad u_3 \quad u_4] = [i_1 \quad i_2 \quad i_3 \quad i_4]$ is the information word;

– $[u_5 \quad u_6 \quad u_7]$ is the control word concatenated to the information word.

9) Checking of code membership or code non-membership:

$$[v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \quad v_6 \quad v_7] \times [H']^t = [s_3' \quad s_2' \quad s_1']$$

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The first word received is not a member of the code: error on the 2nd bit.

The second word received is not a member of the code: error on the 4th bit.

The 3rd word received is a member of the code: no error detected.

10) The relationship:

$$i^t \times [G] = u^t$$

makes it possible to determine the control bits as a function of the information bits:

$$[u_5 \quad u_6 \quad u_7] = f[u_1 \quad u_2 \quad u_3 \quad u_4]$$

and, from 8), we get:

$$u_5 = u_2 \oplus u_3 \oplus u_4$$

$$u_6 = u_1 \oplus u_3 \oplus u_4$$

$$u_7 = u_1 \oplus u_2 \oplus u_4$$

These same equations can be obtained from the following relationships:

$$u_{7,1}^t \times [H']_{3,7}^t = [0]_{1,3} \quad \text{or again :} \quad u_{7,1}^t \times [H]_{3,7}^t = [0]_{1,3}$$

Hamming coder

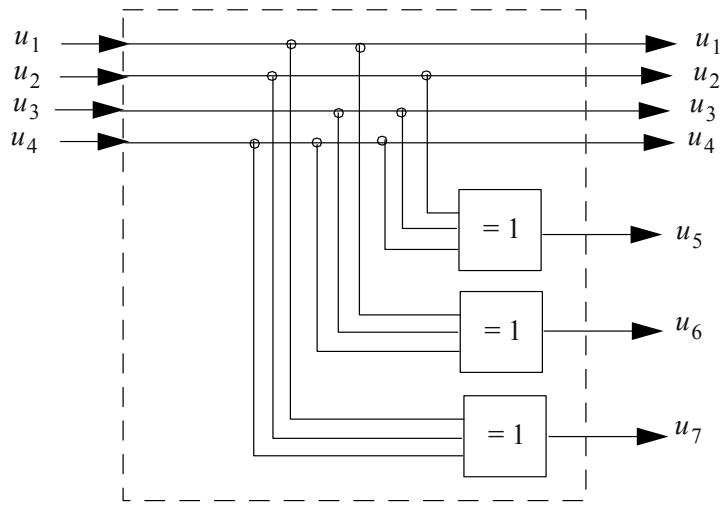


Figure 1.7. *Block diagram of Hamming coder* $C(7,4)$

Hamming decoder

The decoder is based on:

1) The calculation of the syndrome given by the relation: $s_{3,1}'^t = v_{7,1}^t \times [H']_{3,7}^t$:

$$[s_3' \quad s_2' \quad s_1'] = [v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \quad v_6 \quad v_7] \times \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

hence:

$$s_3' = v_4 \oplus v_5 \oplus v_6 \oplus v_7$$

$$s_2' = v_2 \oplus v_3 \oplus v_6 \oplus v_7$$

$$s_1' = v_1 \oplus v_3 \oplus v_5 \oplus v_7$$

2) The identification of the position of the error and its possible correction:

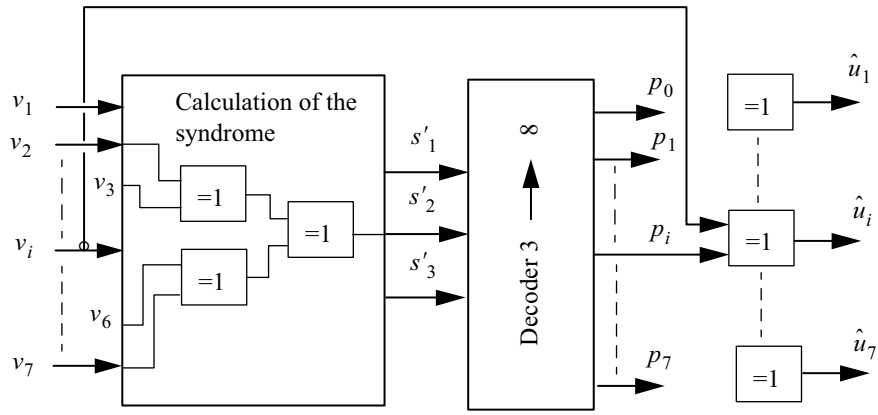


Figure 1.8. *Block diagram of Hamming decoder* $C(7,4)$

1.11. Problem 11 – Cyclic coding (1)

The problem of coding the information to be transmitted in order to protect it against transmission errors is tackled. For that, we propose to use a cyclic code C defined by its generator polynomial $g(x)$ of degree $k = 3$:

$$g(x) = x^3 + x^2 + 1$$

with $n = 7$, the length of the codes generated by $g(x)$.

1) What is the necessary and sufficient condition for the code generated by $g(x)$ to be a cyclic code?

2) Give explicitly the generator matrix $[G]$ of the code C.

3) Determine the polynomial $h(x)$, then the corresponding matrix $[H]$.

4) Determine the expressions of the control bits according to the information bits, based on:

a) the matrix $[H]$;

b) the generator polynomial $g(x)$.

Let $i(x) = x^3 + 1$ be the polynomial information (information word) to encode.

5) Determine the polynomials $c(x)$ and $u(x)$ corresponding to the control word and to the codeword, respectively.

6) Give the implementation scheme of the encoder (based on D flip-flops) providing a systematic code after n clock cycles.

7) Give the implementation scheme of the decoder associated with the coder from question 6.

8) Give the implementation scheme of the encoder based on LFSR register (linear feedback shift register) providing a systematic code after n clock cycles.

9) Give the implementation scheme of the decoder associated with the coder from question 8.

10) Does the generated cyclic code detect single, double or triple errors? Justify your answers.

11) Determine the length-percentage pairs of detectable error packets by this code.

12) Give the implementation scheme of the pseudo-random number generator based on the generator polynomial $g(x)$. Starting from the initial state $[Q]^t = [Q_0 = 0 \quad Q_1 = 0 \quad Q_2 = 1]$, give the state of the register at each clock cycle and until the register returns to its initial state.

What is the length of the cycle produced at the output of this pseudo-random number generator?

Solution of problem 11

1) The necessary and sufficient condition for $g(x)$ to generate a cyclic code is that $g(x)$ divides $(x^n + 1)$ but does not divide $(x^{n_1} + 1)$, with $n_1 < n = 7$:

$g(x)$ divides $(x^7 + 1)$

because:

$$(x^7 + 1) = (x^3 + x^2 + 1) \times (x^4 + x^3 + x^2 + 1)$$

but does not divide $(x^{n_1} + 1)$, with $n_1 < n = 7$.

2) We have: $n = m + k$, with $n = 7$ and $k = 3 \rightarrow m = 4$.

Generator matrix of the cyclic code:

$$[G]_{m,n} = [G]_{4,7} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \leftarrow g(x)$$

3) Polynomial $h(x)$ and matrix $[H]$:

$$h(x) = \frac{x^7 + 1}{g(x)} = \frac{x^7 + 1}{x^3 + x^2 + 1} = x^4 + x^3 + x^2 + 1; \quad d^\circ h(x) = m = 4$$

$$[H]_{k,n} = [H]_{3,7} = \begin{matrix} h(x) \rightarrow \\ \end{matrix} \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

4) a) Expression of the control bits from the matrix $[H]$.

