Spread spectrum

(EP2950)

KTH Technology
and Health

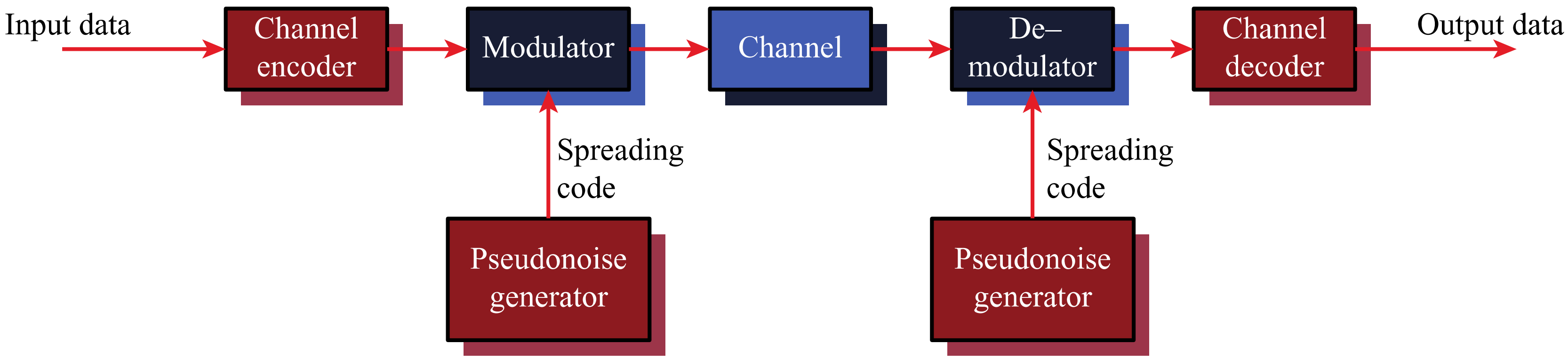
Outline

- Introduction
- Frequency hopping spread spectrum
- Direct sequence spread spectrum
- Code division multiple access (CDMA)
- Spreading sequences and orthogonal codes

Spread spectrum

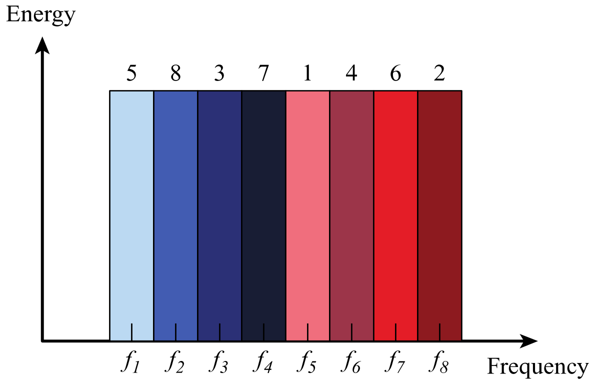
- The signal is modulated using a sequence of digits
  - ✓ Spreading code generated by pseudo-random number generator
- Effect of modulation is to increase bandwidth of the signal
  - ✓ Frequency hopping
  - ✓ Direct sequence
- On the receiving end, the same digit sequence is used to demodulate the spread spectrum (de-spreading)
- Advantages
  - Protection against multipath distortion
  - Hiding and encrypting signals
  - Multiplexing several users with low interference

Model of spread spectrum communication

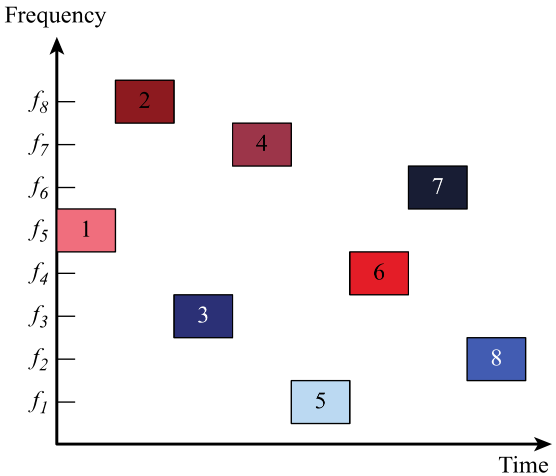


Frequency hopping spread spectrum

- The signal is broadcasted over seemingly random sequence of subchannels
- Channel sequence determined by the spreading code
- The transmitter and receiver switches from frequency to frequency at fixed intervals



