

Assignment -02

Shahru Hameed S

18CS46

1KNI8CS097

4th Sem, CSE 'A' Sec

- ① Explain the mechanisms for modulating digital data into analog signal (data).

→ (a) Amplitude Shift Keying (ASK)

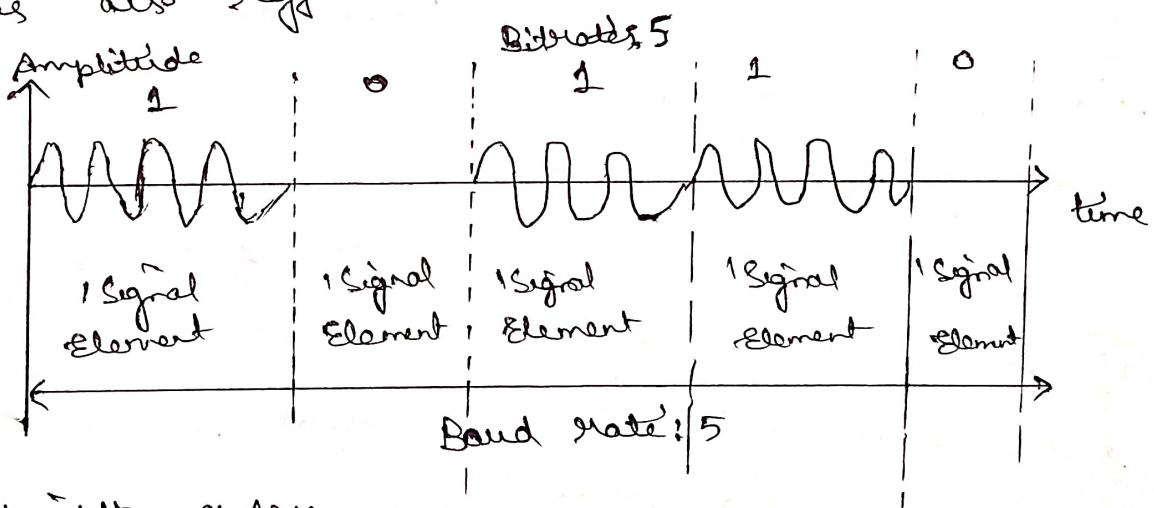
- * The amplitude of the carrier signal is varied to represent different signal elements.

- * Both frequency and phase remain constant for all signal elements.

→ Binary ASK (BASK)

- * It is implemented using only 2 levels.

- * This is also referred to asOOK (On-off keying).



⇒ Bandwidth of ASK

$$B = (1/d) \times S$$

$B = \text{Bandwidth}$, $S = \text{Signal rate}$ $d = 0 < d < 1$
 This d depends on modulation and filtering process

→ (b) frequency shift keying (FSK)

- * The frequency of the carrier signal is varied to represent different signal elements.

* The frequency of modulated signal is constant for the duration of one signal element, but varies for the next signal element if the data element changes.

\Rightarrow Binary FSK

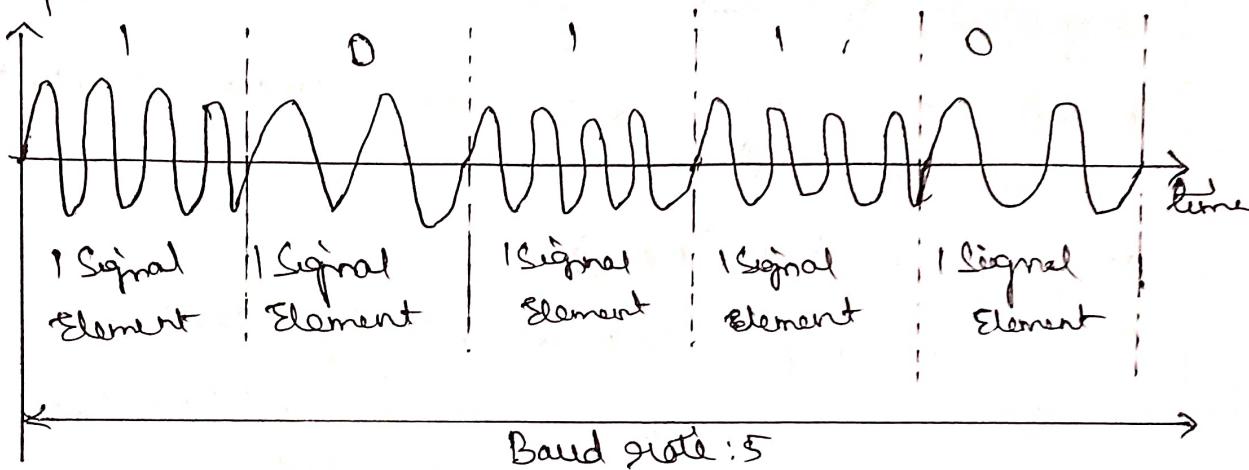
* This uses 2 carrier frequencies: f_1 and f_2 .