We have:

$$[H]_{k,n} \times [u]_{n,1} = [0]_{k,1} \rightarrow [H]_{3,7} \times [u]_{7,1} = [0]_{3,1}$$

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} u_6 \\ u_5 \\ u_4 \\ u_3 \\ u_2 \\ u_1 \\ u_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\rightarrow \begin{cases} u_6 \oplus u_4 \oplus u_3 \oplus u_2 = 0 \\ u_5 \oplus u_3 \oplus u_2 \oplus u_1 = 0 \\ u_4 \oplus u_2 \oplus u_1 \oplus u_0 = 0 \end{cases} \rightarrow \begin{cases} u_2 = u_3 \oplus u_4 \oplus u_6 \\ u_1 = u_5 \oplus u_3 \oplus u_3 \oplus u_4 \oplus u_6 = u_4 \oplus u_5 \oplus u_6 \\ u_0 = u_4 \oplus u_2 \oplus u_1 = u_3 \oplus u_4 \oplus u_5 \end{cases}$$

4) b) Expression of the control bits from the generator polynomial $g(x)$:

$$c(x) = \text{Remainder}\left\{\frac{x^k i(x)}{g(x)}\right\} = \text{Remainder}\left\{\frac{x^3[u_6x^3 + u_5x^2 + u_4x + u_3]}{x^3 + x^2 + 1}\right\}$$

$$\begin{array}{rcl} & & x^3 + x^2 + 1 \\ \oplus & u_6x^6 + u_5x^5 + u_4x^4 + u_3x^3 & \mid - - - - - - - - - - - - - - - - - \\ & u_6x^6 + u_6x^5 + u_6x^3 & \mid u_6x^3 + (u_5 + u_6)x^2 + (u_4 + u_5 + u_6)x \\ & - - - - - - - - - - - - & \mid \quad +(u_3 + u_4 + u_5) \end{array}$$

$$\begin{array}{c} (u_5 + u_6)x^5 + u_4x^4 + (u_3 + u_6)x^3 \\ (u_5 + u_6)x^5 + (u_5 + u_6)x^4 + (u_5 + u_6)x^2 \\ - - - - - - - - - - - - - - - - - - \\ (u_4 + u_5 + u_6)x^4 + (u_3 + u_6)x^3 + (u_5 + u_6)x^2 \\ (u_4 + u_5 + u_6)x^4 + (u_4 + u_5 + u_6)x^3 + (u_4 + u_5 + u_6)x \\ - \\ (u_3 + u_4 + u_5)x^3 + (u_5 + u_6)x^2 + (u_4 + u_5 + u_6)x \\ (u_3 + u_4 + u_5)x^3 + (u_3 + u_4 + u_5)x^2 + (u_3 + u_4 + u_5) \\ - \\ \underbrace{(u_3 + u_4 + u_6)}_{u_2} x^2 + \underbrace{(u_4 + u_5 + u_6)}_{u_1} x + \underbrace{(u_3 + u_4 + u_5)}_{u_0} \end{array}$$

$$\rightarrow c(x) = u_2x^2 + u_1x + u_0$$

5) Control word and codeword generation:

$$i(x) = x^3 + 1 \rightarrow x^k i(x) = x^3(x^3 + 1) = x^6 + x^3$$

$$\rightarrow \begin{cases} u_6 = u_3 = 1 \\ u_4 = u_5 = 0 \end{cases}$$

$$\rightarrow \begin{cases} u_2 = 0 \\ u_1 = u_0 = 1 \end{cases}$$

$$\rightarrow c(x) = x + 1 \text{ and } u(x) = x^k i(x) + c(x) = x^6 + x^3 + x + 1$$

6) Design of the encoder implementation scheme:

The multiplexers (Muxs) $c1$ and $c2$ are in position 1 during $m = 4$ clock cycles. At the clock cycles $m + 1, m + 2, \cdots, n$ that is from 5 to 7, the multiplexers are in position 2.

7) Design of the decoder implementation scheme:

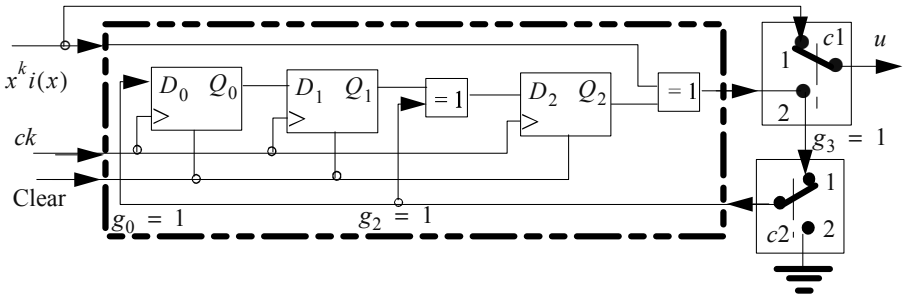


Figure 1.9. *Implementation scheme of the encoder*

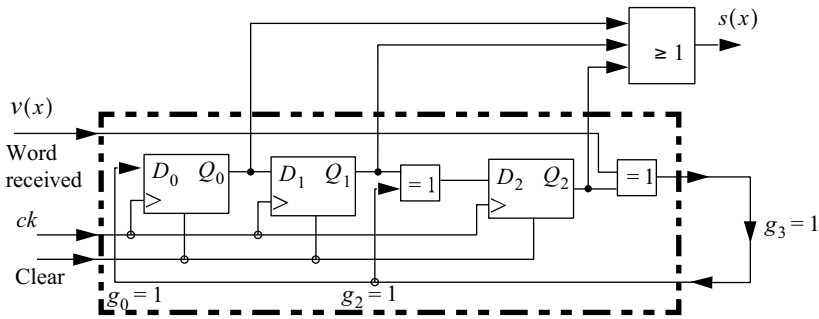


Figure 1.10. *Implementation scheme of the decoder*

After $n = 7$ clock cycles, we look at the value of the syndrome $s(x)$:

$$\rightarrow \text{if } \begin{cases} s(x) = 0 \rightarrow \text{no transmission error detected} \\ s(x) = 1 \rightarrow \text{detection of transmission error} \end{cases}$$

8) Coder based on a linear feedback shift register (LFSR).

The multiplexer (Mux) c is in position 1 during $m = 4$ clock cycles, then in position 2 for the next clock cycles $m + 1, m + 2, \cdots, n$, that is from 5 to 7.

9) Decoder based on a linear feedback shift register (LFSR).

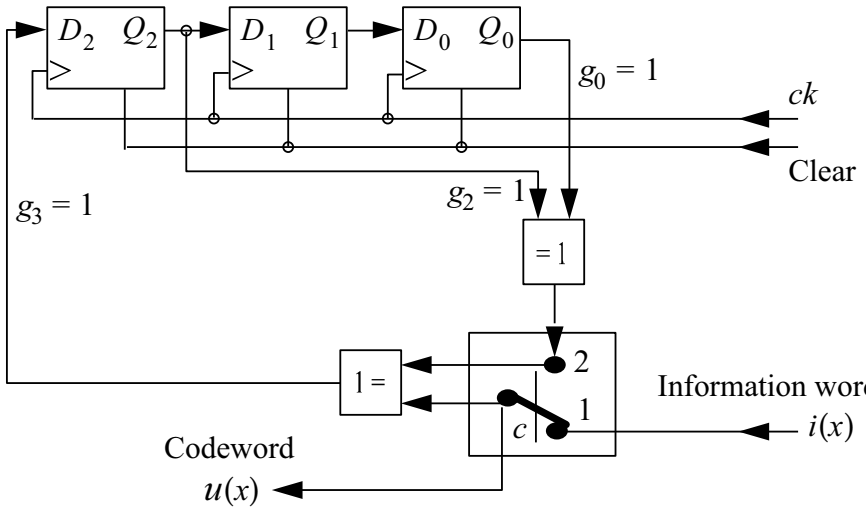


Figure 1.11. *Implementation scheme of the coder based on a linear feedback shift register*