(a) Channel assignment



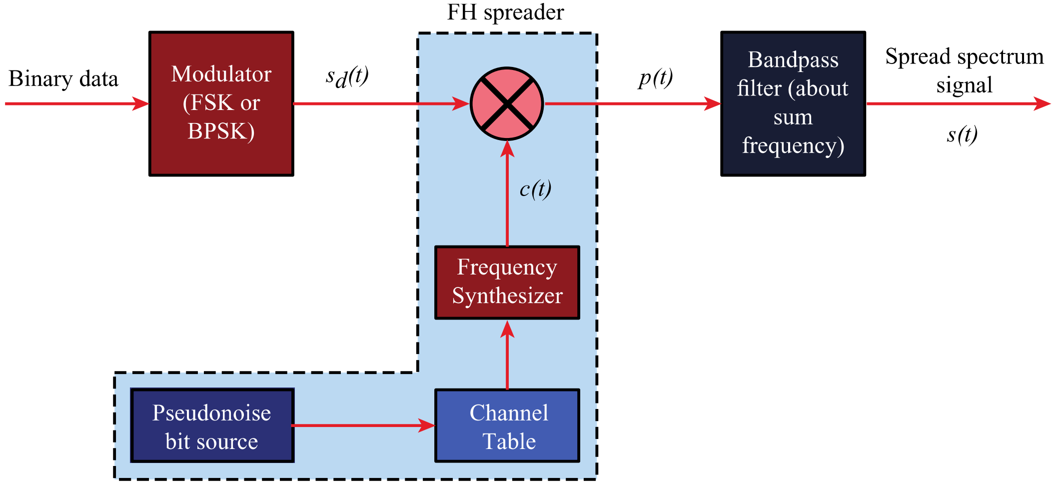
(b) Channel use

Processing gain for FHSS

$$G_p = 2^k = \frac{W_s}{W_d}$$

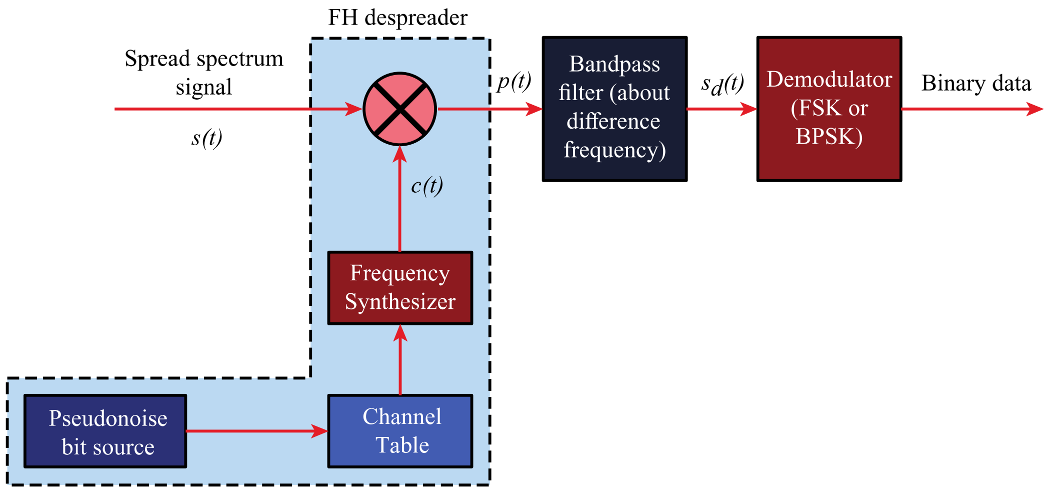
k = number of bits in spreading code
2^k = number of spreading channels
W_s = spread bandwidth
W_d = unspread bandwidth