f_1 is called for data-element = 1

f_2 is called for data-element = 0

Amplitude

Bitrate: 5



\Rightarrow Bandwidth for BFSK

$$B = \pi d \times S + 2 \Delta f$$

where Δf is the difference b/w f_1 and f_2

(c) Phase Shift Keying:

* The phase of the carrier signal is varied to represent different signal elements.

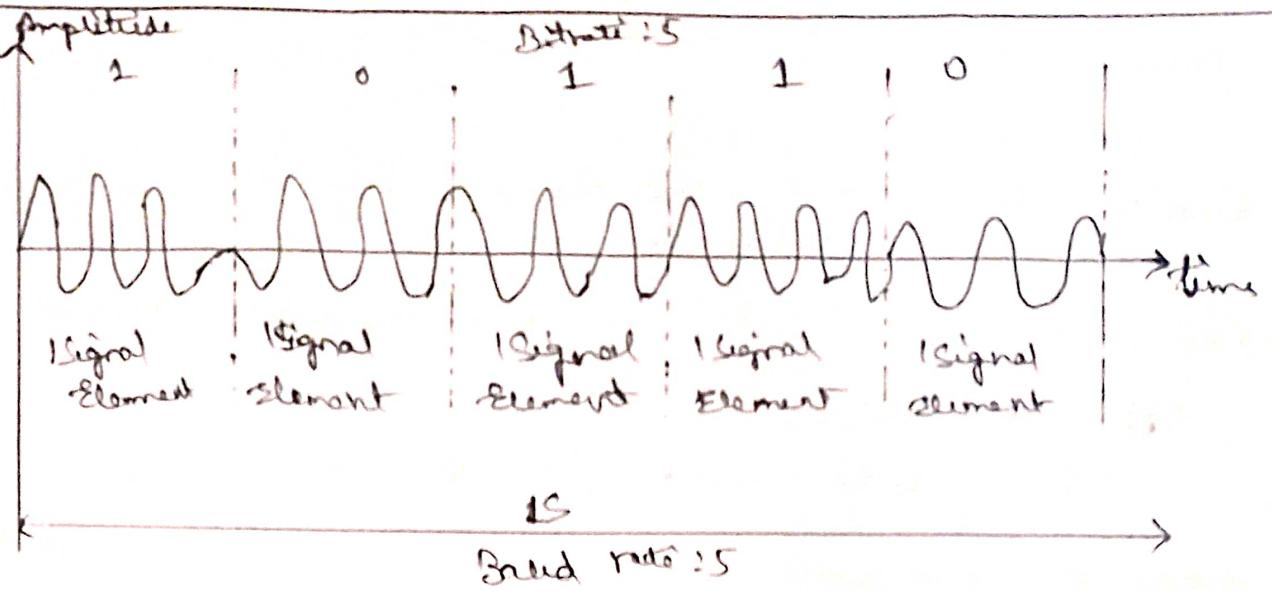
* Both amplitude and frequency remain constant for all signal elements.

\Rightarrow Binary PSK

* we have only two signal elements.

1). First Signal-Element with a phase of 0°

2). Second Signal-Element with a phase of 180° .



* Bandwidth of BPSK is same as BASK i.e.

$$B = (1+d) \times s$$

s: Signal rate d (0 < d < 1)

(d) Quadrature Amplitude Modulation (QAM)

* it is a combination of ASK and PSK,

* using 2 carriers one in the phase and the other quadrature, with different amplitude levels of the each carrier.

* Band width of QAM is same as ASK and PSK i.e

$$B = (1+d) \times s$$

② Four 1 kbps connections are multiplexed together. A unit is given find, (i) duration of 1 bit before multiplexing.

(ii) transmission rate of link (iii) Duration of each time slot.