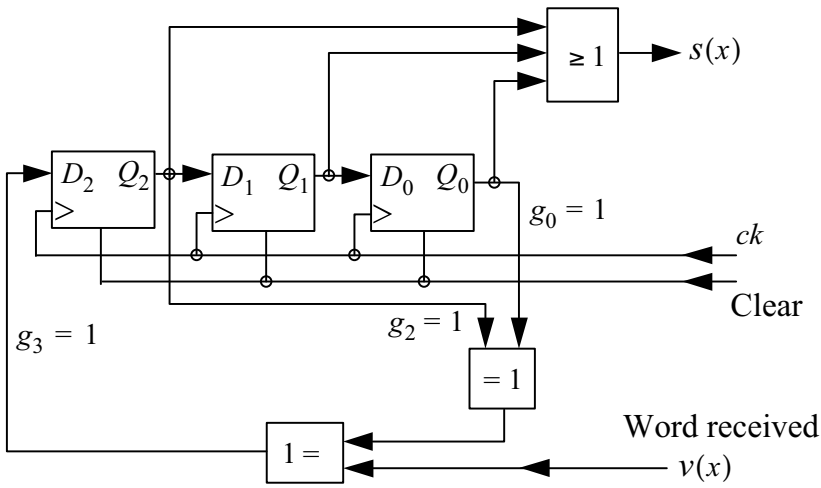


Figure 1.12. *Implementation scheme of the decoder based on a linear feedback shift register*

10) The received word is: $v(x) = u(x) + \varepsilon(x)$

The syndrome is given by:

$$s(x) = \text{Remainder}\left[\frac{v(x)}{g(x)}\right] = \text{Remainder}\left[\frac{\varepsilon(x)}{g(x)}\right]$$

Error detection is possible if $v(x)$ does not belong to the code and if $g(x)$ does not divide $\varepsilon(x)$.

– *Single errors*: in this case, $\varepsilon(x)$ is of the form $\varepsilon(x) = x^i$ which is not divisible by $g(x)$ of the form $g(x) = 1 + \cdots,$ consequently, detection of all the single errors.

– *Triple errors*: if $g(x) \neq (1 + x)p(x)$, then no detection of all the triple errors (see question 11).

– *Double errors*: in this case, $\varepsilon(x)$ is of the form $\varepsilon(x) = x^i + x^j = x^i(x^{j-i} + 1)$. Since $g(x)$ does not divide x^i, it then suffices that $g(x)$ does not divide $(x^{j-i} + 1)$ either. The generator polynomial $g(x)$ divides $x^n + 1$ but does not divide $x^{n_1} + 1$, with $n_1 < n$, then $g(x)$ is said to be of order n. If $n = 2^k - 1$, then $g(x)$ is a primitive polynomial. Here, $n = 7,\ k = 3$, and $7 = 2^3 - 1$, thus, this code is able to detect all the double errors because $(j - i) < n$.

11) A packet of errors that starts in position j and is of length l is written:

$$\varepsilon(x) = x^j + \varepsilon_{j+1}x^{j+1} + \cdots + x^{j+l-1}$$

where the first and the last coefficients are at 1 and the others can be 0 or 1:

$$\varepsilon(x) = x^j \times \left[1 + \varepsilon_{j+1}x + \cdots + x^{l-1}\right] = x^j \times \varepsilon 1(x)$$

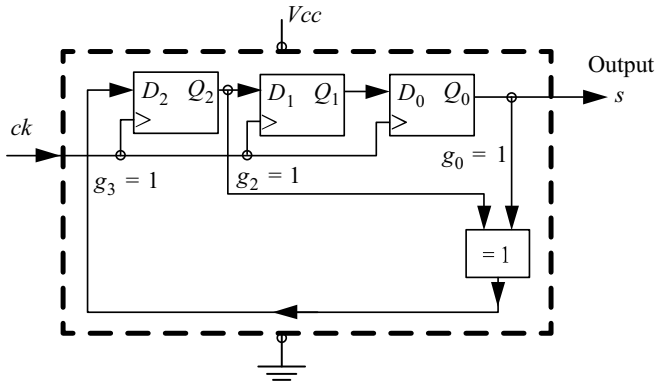
Several cases are to be considered:

– $l - 1 < k : k = 3 \rightarrow l = 3$, then detection of 100% of the error packets with $l \leq k$;

– $l - 1 = k \rightarrow l = 4$, and the proportion of detectable error packets is then: $1 - 2^{-(k-1)} = 1 - 2^{-2} = 0.75,$ i.e. 75% of the error packets;

– $l - 1 > k \rightarrow l > 4$, and the proportion of detectable error packets is then: $1 - 2^{-k} = 1 - 2^{-3} = 0.875,$ i.e. 87.5% of the error packets.

12) Pseudo-random number generator.



	Q_2	Q_1	Q_0	
N° ck	1	0	0	Initial state
1°	1	1	0	
2°	1	1	1	
3°	0	1	1	
4°	1	0	1	
5°	0	1	0	
6°	0	0	1	
7°	1	0	0	Return to initial state

Output: $s = Q_0$

Figure 1.13. *Pseudo-random number generator and register states*

The cycle length is: $l = 2^k - 1 = 2^3 - 1 = 7 \rightarrow \begin{cases} 4 \text{ bits at } 1 \\ 3 \text{ bits at } 0 \end{cases}$: a quasi-balanced sequence.

1.12. Problem 12 – Cyclic coding (2)

The problem of coding the information to be transmitted in order to protect it against transmission errors is considered. For that, we use a cyclic code C defined by its generator polynomial $g(x)$ of degree k and the polynomial $h(x)$ of degree m, orthogonal to $g(x)$ modulo $(x^n + 1)$.

We set $n = 15$ and the associated generator polynomial is:

$$g(x) = x^5 + x^4 + x^2 + 1$$

1) Does the cyclic code C detect double errors? Justify your answer.

We impose that the cyclic code be a systematic code, that will be denoted code C_1. In this case, a word to be encoded is represented by the polynomial $i(x)$, and from this the coded word represented by the polynomial $u(x)$ is obtained.

2) What is the structure of the polynomial $u(x)$: format of each of the two parts of $u(x)$?

3) From the construction mechanism of the codewords u by the code C_1, determine the implementation scheme of the coder associated with the code C_1 (using only the operators: D flip-flop; multiplexer 2 to 1; XOR).

Taking as an example the word to be coded i represented by the polynomial $i(x) = x^8 + x^6 + x^3 + x + 1$, describe the operation of the pre-multiplied encoder: internal state and values of the input and output at each clock cycle.

Deduce the polynomial code $u_1(x)$ associated with $i(x)$.

4) Determine the implementation scheme of the decoder associated with the code C_1 making it possible for the detection of errors and explain how it operates.

We no longer impose the cyclic code C to be systematic. Let C_2 be the code C such that $u(x)$ is obtained by multiplication of $i(x)$ by $g(x)$.

5) Determine the implementation scheme of the coder associated with the code C_2 (using only the operators: D flip-flop; XOR).

Taking as an example the word to be coded i from question 3, describe the operation of the coder. Deduce the polynomial code $u_2(x)$ associated with $i(x)$.

Solution of problem 12

1) We have $n = 15; k = 5$ and:

$$g(x) = x^5 + x^4 + x^2 + 1 = (x + 1)(x^4 + x + 1) = (x + 1) \times p(x)$$

$g(x)$ is not primitive, but $p(x)$ which is of degree 4, is primitive because: $n = 15 = 2^4 - 1$.