- Frequency hopping system



(a) Transmitter

Sender



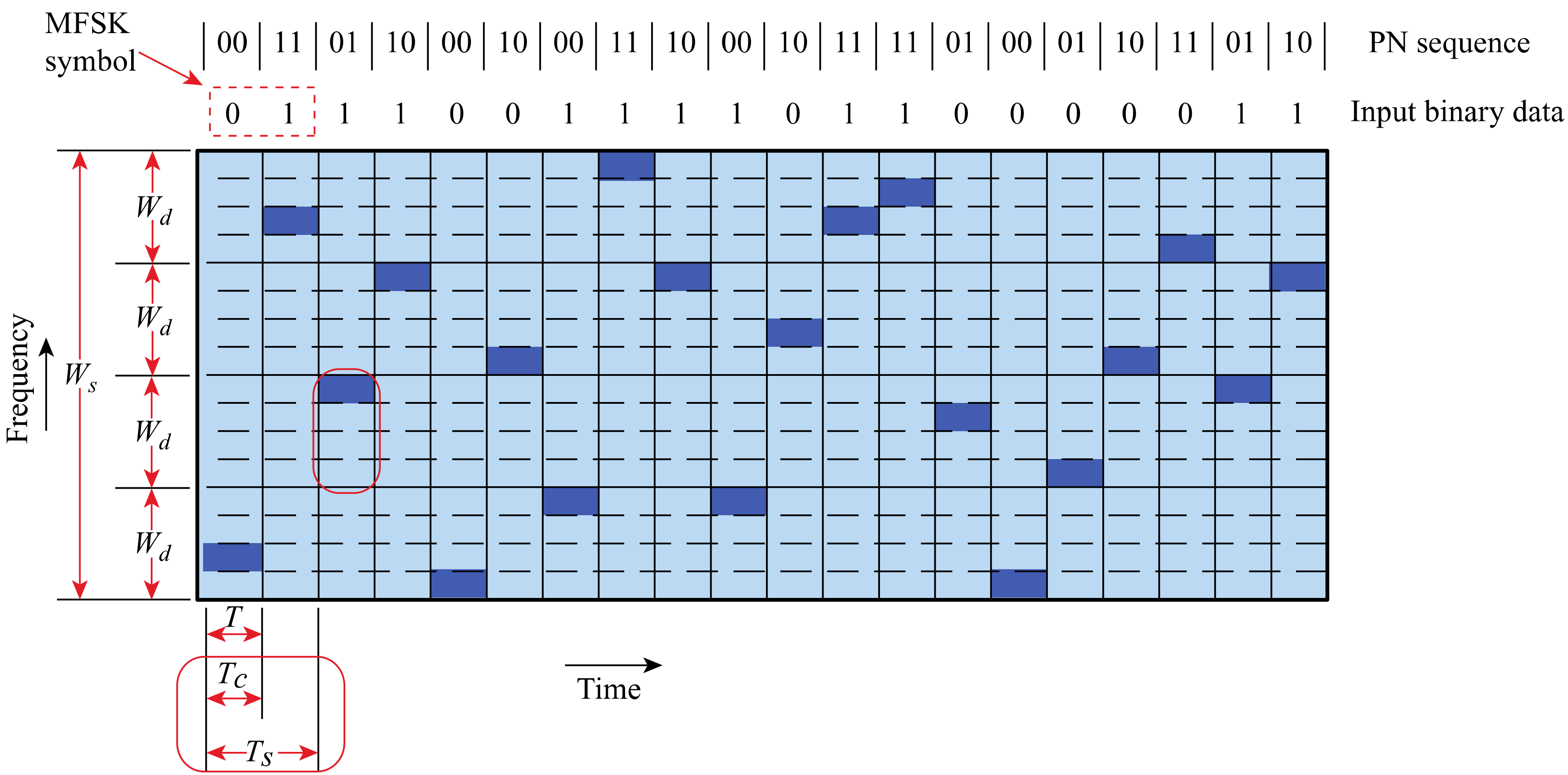
(b) Receiver

Receiver

FHSS using multiple FSK

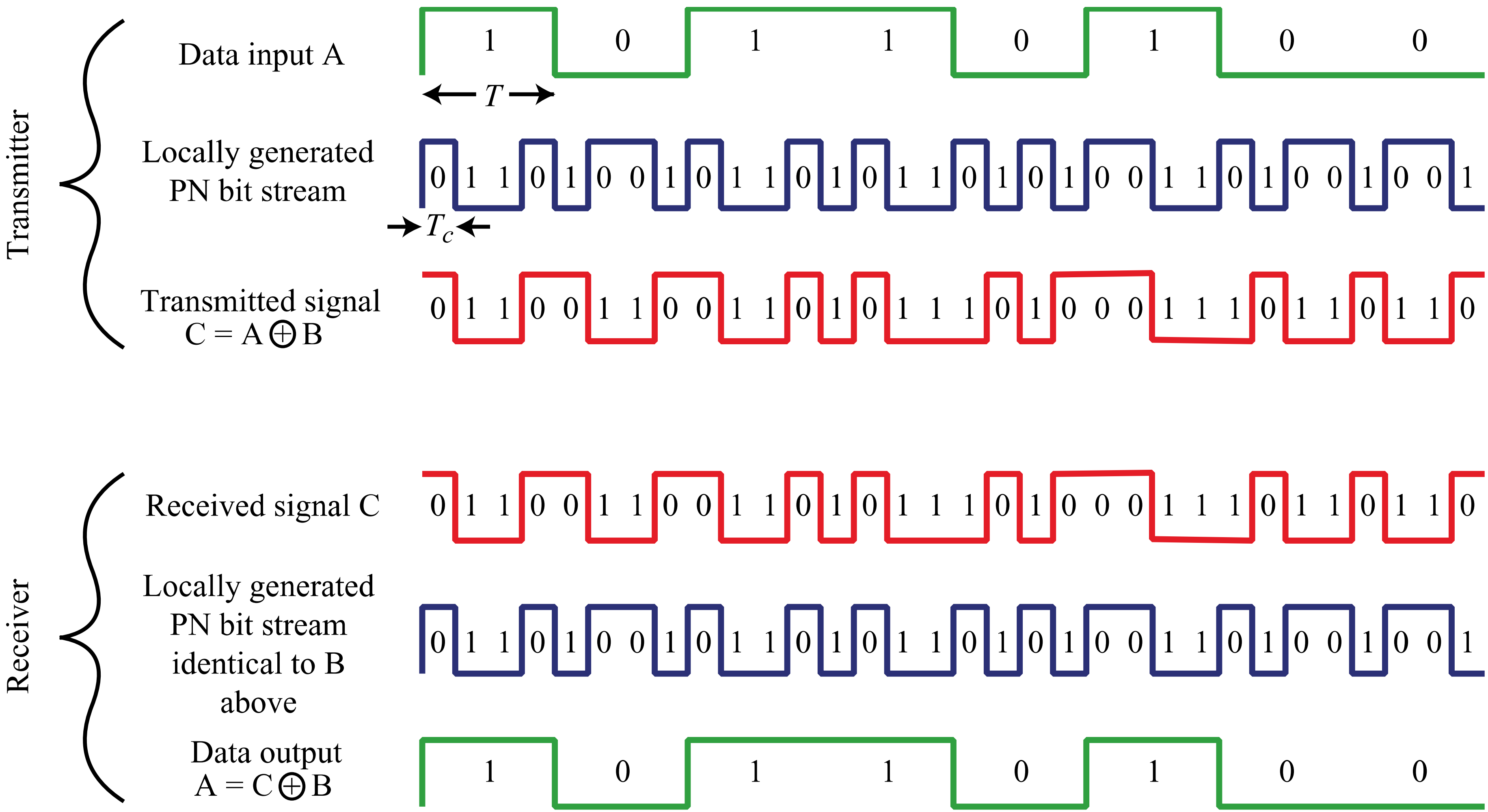
- Multiple frequency shift keying
  - ✓ New frequency every T_c
- Data rate R
  - ✓ Duration of a bit: $T = 1/R$ seconds
  - ✓ Duration of symbol: $T_s = LT$ seconds
  - ✓ L = nr of bits per signal element (symbol)
  - ✓ $M = 2^L$ (number of symbols)
- $T_c \geq T_s$ - slow-frequency-hop spread spectrum
- $T_c < T_s$ - fast-frequency-hop spread spectrum