> we can answer the question as follows,

(i) the duration of 1 bit before multiplexing is $1/1 \text{ kbps}$, or 0.001 s (1 ms)

(ii) The rate of link is 4 times the rate of connection of 14 kbps .

(iii) the duration of each time slot is one-fourth of the duration of each bit before multiplexing, or 114 ms or

550ms.

(B)

Explain the Encoder and decoder logic of cyclic redundancy check (CRC). Coding with neat diagram.

* CRC is a cyclic code, that is used in networks such as LANs and WANs.

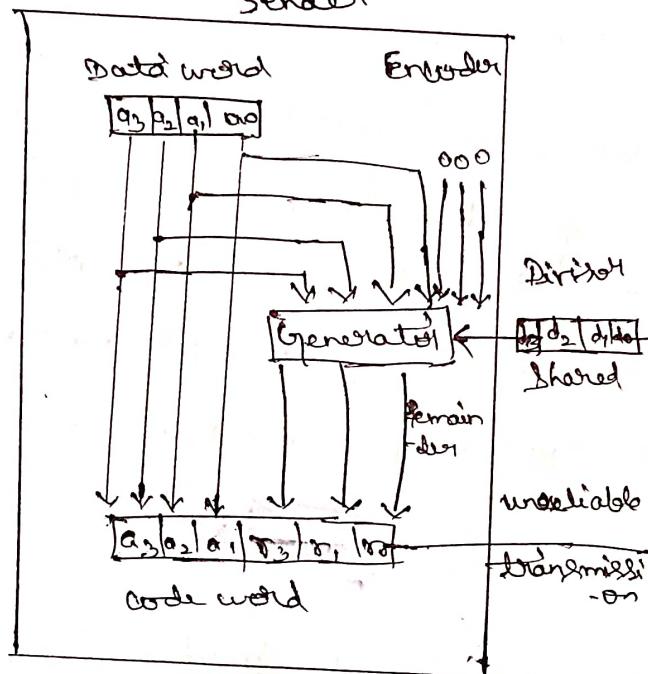
* Let size of data word = k bits (here $k=4$).

* Size of code word = n bits (here $n=7$).

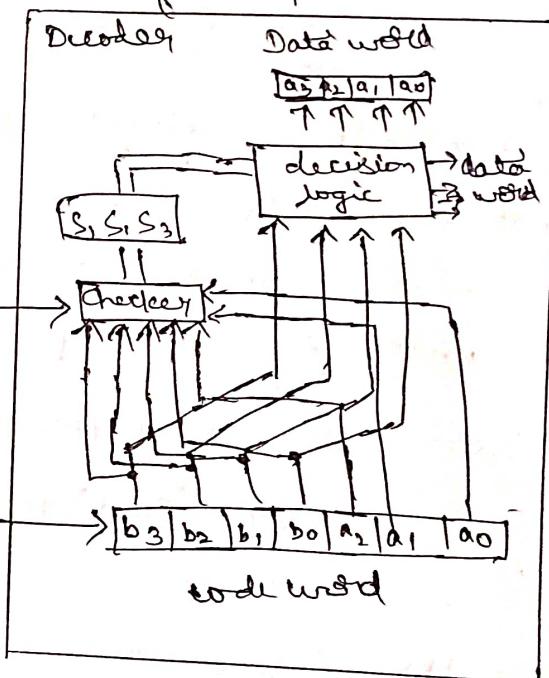
* Size of divisor : $n-k+1$ bits (here $n-k+1=4$)

* Here is how it works.

Sender



Receiver



CRC Encoder & decoder.

1). At Sender

* $n-k$'s is appended to the data word to create augmented data word (here $n-k=3$)

* the augmented data word is fed into the generator

* The generator divides the augmented data word by divisor.

* The remainder is called check bits ($r_3 | r_2 | r_1 | r_0$)

* The check bits ($r_3 | r_2 | r_1 | r_0$) are appended to the data word to create the code word.

At receiver:

- * The possibly corrupted code-word is fed into the checker.
- * The checker is a replica of the generator.
- * The checker divides the code-word by the divisor.
- * The remainder is called syndrome bits (r_2, r_1, r_0)
- * The syndrome bits are fed to the decision-logic analyzer.
- * The decision logic analyzer performs following functions:
 - (i) for no error.
 - (ii) for error.