Two errors occurring in position i and j of a codeword are characterized by a polynomial error of type:

$$\varepsilon(x) = x^i + x^j = x^i(x^{j-i} + 1) \quad \text{with} \quad n > j > i$$

The polynomial $p(x)$ being primitive, thus $p(x)$ does not divide any of the polynomials of the form $(x^{n_1} + 1)$ with $n_1 < n$. Then $(j - i)$ is at most equal to $(n - 1)$. In addition, $p(x)$ does not divide x^i, hence this cyclic code detects all the double errors.

It should also be noted that the polynomial $(x + 1)$ detects all the single and triple errors.

2) Structure of the polynomial:

$$x^k i(x) = g(x) \times q(x) + c(x) \rightarrow x^k i(x) + c(x) = g(x) \times q(x) = u(x)$$

with:

– $x^k i(x)$: polynomial information cyclically shifted from k positions to the left;

– $c(x)$: polynomial control.

3) We have:

$$x^k i(x) = g(x) \times q(x) + c(x)$$

hence:

$$x^5 \times (x^8 + x^6 + x^3 + x + 1) = (x^5 + x^4 + x^2 + 1) \times q(x) + c(x)$$

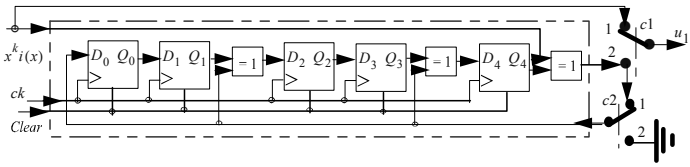
$$x^{13} + x^{11} + x^8 + x^6 + x^5 = (x^5 + x^4 + x^2 + 1) \times (x^8 + x^7 + x^5 + x + 1) + x^4 + x^3 + x^2 + x + 1$$

So finally:

$$u_1(x) = x^k i(x) + c(x)$$

$$u_1(x) = x^{13} + x^{11} + x^8 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$$

Diagram of implementation of the coder associated with code C_1 (block diagram of Table 1.15) and description of its operation.



c1c2 1 1	ck	$i(x)$	D_0	Q_0 D_1	Q_1	D_2	Q_2 D_3	Q_3	D_4	Q_4	u_1
		1 x^8	1	**0**	**0**	1	**0**	**0**	1	**0**	1
	1°			1	0		1	0		1	
		0	1	1		1	1		1		0
	2°			1	1		1	1		1	
		1	0	1		1	1		1		1
	3°			0	1		1	1		1	
		0	1	0		0	1		0		0
	4°			1	0		0	1		0	
		0	0	1		0	0		1		0
	5°			0	1		0	0		1	
		1	0	0		1	0		0		1
	6°			0	0		1	0		0	
		0	0	0		0	1		0		0
	7°			0	0		0	1		0	
		1	1	0		1	0		0		1
	8°			1	0		1	0		0	
		1	1	1		1	1		1		1
	9°			1	1		1	1		1	
2 2		0	0	1		1	1		1		1
	10°			0	1		1	1		1	
		0	0	0		1	1		1		1
	11°			0	0		1	1		1	
		0	0	0		0	1		1		1
	12°			0	0		0	1		1	
		0	0	0		0	0		1		1
	13°			0	0		0	0		1	
		0	0	0		0	0		0		1
	14°			0	0		0	0		0	

Table 1.15. *Description of the operations of the premultiplied coder. For a color version of this table, see www.iste.co.uk/assad/digital2.zip*

4) The structure of the decoder for error detection is given in Figure 1.14.

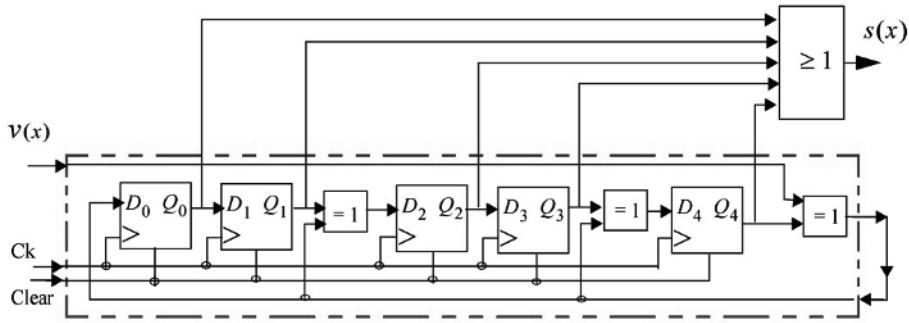


Figure 1.14. *Structure of the decoder for the detection of errors. For a color version of this figure, see www.iste.co.uk/assad/digital2.zip*

The detection process is as follows:

– initialization: reset the register by performing the action Clear;

– during n clock cycles, the received word $v(x)$ enters the divisor. The remainder of the division $x^k s(x)$ is stored in the register at the n^{th} clock cycle, the output of the OR gate will then indicate whether there is an error or not.

5) We have:

$$u_2(x) = i(x) \times g(x) \mod (x^n + 1)$$

$$u_2(x) = \sum_{s=0}^{m-1} i_s x^s \sum_{j=0}^{k} g_j x^j = \sum_{s=0}^{m-1} \sum_{j=0}^{k} i_s \times g_j \times x^{s+j}$$

Let's set: $l = s + j$:

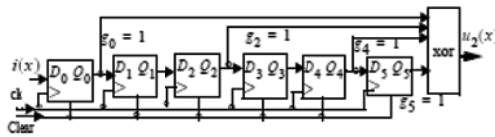
$$u_2(x) = \sum_{l=0}^{m+k-1} \left[\sum_{s=0}^{m-1} i_s \times g_{l-s} \right] x^l \text{ with } (l-s) \in [0, \cdots, k]$$

again:

$$u_2(x) = i_0 g_0 + (i_0 g_1 + i_1 g_0)x + (i_0 g_2 + i_1 g_1 + i_2 g_0)x^2 + \cdots$$
$$+(i_{m-2} g_k + i_{m-1} g_{k-1})x^{m+k-2} + i_{m-1} g_k x^{m+k-1}$$

A hardware implementation of this relation defines the coder associated with the code C_2 (see the block diagram of Table 1.16). The information word is entered in a shift register, least significant bit first, and the bits corresponding to the terms

present in the register are added (modulo 2). The bits of the product come out, least significant bit first.



ck	$i(x)$	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	$u_2(x)$
	1	0	0	0	0	0	0	
1°		1	0	0	0	0	0	$1\,x^0$
	1							
2°		1	1	0	0	0	0	1
	0							
3°		0	1	1	0	0	0	1
	1							
4°		1	0	1	1	0	0	0
	0							
5°		0	1	0	1	1	0	1
	0							
6°		0	0	1	0	1	1	1
	1							
7°		1	0	0	1	0	1	0
	0							
8°		0	1	0	0	1	0	1
	1							
9°		1	0	1	0	0	1	1
	0							
10°		0	1	0	1	0	0	0
	0							
11°		0	0	1	0	1	0	0
	0							
12°		0	0	0	1	0	1	1
	0							
13°		0	0	0	0	1	0	1
	0							
14°		0	0	0	0	0	1	$1\,x^{13}$
	0							
15°		0	0	0	0	0	0	

Table 1.16. *Description of the operations of the encoder* C_2

Indeed:

$$u_2(x) = i(x) \times g(x) = (x^8 + x^6 + x^3 + x + 1) \times (x^5 + x^4 + x^2 + 1)$$
$$= x^{13} + x^{12} + x^{11} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

1.13. Problem 13 – Cyclic coding and Hamming coding (1)

We consider a linear block code C of parameter $n = 7$ and of primitive generator polynomial: $g(x) = x^3 + x^2 + 1$.