- Fast frequency-hopping using MFSK (M=4, k=2)



Direct sequence spread sequence

- Each bit in the original signal is represented by multiple bits in the transmitted signal
- Spreading code spreads the signal across a wider frequency band
  - ✓ The spreading is proportional to the number of bits in the spreading code
- One technique is to XOR (exclusive-OR) the data stream and the spreading code
- Processing gain
  - ✓ $G_p = \frac{W_s}{W_d}$, where $\mathrm{W_s}$ = spread bandwidth and $\mathrm{W_s}$ = signal bandwidth



Transmitter
Data input A
1 0 1 1 0 1 0 0
T
Locally generated PN bit stream
0 1 1 0 1 0 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 0 1 1 0 1 0 0 1 0 0 1
T_c
Transmitted signal
C = A ⊕ B
0 1 1 0 0 1 1 0 0 1 1 0 1 0 1 1 1 0 1 0 0 0 1 1 1 0 1 1 0 1 1 0
Receiver
Received signal C
0 1 1 0 0 1 1 0 0 1 1 0 1 0 1 1 1 0 1 0 0 0 1 1 1 0 1 1 0 1 1 0
Locally generated PN bit stream identical to B above
0 1 1 0 1 0 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 0 1 1 0 1 0 0 1 0 0 1
Data output
A = C ⊕ B
1 0 1 1 0 1 0 0

Code division multiple access

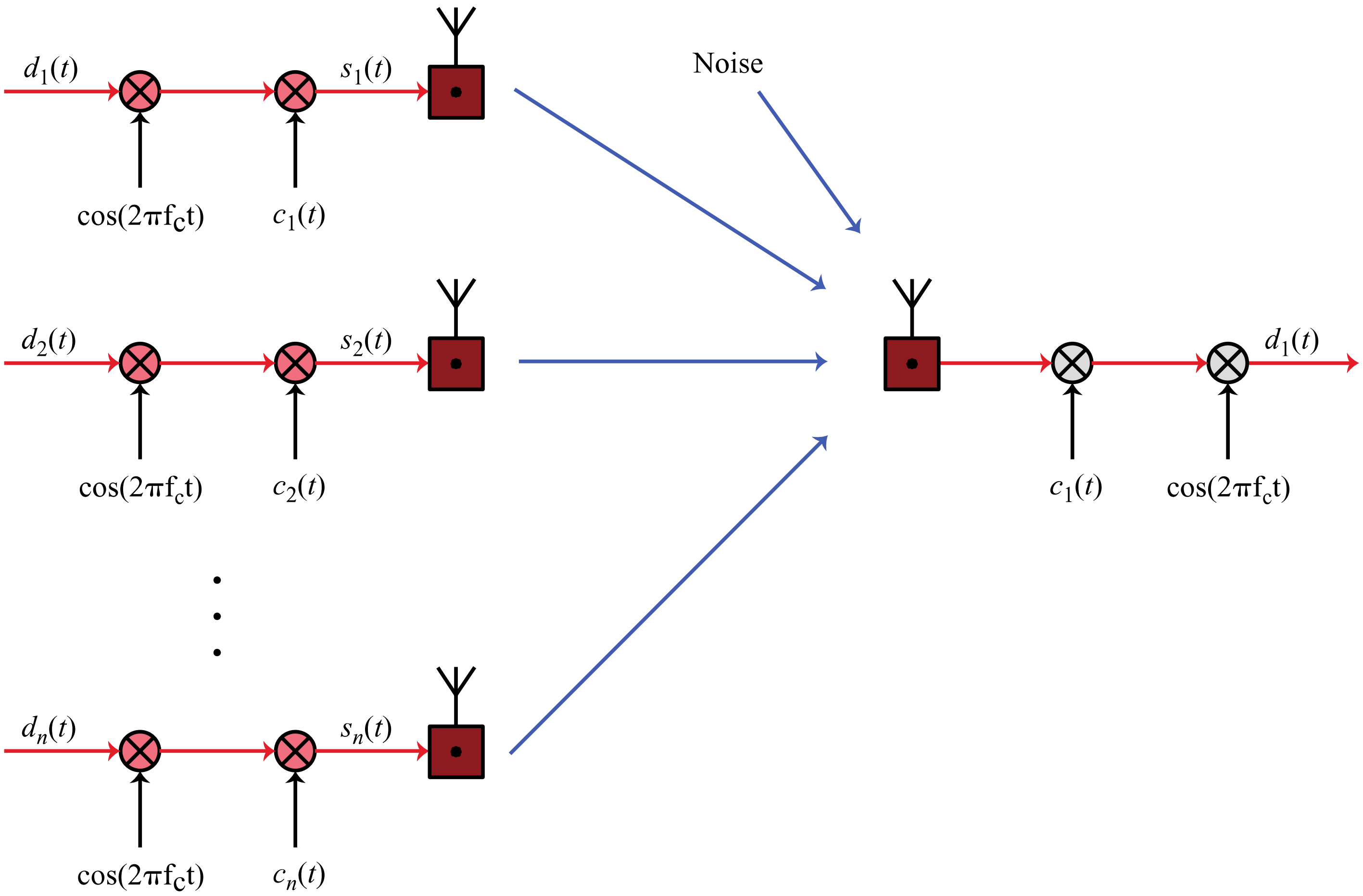
- Basic idea is multiplexing of several users
- Data bit rate D (bps)
- Each bit is divided into k chips
  - ✓ Chip sequence is a fixed user-specific pattern
- Chip rate = $k \times D$