- (i) for no error.
- * If all syndrome bits are 0s, the received code-word is accepted.
 - * Data word is extracted from received code-word.
- (ii) for error.
- * If all syndrome bits are not 0s, the received code-word is discarded.

Encoding: Data word $\boxed{10011}$

Multiply: AND
Subtract: XOR

Divisor: $1011 \overline{)10011000} \leftarrow \text{divided}$

$$\begin{array}{r} 0100 \\ 0000 \\ \hline 1000 \end{array}$$

leftmost bit 0
use 0000 divisor

leftmost bit 0
use 0000 divisor

$$\begin{array}{r} 0110 \\ 0000 \\ \hline 110 \end{array}$$

↓
remainder

Code word $\boxed{100110}$

Division in CRC encoder.

uncorrupted
code word $\boxed{1001110}$

Decoder

$$1011 \overline{) 1001110} \leftarrow \text{code word}$$

$$\begin{array}{r} 1011 \\ 1011 \\ \hline 0101 \\ 0000 \\ \hline 1011 \\ 1011 \\ \hline 0000 \\ 0000 \\ \hline \end{array}$$

Zero $\boxed{000}$ syndrome

\downarrow
Data word $\boxed{1001}$
accepted

corrupted
Code word $\boxed{1000110}$

$$1011 \overline{) 1000110}$$

$$\begin{array}{r} 1011 \\ 1011 \\ \hline 0111 \\ 0000 \\ \hline 1111 \\ 1011 \\ \hline 0000 \\ 1011 \\ \hline \end{array}$$

non-zero $\boxed{1011}$ syndrome

\downarrow
Data word discarded

Division in the CRC decoder for two cases.

(Q) Given data word 1001 and divisor 1011

- (i) Show the generation of the code word at the sender site.
- (ii) Show the checking of code word at receiver site (assume no error).

> (i) Data word 1001

Divisor 1011

$$1011 \overline{) 1001000}$$

$$\begin{array}{r} 1011 \\ 1011 \\ \hline 0100 \\ 0000 \\ \hline 1000 \\ 1011 \\ \hline 0110 \\ 0000 \\ \hline \end{array}$$

Code word = Data word + Remainder
 $\therefore 1001 + 110 = 1001110$

(ii) Data word 1001110

Divisor 1011

$$1011 \overline{) 1001110}$$

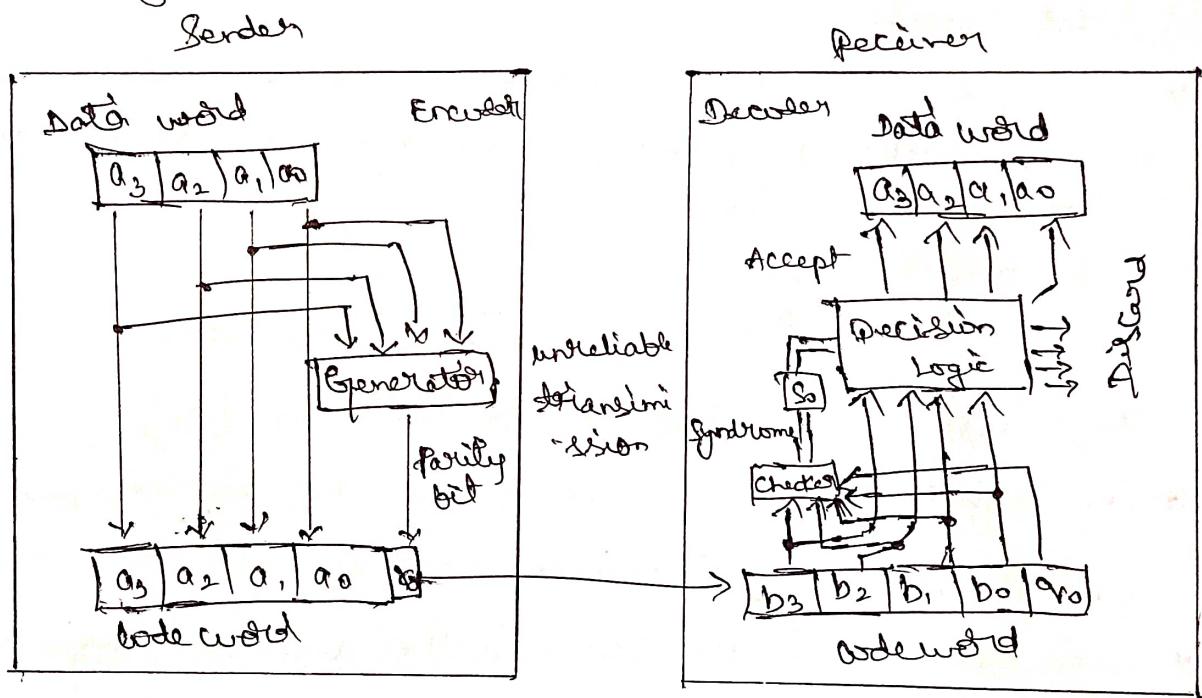
$$\begin{array}{r} 1011 \\ 1011 \\ \hline 0101 \\ 0000 \\ \hline 1011 \\ 1011 \\ \hline 0000 \\ 0000 \\ \hline \end{array}$$