1) Show that this code is cyclic. Deduce the second primitive generator polynomial $g_1(x)$.

2) Determine a matrix $[G]$ generating this code. Deduce the generator matrix $[G_s]$ from the systematic version of the code in question.

3) Determine the codeword $u(x)$ in systematic form which is associated with the information word: $i(x) = x^3 + 1$.

4) Design the premultiplied coder making it possible to generate the codeword $u(x)$ from the information word: $i(x) = x^3 + 1$.

5) Give the control matrix $[H]$ of the dual code to the code C.

6) Find, from the relation linking the control matrix $[H]$ and the codeword u, the control bits as a function of the information bits of question 3.

7) Make your comments about the code C and its dual.

Solution of problem 13

1) The code is cyclic if $g(x)$ divides $(x^n + 1)$ but does not divide $(x^{n_1} + 1)$ with $n_1 < n$.

Here $n = 7$ and:

$$(x^7 + 1) = (x^3 + x^2 + 1) \times (x^4 + x^3 + x^2 + 1)$$

$$(x^7 + 1) = (x^3 + x^2 + 1) \times (x^3 + x + 1) \times (x + 1)$$

So, $g(x)$ divides $(x^7 + 1)$, and the code C is a cyclic code.

The second primitive generator polynomial $g_1(x)$ is:

$$g_1(x) = (x^3 + x + 1)$$

2) The generator matrix $[G]$ of the code $C(7,4)$ is given from the generator polynomial $g(x)$ as follows:

$$[G]_{4,7} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \begin{matrix} x^3 g(x) \\ x^2 g(x) \\ x g(x) \\ g(x) \end{matrix}$$

To get a systematic code, the generating matrix $[G_s]$ must have the form $[G_s] = [I_{4,4} \mid P_{4,3}]$ obtained from the arithmetic operations on the rows of the matrix $[G]_{4,7}$. Indeed, from the form of the matrix $[G]$ we find that:

– the row 1 of the matrix $[G_s]$ is obtained by the sum of the rows: 1 + 2 + 3 of the matrix $[G]$;

– the row 2 of the matrix $[G_s]$ is obtained by the sum of the rows: 2 + 3 + 4 of the matrix $[G]$;

– the row 3 of the matrix $[G_s]$ is obtained by the sum of the rows: 3 + 4 of the matrix $[G]$;

– the row 4 of the matrix $[G_s]$ is identical to the row 4 of the matrix $[G]$, hence:

$$[G_s] = \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{array}\right]$$

3) We have:

$$x^k i(x) = g(x) \times q(x) + c(x) \rightarrow x^k i(x) + c(x) = g(x) \times q(x) = u(x)$$

with:

$$i(x) = x^3 + 1\,; k = 3 \;\rightarrow x^k i(x) = x^3 \times (x^3 + 1\,) = x^6 + x^3$$

hence:

$$\begin{array}{rl} x^6 + x^3 & \mid\; x^3 + x^2 + 1 \leftarrow g(x) \\ \oplus & \mid\; - - - - - - - - - - - \\ x^6 + x^5 + x^3 & \mid\; x^3 + x^2 + x + 1 \leftarrow q(x) \\ - - - - - - - & \mid \end{array}$$

$$
\begin{array}{c}
x^5 \\
x^5 + x^4 + x^2 \\
\hline
x^4 + x^2 \\
x^4 + x^3 + x \\
\hline
x^3 + x^2 + x \\
x^3 + x^2 + 1 \\
\hline
x + 1 \leftarrow c(x)
\end{array}
$$

$$\rightarrow u(x) = x^6 + x^3 + x + 1$$

$$u = [1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1]$$

4) Construction of the pre-multiplied coder.

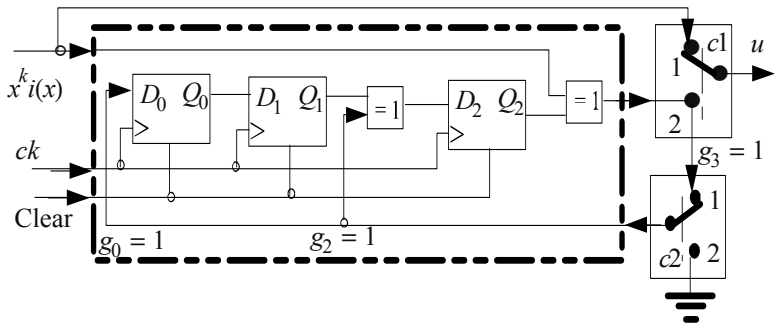


Figure 1.15. *Implementation scheme of the pre-multiplied coder*

5) Control matrix $[H]$ of the dual code to the code C.

It is such that we have:

$$[G_s] \times [H]^t = [0]\,; \quad [G_s] = \left[I_{4,4} \mid P_{4,3}\right] \rightarrow [H] = \left[P_{4,3}^t \mid I_{3,3}\right]$$

$$\rightarrow [H]_{3,7} = \begin{bmatrix} 1 & 0 & 1 & 1 & | & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & | & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & | & 0 & 0 & 1 \end{bmatrix}$$

6) Control bits according to the information bits of question 3:

$$[H]_{3,7} \times u_{7,1} = [0]_{3,1}$$

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} u_6 \\ u_5 \\ u_4 \\ u_3 \\ u_2 \\ u_1 \\ u_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\rightarrow \begin{cases} u_6 + u_4 + u_3 + u_2 = 0 \\ u_6 + u_5 + u_4 + u_1 = 0 \\ u_5 + u_4 + u_3 + u_0 = 0 \end{cases} \rightarrow \begin{cases} u_2 = u_3 + u_4 + u_6 \\ u_1 = u_4 + u_5 + u_6 \\ u_0 = u_3 + u_4 + u_5 \end{cases}$$

Thus, we have:

$$u^t = \left[\underbrace{u_6\, u_5\, u_4\, u_3}_{\text{information bits}} \quad \underbrace{u_2\, u_1\, u_0}_{\text{control bits}}\right] = [1\,0\,0\,1\,0\,1\,1]$$

7) The dual code of a maximum length cyclic code is the Hamming code.

1.14. Problem 14 – Cyclic coding and Hamming coding (2)

We consider a linear block code defined by its following generator matrix:

$$[G]_{m,n} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

1) Give the expression of the generator polynomial $g(x)$ associated with $[G]_{m,n}$.

2) Is the code generated by $g(x)$ cyclic? Justify your answer.

It is required that the cyclic code generated by $g(x)$ is systematic.

3) Determine the polynomial (codeword) $u(x)$ from the polynomial (information word): $i(x) = x^2 + 1$.

4) Give the implementation scheme of the pre-multiplied encoder making it possible to generate the codeword $u(x)$ from the information word $i(x) = x^2 + 1$ and describe its operation: internal state and input and output values for three clock cycles.

5) Does the generated code detect odd numbers of errors and double errors? Justify your answer.

6) Determine the proportion of error packets of length $l > 5$, detectable by the generated code.

7) Give explicitly the generating matrix $[G_s]$, from the matrix $[G]$ given above, which allows the generation of a systematic code C.

8) Determine explicitly the form of the control matrix $[H]$ which enables the generation of a code D dual to the code C.

9) Find, from the relation between the control matrix $[H]$ and the codeword u, the control bits as a function of the information bits.

10) Give the implementation scheme of the pseudo-random number generator (PRNG) based on $g(x)$.