- Example using CDMA
- If k=6 and code is a sequence of 1's and -1's
- User A represents a 1-bit with the chip pattern code
  - ✓ <c1, c2, c3, c4, c5, c6>
- User A represents a 0-bit with the complement chip sequence
  - ✓ <-c1,-c2, -c3, -c4, -c5, -c6>
- Receiver knows the sender's code and decodes the received chip sequence

$$S_u(d) = d1 \times c1 + d2 \times c2 + d3 \times c3 + d4 \times c4 + d5 \times c5 + d6 \times c6$$

- ✓ <d1, d2, d3, d4, d5, d6> = received chip pattern
- ✓ <c1, c2, c3, c4, c5, c6> = sender's code

CDMA system



$d_1(t)$
$s_1(t)$
$\cos(2\pi f_c t)$
$c_1(t)$
Noise
$d_2(t)$
$s_2(t)$
$\cos(2\pi f_c t)$
$c_2(t)$
$d_1(t)$
$c_1(t)$
$\cos(2\pi f_c t)$
$d_n(t)$
$s_n(t)$
$\cos(2\pi f_c t)$
$c_n(t)$

- CDMA example – user A sends bits to receiver R
- User A code = <1, –1, –1, 1, –1, 1>
  - ✓ To send a 1 bit = <1, –1, –1, 1, –1, 1>
  - ✓ To send a 0 bit = <–1, 1, 1, –1, 1, –1>
- User B code = <1, 1, –1, – 1, 1, 1>
  - ✓ To send a 1 bit = <1, 1, –1, –1, 1, 1>
  - ✓ To send a 0 bit = <-1, -1, 1, 1, -1, -1>
- The receiver R performs
  - ✓ (A’s code) x (received chip pattern)
    User A’s 1-bit becomes 6 – interpreted as 1
    User A’s 0-bit becomes -6 – interpreted as 0
    User B’s 1-bit becomes 0 which is ignored
    User B’s 0-bit becomes 0 which is ignored

CDMA example

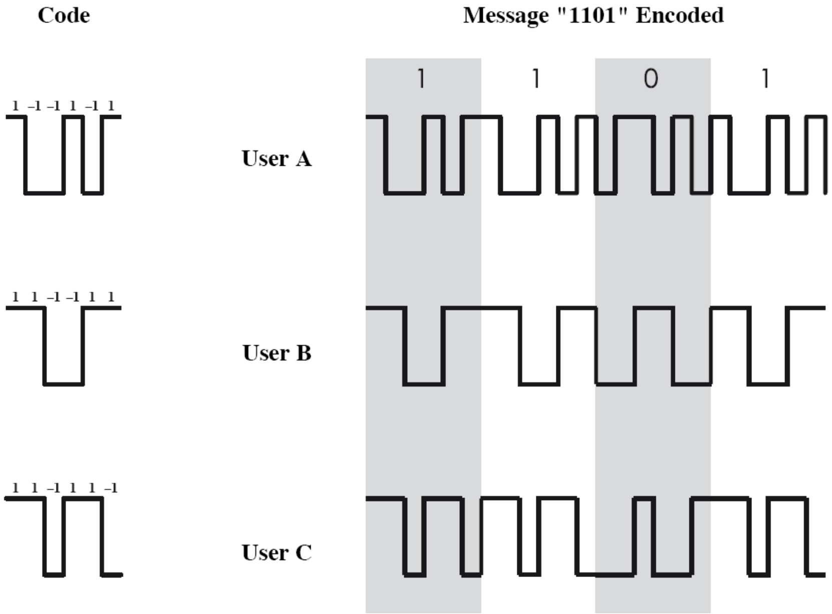


Table 7.1 CDMA Example

(a) User's codes

User A	1	–1	–1	1	–1	1
User B	1	1	–1	–1	1	1
User C	1	1	–1	1	1	–1

(b) Transmission from A

Transmit (data bit = 1)	1	–1	–1	1	–1	1	
Receiver codeword	1	–1	–1	1	–1	1	
Multiplication	1	1	1	1	1	1	= 6