$\therefore 1011$ zero (no error)

5

Explain with neat diagram, parity check

- * This code is a linear block code. This code can detect an odd number of errors.
- * A k -bit data-word is changed to an n -bit code word where $n = k + 1$.
- * One extra bit is called the parity bit.
- * The parity bit is selected to make the total number of 1's in the code-word even.
- * Minimum Hamming distance $d_{\min} \leq 2$, this means the code is a single-bit error detecting code receiver.



Encoder and decoder for Simple parity check code

Here is how it works

1. Sender

- * The encoder uses a generator that takes a copy of a 4-bit data word (a_0, a_1, a_2, a_3) and generates a parity bit a_4 .
- * The encoder → accepts a copy of a 4-bit data word (a_0, a_1, a_2 & a_3) and generates a parity bit using a generator → generates a 5-bit code word.

- * The parity bit and 4-bit data word are added to make the number of 1's in the code word even to $c_0 = a_0 + a_1 + a_2 + a_3$, the result of addition is the parity bit.
- * If the no. of 1's in data word = even, result = 0 ($r_0 = 0$)
- * If the no. of 1's in data word = odd, result = 1 ($r_0 = 1$)
- * In both cases, the total number of 1's in the code-word is even.
- * The sender sends, the code-word, which may be corrupted during transmission.

(2) At receiver

- * The receiver receives a 5-bit word.
- * The checker performs the same operation as the generator with one exception.
The addition is done over all 5-bits,

$$S_0 = b_3 + b_2 + b_1 + b_0 + r_0 \pmod{2}$$
- The result is called the Syndrome bit (S_0)
- * Syndrome bit = 0 when no. of 1's in the received code word is even; otherwise it is 1.
- * The Syndrome is passed to the decision logic analyzer.
 - If $S_0 = 0$ there is no error in the received code-word
the data portion of the received code-word is accepted as data word
 - If $S_0 = 1$ there is error in the received code-word
the data portion of the received code-word is discarded
the data word is not created.

(6)

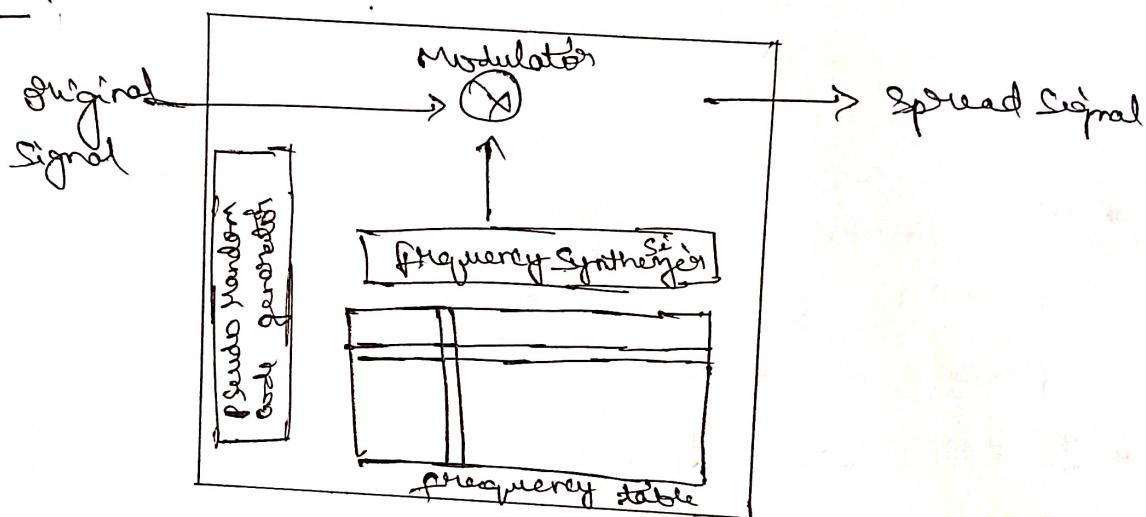
what is spread spectrum? explain
example.