Solution of problem 14

1) The last row of $[G]_{m,n} = [G]_{3,7}$ is the lower-level codeword that represents the generator polynomial $g(x)$:

$$\rightarrow g(x) = x^4 + x^3 + x^2 + 1 \;\; \rightarrow k = 4$$

2) From $[G]_{3,7} \rightarrow m = 3$ and $n = m + k = 3 + 4 = 7$. The generated code is cyclic if $g(x)$ divides $(x^n + 1)$ but does not divide $(x^{n_1} + 1)$, with $n_1 < n = 7$:

$$\begin{array}{rl} x^7 + 1 & \big| \; x^4 + x^3 + x^2 + 1 \leftarrow g(x) \\ \oplus & \big| \; \overline{\quad\quad\quad\quad\quad\quad\quad} \\ x^7 + x^6 + x^5 + x^3 & \big| \; x^3 + x^2 + 1 \leftarrow q(x) \\ \overline{\quad\quad\quad\quad\quad\quad} & \big| \end{array}$$

$$x^6 + x^5 + x^3 + 1$$

$$x^6 + x^5 + x^4 + x^2$$

$$\overline{\quad\quad\quad\quad\quad\quad}$$

$$x^4 + x^3 + x^2 + 1$$

$$x^4 + x^3 + x^2 + 1$$

$$\overline{\quad\quad\quad\quad\quad\quad\quad}$$

$$0 \quad 0 \quad 0 \quad 0$$

$$\rightarrow (x^7+1) = \underbrace{(x^4+x^3+x^2+1)}_{g(x)} \times \underbrace{(x^3+x^2+1)}_{q(x)}$$

Therefore, since $g(x)$ divides (x^7+1), but does not divide (x^6+1), (x^5+1), or (x^4+1), then the code generated by $g(x)$ is a cyclic one.

3) Determination of the polynomial $u(x)$ associated to the polynomial: $i(x) = x^2+1$.

We have:

$$x^k i(x) = g(x) \times q(x) + c(x) \rightarrow x^k i(x) + c(x) = g(x) \times q(x) = u(x)$$

$$x^k i(x) = x^4 \times (x^2+1) = x^6 + x^4$$

$$\begin{array}{rl} x^6+x^4 & \vert\ x^4+x^3+x^2+1 \leftarrow g(x) \\ \oplus & \vert\ \overline{\qquad\qquad\qquad} \\ x^6+x^5+x^4+x^2 & \vert\ x^2+x+1 \leftarrow q(x) \\ \overline{\qquad\qquad} & \vert \end{array}$$

$$x^5 + x^2$$

$$x^5 + x^4 + x^3 + x$$

$$\overline{\qquad\qquad\qquad}$$

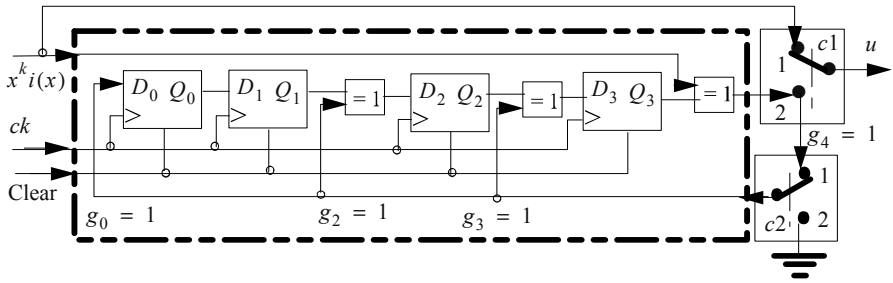
$$x^4 + x^3 + x^2 + x$$

$$x^4 + x^3 + x^2 + 1$$

$$\overline{\qquad\qquad\qquad}$$

$$c(x) = x+1 \ \rightarrow u(x) = x^k i(x) + c(x) = x^6 + x^4 + x + 1$$

4) Block diagram of the pre-multiplied encoder generating the codeword $u(x)$ from the information word $i(x) = x^2+1$ and description of its operation (see the block diagram in Table 1.17).



c1c2	ck	$x^k i(x)$	D_0	Q_0 D_1	Q_1	D_2	Q_2	D_3	Q_3	u_1
1 1		1 x^6	1	**0**	**0**	1	**0**	1	**0**	1
	1°			1	0		1		1	
		0 x^5	1	1		1		0		0
	2°			1	1		1		0	
		1 x^4	1	1		0		0		1
	3°			1	1		0		0	

Table 1.17. *Block diagram of the premultiplied encoder and encoder operation: internal state and input and output values for three clock cycles*

5) a) Detection of an odd number of errors.

If $g(x)$ can be set in the form $g(x) = (x+1)p(x)$, then $g(x)$ detects odd number of errors with $(x+1)$ (see Volume 1, Chapter 4).

$$
\begin{array}{rl}
 & x^4 + x^3 + x^2 + 1 \\
\oplus & \\
 & \underline{x^4 + x^3} \qquad\qquad x+1 \\
 & x^2 + 1 \qquad\quad | - - - - - - - - - - - \\
 & x^2 + x \qquad\quad | \; x^3 + x + 1 \leftarrow q(x) \\
 & \underline{x + 1} \qquad\quad | \\
 & x + 1 \\
 & \underline{0 \quad 0}
\end{array}
$$

$$\rightarrow \; g(x) = (x+1) \times (x^3 + x + 1) = (x+1) \times p(x)$$

So, detection of an odd number of errors.

5) b) Detection of double errors.

The generator polynomial $g(x)$ contains the polynomial $p(x) = x^3 + x + 1$ that is primitive, because $2^3 - 1 = 7 = n$, so it makes it possible $p(x)$ to detect all the double errors.

6) Proportion of detectable error packets of length $l > 5$.

We have $k = 4$, so the proportion of detectable error packets of length $l > k + 1 \rightarrow l > 5$ is:

$$1 - 2^{-k} = 1 - 2^{-4} = 93.75\ \%$$

7) The matrix $[G_s]$ is taken from the matrix $[G]_{3,7}$ by shifting the positions of some columns verifying the expected form of $[G_s]_{m,n} = [G_s]_{3,7} = \left[I_{3,3} \mid P_{4,3}\right]$:

$$[G]_{3,7} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \rightarrow [G_s]_{3,7}$$

$$= \left[\begin{array}{ccc|cccc} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{array}\right]$$

8) Form of the control matrix $[H]$:

$$[G_s]_{m,n} \times [H]_{k,n}^t = [0]_{m,k}$$

$$[G_s]_{m,n} = \left[I_{m,m} \mid P_{m,k}\right] \rightarrow [H]_{k,n} = \left[P_{m,k}^t \mid I_{k,k}\right]$$

$$[G_s]_{3,7} = \left[I_{3,3} \mid P_{3,4}\right] \rightarrow [H]_{4,7} = \left[P_{3,4}^t \mid I_{4,4}\right]$$

$$\rightarrow [H]_{4,7} = \left[\begin{array}{ccc|cccc} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right]$$

9) Control bits as a function of information bits?

We have:

$$[H]_{4,7} \times [u]_{7,1} = [0]_{4,1}$$

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

with:

u_1, u_2, u_3: information bits

u_4, u_5, u_6, u_7: control bits

$$\rightarrow \begin{cases} u_1 \oplus u_2 \oplus u_4 = 0 \\ u_1 \oplus u_2 \oplus u_3 \oplus u_5 = 0 \\ u_2 \oplus u_3 \oplus u_6 = 0 \\ u_1 \oplus u_3 \oplus u_7 = 0 \end{cases} \rightarrow \begin{cases} u_4 = u_1 \oplus u_2 \\ u_5 = u_1 \oplus u_2 \oplus u_3 \\ u_6 = u_2 \oplus u_3 \\ u_7 = u_1 \oplus u_3 \end{cases}$$

10) Implementation scheme of the pseudo-random number generator (PRNG) based on:

$$g(x) = x^4 + x^3 + x^2 + 1$$

The implementation scheme of the pseudo-random generator (PRNG) based on the polynomial $g(x)$ is given in Figure 1.16.