Transmit (data bit = 0)	–1	1	1	–1	1	–1	
Receiver codeword	1	–1	–1	1	–1	1	
Multiplication	–1	–1	–1	–1	–1	–1	= –6

(c) Transmission from B, receiver attempts to recover A's transmission

Transmit (data bit = 1)	1	1	–1	–1	1	1	
Receiver codeword	1	–1	–1	1	–1	1	
Multiplication	1	–1	1	–1	–1	1	= 0

(d) Transmission from C, receiver attempts to recover B's transmission

Transmit (data bit = 1)	1	1	–1	1	1	–1	
Receiver codeword	1	1	–1	–1	1	1	
Multiplication	1	1	1	–1	1	–1	= 2

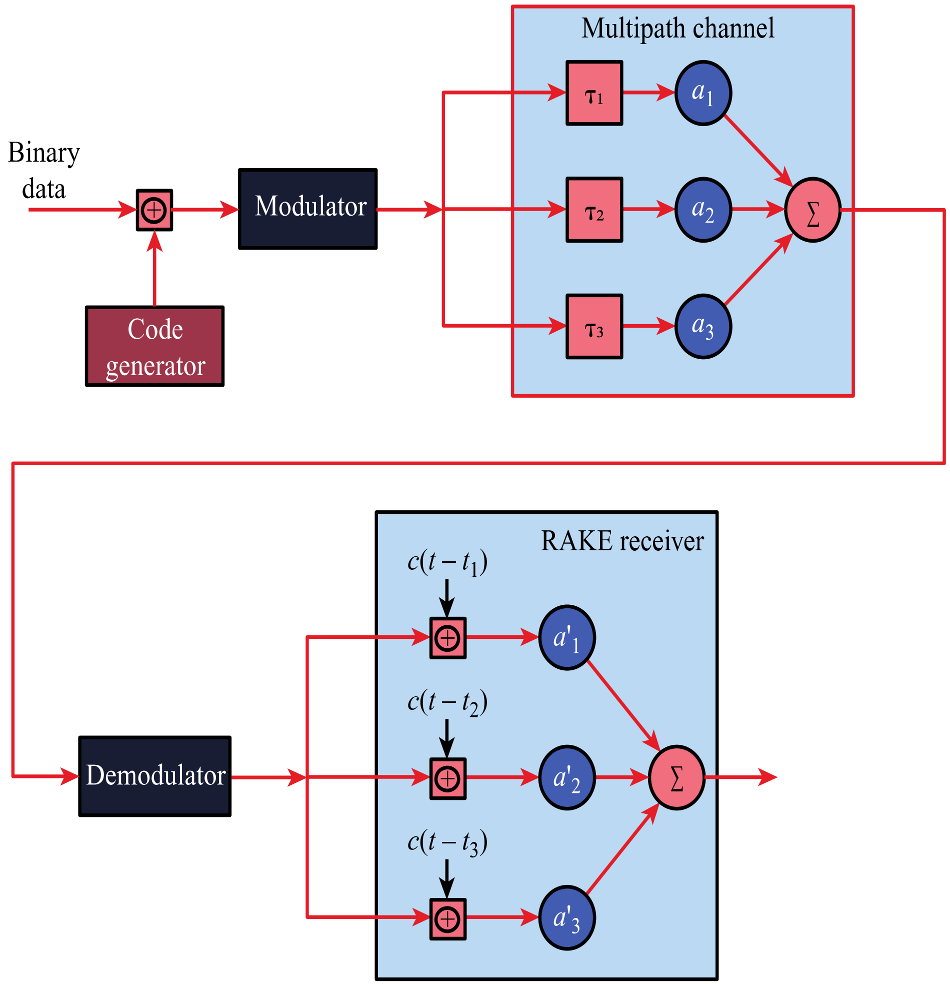
(e) Transmission from B and C, receiver attempts to recover B's transmission

B (data bit = 1)	1	1	–1	–1	1	1	
C (data bit = 1)	1	1	–1	1	1	–1	
Combined signal	2	2	–2	0	2	0	
Receiver codeword	1	1	–1	–1	1	1	
Multiplication	2	2	2	0	2	0	= 8

Rake receiver

- RAKE receiver
  - Multiple versions of a signal arrive more than one chip interval apart
  - RAKE receiver attempts to recover signals from multiple paths and combine them
- This method achieves better performance than simply recovering dominant signal and treating remaining signals as noise