FHSS and DSSS with

Multiplexing combines signals from several sources to achieve bandwidth efficiency; the available bandwidth of a link is divided b/w the sources.

There are two types of spread spectrum.

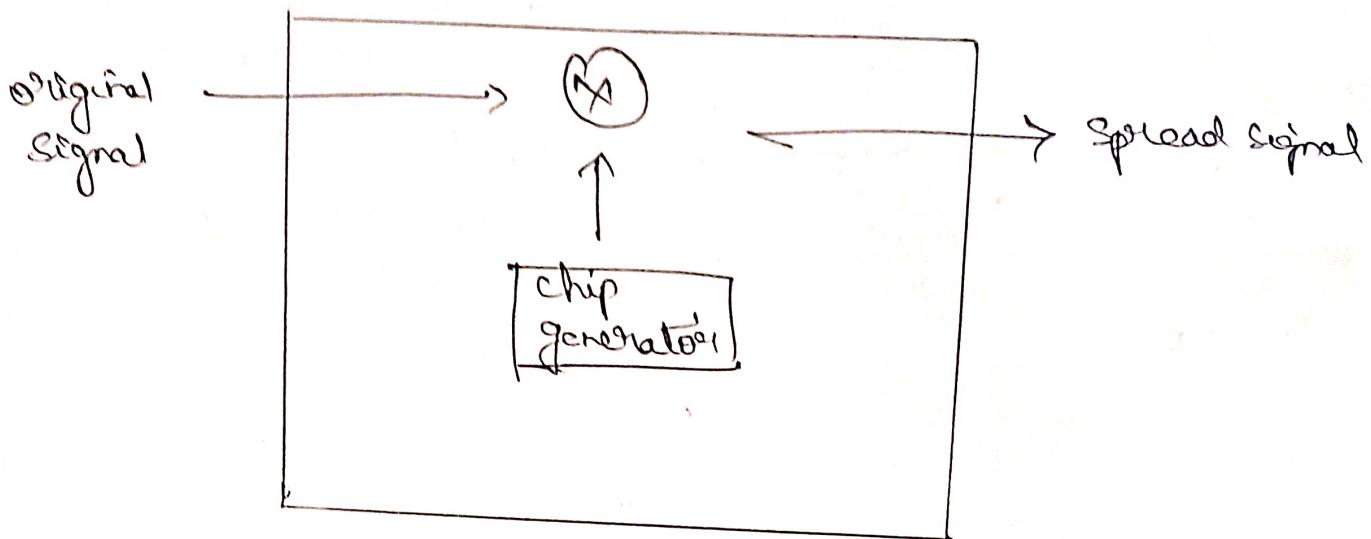
- frequency hopping spread spectrum (FHSS).
- Direct Sequence Spread Spectrum (DSSS).
- FHSS:



- * The frequency hopping spread spectrum technique uses different carrier frequencies that are modulated by the source signal.
- * Figure shows the general layout of FHSS. A random pseudo code generator, called Pseudo Random noise (PRN), creates a k-bit for every period.
- * The frequency table uses the pattern to find the frequency to be used for this hopping period and passes it to the Frequency Synthesizer.
- * The frequency synthesizer treats a carrier signal of that frequency, and source signal modulates the carrier signal.

- * The pseudo-random code generator will create eight different 3-bit patterns. These are mapped to 8 different frequency in frequency table.

(b) DSSS:



- * The direct sequence spread spectrum technique also expands the bandwidth of the original signal. but the process is direct.
- * In DSSS, we replace each data bit with 11 bits using a spreading code.
- * An example, let us consider the sequence used in a wireless LAN, the famous Barker sequence where n is 11 we assume that the original signal and the chips in the chip generator use polar NRZ encoding.
- * Bandwidth sharing: Can we share a bandwidth in DSSS as we did in FHSS? The answer is no and yes. If we use a spreading code that spreads signals that cannot be combined and separated we cannot share a bandwidth.