NOTE.– At start, the initial state $[Q_3, Q_2, Q_1, Q_0]$ of the register must be different from zero.

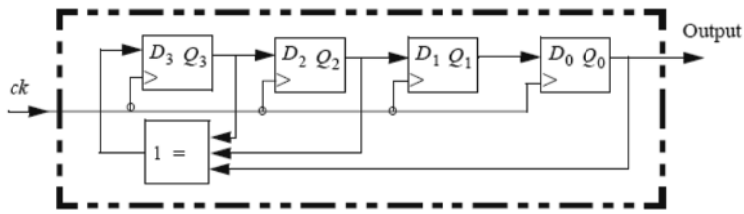


Figure 1.16. *Implementation scheme of the pseudo-random number generator (PRNG)*

1.15. Problem 15 – Cyclic code, M-sequences, and Gold sequences

We consider the problem of coding the information to be transmitted so as to protect it against transmission errors. For this purpose, a cyclic code C defined by its following generator polynomial: $g(x) = x^5 + x^2 + 1$, and $n = 31$ is used.

1) What is the necessary and sufficient condition for the proposed polynomial $g(x)$ to be primitive and generate a cyclic code?

It is desired to produce a systematic code C.

2) Give the expression of the codeword represented by the polynomial $u(x)$ corresponding to the information word represented by the polynomial: $i(x) = x^7 + x^4 + x + 1$.

3) Give the implementation scheme of the encoder based on a division circuit pre-multiplied by x^k, where k is the degree of the generator $g(x)$. Describe how it works.

4) Give the implementation scheme of the decoder associated with the code C allowing the detection of errors and explain how it works.

5) Does the generated cyclic code detect single, double or triple errors? Justify your answer in each case.

6) Determine the length-percentage pairs of error packets detectable by this code.

7) Give the wiring diagram of the pseudo-random number generator of maximum length (M-sequences), based on the primitive polynomial $g(x)$ defined above.

8) Give the expression of the generator polynomial $g_{rec}(x)$ reciprocal of the generator polynomial $g(x)$. What is the essential characteristic of the M-sequence generated by $g_{rec}(x)$ compared to that generated by $g(x)$?

9) Give the number of M-sequences generated by $g(x)$ and the ratio between the maximum of cross-correlation and that of the autocorrelation.

10) Show that the generator $g_1(x) = x^5 + x^4 + x^2 + x + 1$ associated with $g(x)$ forms a preferred pair.

11) Give the wiring diagram of the Gold generator based on $g(x)$ and $g_1(x)$, to generate all the Gold sequences.

12) Give the number of Gold sequences generated and the ratio between the maximum of cross-correlation and that of the autocorrelation.

Solution of problem 15

1) The necessary and sufficient condition that $g(x)$ is primitive is that:

$$2^k - 1 = n = 2^5 - 1$$

so $g(x)$ is primitive.

The polynomial $g(x)$ is generating a cyclic code, if it divides $x^{31} + 1$ but does not divide $(x^{n_1} + 1)$, with $n_1 < 31$.

The generator $g(x)$ divides $x^{31} - 1$, because, after division we get a null remainder:

$$x^{31} + 1 = (x^5 + x^2 + 1)$$
$$\times (x^{26} + x^{23} + x^{21} + x^{20} + x^{17} + x^{16} + x^{15} + x^{14} + x^{13}$$
$$+x^9 + x^8 + x^6 + x^5 + x^4 + x^2 + 1)$$

2) Expression of the codeword represented by the polynomial $u(x)$ corresponding to the information word represented by the polynomial $i(x)$:

We have:

$$x^k i(x) = g(x) \times q(x) + c(x) \rightarrow x^k i(x) + c(x) = g(x) \times q(x) = u(x)$$

or again:

$$x^k i(x) = x^5 \times (x^7 + x^4 + x + 1) = x^{12} + x^9 + x^6 + x^5$$

$$\begin{array}{l|l} x^{12} + x^9 + x^6 + x^5 & x^5 + x^2 + 1 \leftarrow g(x) \\ \oplus \quad\quad x^{12} + x^9 + x^7 & \hline x^7 + x^2 + x + 1 \leftarrow q(x) \\ \hline x^7 + x^6 + x^5 & \\ x^7 + x^4 + x^2 & \\ \hline x^6 + x^5 + x^4 + x^2 & \\ x^6 + x^3 + x & \end{array}$$

$$- - - - - - - -$$

$$x^5 + x^4 + x^3 + x^2 + x$$

$$x^5 + x^2 + 1$$

$$- - - - - - - - - -$$

$$c(x) = x^4 + x^3 + x + 1$$

$$\rightarrow u(x) = \underbrace{x^{12} + x^9 + x^6 + x^5}_{x^k i(x)} + \underbrace{x^4 + x^3 + x + 1}_{c(x)}$$

3) Implementation scheme of the coder based on a division circuit premultiplied by x^k.

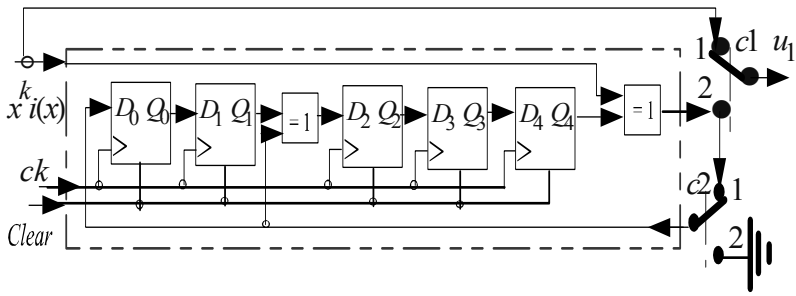


Figure 1.17. *Implementation scheme of the coder. For a color version of this figure, see www.iste.co.uk/assad/digital2.zip*

The operation of the encoder is as follows:

– resetting the D flip-flops;

– during $m = 8$ clock cycles, the multiplexers (Muxs) 1 and 2 are in position 1. The information bits are applied simultaneously to the divider and to the output. The k control bits ($k = 5$) are in the k flip-flops of the register;

– during k clock cycles, multiplexers (Muxs) 1 and 2 are in position 2; zeros enter the register and the control bits go out. The encoder uses $(m + k) = (8 + 5 = 13)$ clock cycles and the transmission channel is used throughout the operation. At the 13th clock cycle, the register flip-flops are zero and the encoder is ready to receive another information word to code. The encoder has a good efficiency.

4) Implementation scheme of the decoder associated with the code C.

The received word $v(x)$ is written:

$$v(x) = u(x) + \varepsilon(x)$$

with $\varepsilon(x)$ as a possible error word.

The syndrome is defined by:

$$s(x) = \text{Remainder}\left[\frac{v(x)}{g(x)}\right] = \text{Remainder}\left[\frac{u(x)}{g(x)}\right] + \text{Remainder}\left[\frac{\varepsilon(x)}{g(x)}\right]$$
$$= \text{Remainder}\left[\frac{\varepsilon(x)}{g(x)}\right]$$

So: if $\varepsilon(x)$ is non null, and if $v(x) \notin C(n,m)$, then $s(x) \neq 0$, hence the decoder implementation scheme.