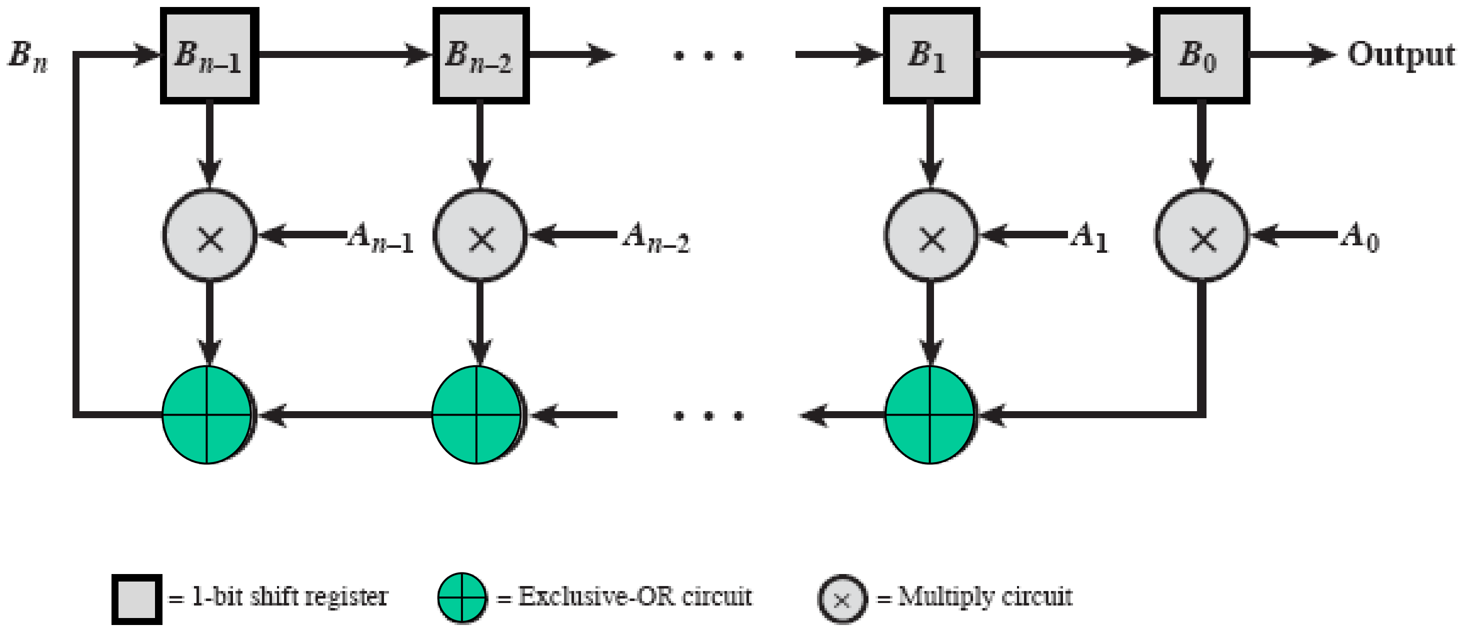
Rake receiver



Pseudo-noise (PN) sequences

- A PN generator produces a periodic sequence that appears to be random
- PN sequences
  - ✓ Generated by an algorithm using initial seed
  - ✓ Seemingly random
  - ✓ Referred to as pseudorandom numbers or pseudo-noise sequences
- PN properties
  - ✓ Uniform distribution (balance and run property)
  - ✓ Independence
  - ✓ Correlation property

- Linear feedback shift register implementation
  - ✓ Maximum-length sequence (m-sequence)



Properties of m-sequences

- Property 1
  - ✓ Has 2^{n-1} ones and 2^{n-1} minus ones (zeros)
- Property 2
  - ✓ For a window of length n that slides along output for N ($=2^{n-1}$) shifts, each n-tuple appears once, except for the all zeros sequence
- Property 3
  - ✓ Sequence contains one run of ones, length n
  - ✓ One run of zeros, length n-1
  - ✓ One run of ones and one run of zeros, length n-2
  - ✓ Two runs of ones and two runs of zeros, length n-3
  - ✓ 2^{n-3} runs of ones and 2^{n-3} runs of zeros, length 1