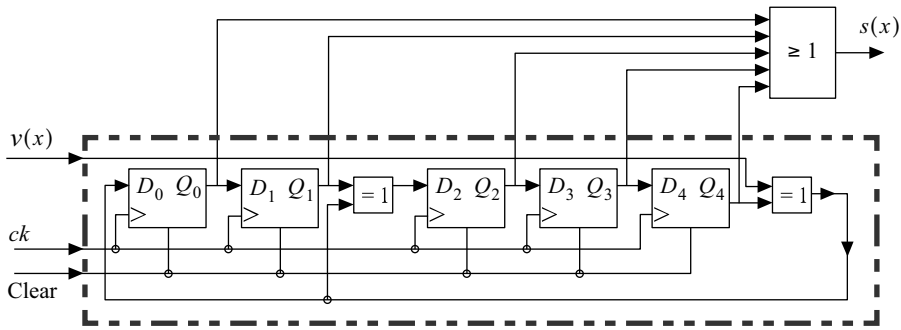


Figure 1.18. *Implementation scheme of the decoder. For a color version of this figure, see www.iste.co.uk/assad/digital2.zip*

The received word $v(x)$ is divided by $g(x)$ during $n = m + k = 8 + 5 = 13$ clock cycles. Then, the contents of the register are verified by a simple OR logic gate. If the content of the register is zero, then the received word is decided to be correct. Otherwise (the content of the register is not zero), the received word is decided to be erroneous.

5) Cyclic code capability to detect single, double or triple errors?

We know that:

$$s(x) = \text{Remainder}\left[\frac{\varepsilon(x)}{g(x)}\right]$$

Thus, error detection is possible if $g(x)$ does not divide $\varepsilon(x)$.

– *Single errors*: in this case an error in position i, represented by $\varepsilon(x) = x^i$ is not divisible by $g(x) = 1 + \cdots$, thus detection of all the single errors.

– *Triple errors*: in this case, $\varepsilon(x) = x^i + x^j + x^l$, and as $g(x) \neq (1 + x)p(x)$ then in principle, no detection of triple errors (see Volume 1, Chapter 4).

– *Double errors*: in this case, $\varepsilon(x)$ is of the form $\varepsilon(x) = x^i + x^j = x^i(x^{j-i} + 1)$ with $i < j < n$. Since $g(x)$ does not divide x^i, it suffices then that $g(x)$ does not divide either $(x^{j-i} + 1)$. The generator $g(x)$ divides $x^n + 1$ but does not divide $x^{n_1} + 1$, with $n_1 < n$, so $g(x)$ is said to be of order n. The primitive polynomials are irreducible. They detect all double errors because $(j - i) < n$.

6) Determination of the length-percentage pairs of detectable error packets.

An error packet that starts in position j and has a length l is written:

$$\varepsilon(x) = x^j + \varepsilon_{j+1}x^{j+1} + \cdots + x^{j+l-1}$$

where the first and the last coefficients of $\varepsilon(x)$ are at 1, the other coefficients can be at 1 or 0:

$$\varepsilon(x) = x^j \times \left[1 + \varepsilon_{j+1}x + \cdots + x^{l-1}\right] = x^j \times \varepsilon 1(x)$$

Three cases are encountered:

– $l - 1 < k$ $(k = 5)$ $\rightarrow l = 5$, hence detection at 100% of all the error packets of length $l \leq k$;

– $l - 1 = k \rightarrow l = k + 1 = 6$, the proportion of error packets detectable is then: $1 - 2^{-(k-1)} = 1 - 2^{-4} = 0.9375,$ i.e. 93.75% of the packets;

– $l - 1 > k \rightarrow l > 6$, the proportion of error packets detectable is then: $1 - 2^{-k} = 1 - 2^{-5} = 0.9687,$ i.e. 96.87%.

7) The implementation scheme of the pseudo-random number generator (PRNG) based on $g(x)$ is given in Figure 1.19.

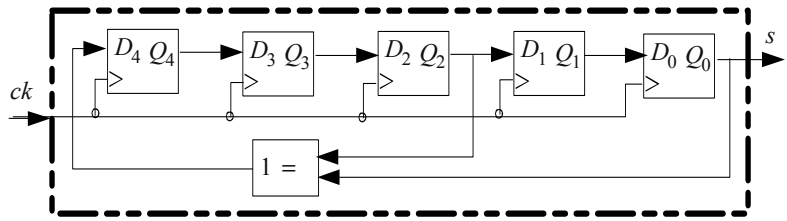


Figure 1.19. *Implementation scheme of the pseudo-random number generator based on* $g(x)$

NOTE.– At the start, the initial state of the D flip-flops $[Q_4, Q_3, Q_2, Q_1, Q_0]$ of the register should be different of zero.

8) Expression of the generator polynomial $g_{rec}(x)$ reciprocal of the generator $g(x)$.

We have:

$$g_{rec}(x) = x^k \times g(1/x) = x^5 \times (x^{-5} + x^{-2} + 1) = x^5 + x^3 + 1$$

The M-sequence generated by $g_{rec}(x)$ corresponds to the one generated by $g(x)$ but in a reverse sense.

9) Number of M-sequences generated by $g(x)$.

$k = 5$, so the number of M-sequences generated by $g(x)$ is 6 (see volume 1, chapter 4). The ratio $R_{sq_{Max}}/R_{ss}(0) = 0.35$ (see Volume 1, Chapter 4).

10) Does the generator polynomial $g_1(x) = x^5 + x^4 + x^2 + x + 1$, form with the polynomial $g(x)$ as a preferred pair?

Let α be a root of: $g(x) = x^5 + x^2 + 1$.

The polynomial $g_1(x) = x^5 + x^4 + x^2 + x + 1$, forms a preferred pair with $g(x)$ because:

$$\text{if } \begin{cases} 1)\ k \text{ is odd, since } k = 5, \text{conditions in 1) are satisfied} \\ 2)\ \ g_1(x) \text{ is such that } \alpha^{2\left[\frac{k-1}{2}\right]+1} = \alpha^5 \text{ is a root of } g_1(x) \end{cases}$$

It means that $g(\alpha)$ divides $g_1(\alpha^5)$. This condition is also satisfied, because:

$$g_1(\alpha^5) = \alpha^{25} + \alpha^{20} + \alpha^{10} + \alpha^5 + 1$$

$$g_1(\alpha^5) = g(\alpha) \times (\alpha^{20} + \alpha^{17} + \alpha^{14} + \alpha^{12} + \alpha^{11} + \alpha^8 + \alpha^7 + \alpha^6 + \alpha^4 + \alpha^2 + 1)$$

with: $g(\alpha) = (\alpha^5 + \alpha^2 + 1)$

11) Implementation scheme of the Gold generator based on $g(x)$ and $g_1(x)$.

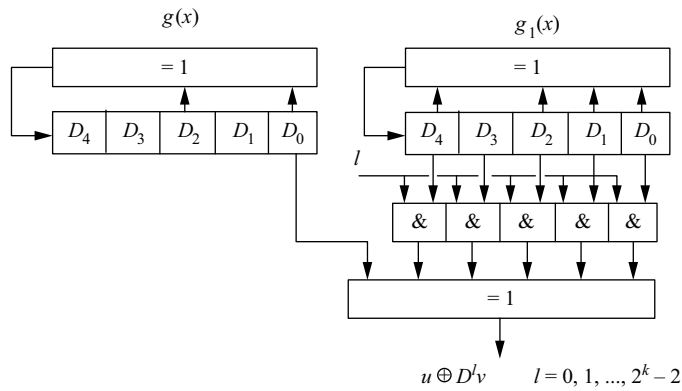


Figure 1.20. *Implementation scheme of the Gold generator*

12) The number of Gold sequences is $G(u, v) = \{u, v, u \oplus D^l v\}$, a set of $n + 2$ sequences. The ratio is $I(k)/R_{ss}(0) = 0.29$ (see Volume 1, Chapter 4).