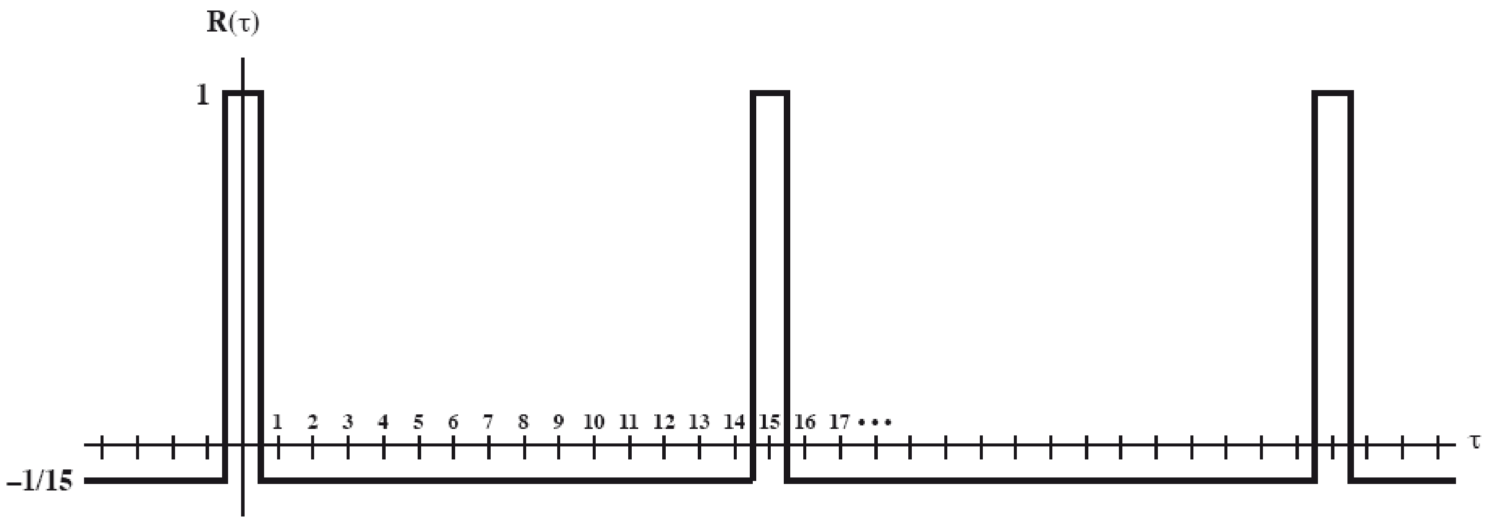
- Property 4
  - ✓ The periodic autocorrelation of a ± 1 m-sequence is

$$R(\tau) = \begin{cases} 1 & \tau = 0, \mathrm{N}, 2\mathrm{N}, \ldots \\ -\dfrac{1}{N} & \text{otherwise} \end{cases}$$

- Correlation
  - ✓ The level of similarity a set of data has with another or itself (autocorrelation)
  - ✓ Range between –1 and 1
    - 1 The second sequence matches the first sequence
    - 0 There is no relation at all between the two sequences
    - -1 The two sequences are mirror images
  - ✓ Cross-correlation
    - The comparison between two sequences from different sources rather than a shifted copy of a sequence with itself

- Autocorrelation
  - ✓ Example with sequence length 15

$$R(\tau) = \frac{1}{N}\sum_{k=1}^{N} B_k\, B_{k-\tau}$$



- Advantages of cross-correlation
  - ✓ Low cross-correlation between an m-sequence and noise
    Noise can be filtered out
  - ✓ Low cross-correlation between two different m-sequences
    - Useful for CDMA applications
    - Enables a receiver to discriminate among spread spectrum signals generated by different m-sequences
  - ✓ Gold and Kasami codes improve cross-correlation property of m-sequences
  - ✓ Cross-correlation between A and B

$$R_{A,B}(\tau) = \frac{1}{N}\sum_{k=1}^{N} A_k \, B_{k-\tau}$$

Orthogonal codes

- All pairwise cross correlations are zero
- Fixed- and variable-length codes used in CDMA systems
- For CDMA application
  - ✓ Differentiate one mobile user from the others
  - ✓ Provides zero cross correlation among all users
- Types
  - ✓ Walsh-Hadamard codes
  - ✓ Variable-length orthogonal codes

Walsh codes

- A set of Walsh codes of length *n* consists of the *n* rows of an $n \times n$ Walsh-Hadamard matrix
  - $W_1 = (0)$
  - *N* codes of length *n* (dimension)
  - Overscore denotes complement (logical NOT)

$$W_{2n} = \begin{pmatrix} W_n & W_n \\ W_n & \overline{W_n} \end{pmatrix}$$

- Every row is orthogonal to every other row and to the complement of every other
- Requires tight synchronization
  - Cross correlation between different shifts of Walsh sequences is not zero

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	1	0	1	1	0	1	0	0	1	0	1	1	0	1	0
0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
0	1	1	0	1	0	0	1	0	1	1	0	1	0	0	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0
0	0	1	1	0	0	1	1	1	1	0	0	1	1	0	0
0	1	1	0	0	1	1	0	1	0	0	1	1	0	0	1
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	1	0	1	1	0	1	0	1	0	1	0	0	1	0	1
0	0	1	1	1	1	0	0	1	1	0	0	0	0	1	1
0	1	1	0	1	0	0	1	1	0	0	1	0	1	1	0

Complement of W_8

- Orthogonal codes
  - ✓ τ is bit duration
  - ✓ M is length of the sequences φ_i and φ_j

$$\sum_{k=0}^{M-1} \varphi_i(k\tau)\,\varphi_j(k\tau) = 0$$

- Example
  - ✓ (0 1 0 1)×(0 0 1 1) → (-1 1 -1 1)×(-1 -1 1 1) = 0

- Variable-length orthogonal codes
  - ✓ Can be used to vary the data rates for the users
  - ✓ Orthogonal variable spreading factor (OVSF)
- Multiple spreading
  - ✓ Channelization codes, e.g. Walsh codes
    - ✓ Provides mutual orthogonality among all users in the same cell
  - ✓ Scrambling codes
    - ✓ PN sequences, e.g. Gold codes
    - ✓ Provides mutual randomness (low cross correlation) between users in different cells