Linear Feedback Shift Registers

- Linear Feedback Shift Registers (LFSRs) are shift registers with feedback that is composed of a network of XOR (or XNOR) gates
- If we choose that feedback in a specific manner, the contents of the shift regsiter form a pseudo-random sequence
- This is VERY useful for a wide variety of applications
- LFSRs are used in a HUGE number of digital circuits and are one of the most important digital structures.

LFSR

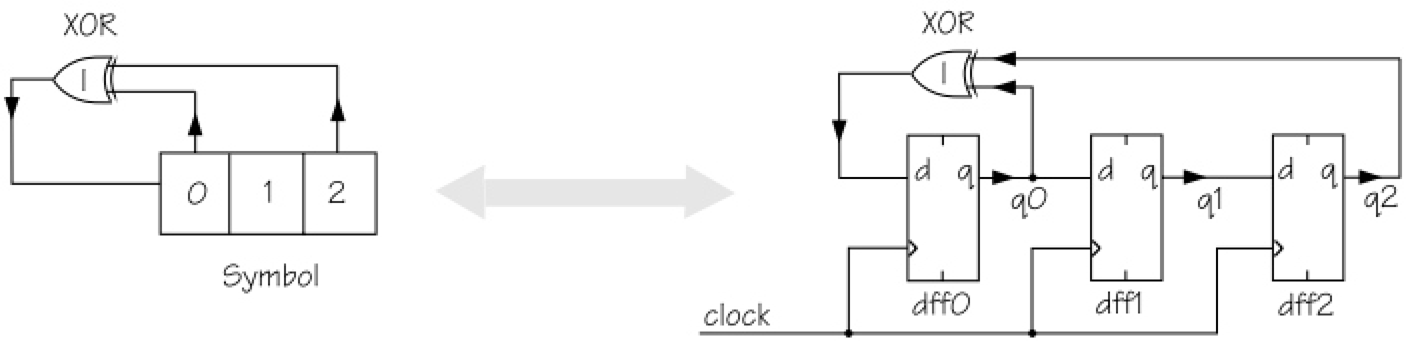
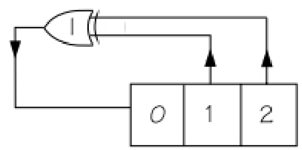
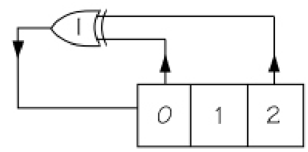


Figure F-1. LFSR with XOR feedback path

LFSRs



Clock Edge	q0	q1	q2	
—	0	0	1	← Initial Value
1	1	0	0	
2	0	1	0	
3	1	0	1	
4	1	1	0	
5	1	1	1	
6	0	1	1	
7	0	0	1	
8	1	0	0	

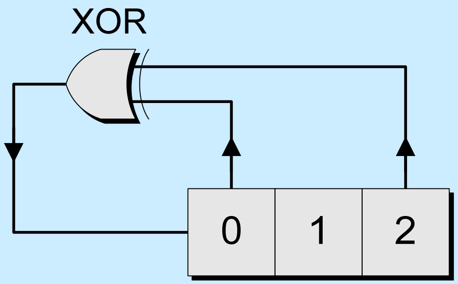


Clock Edge	q0	q1	q2	
—	0	0	1	← Initial Value
1	1	0	0	
2	1	1	0	
3	1	1	1	
4	0	1	1	
5	1	0	1	
6	0	1	0	
7	0	0	1	
8	1	0	0	

Figure F-2. Comparison of alternative tap selections

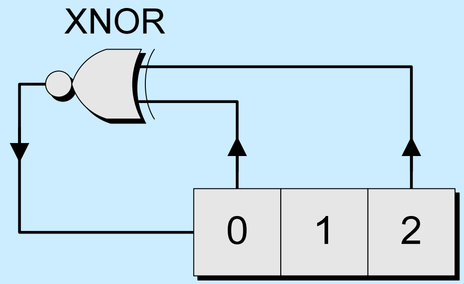
Properties

- When the feedback is properly chosen, the content of the LFSR is periodic with a period of 2^m-1, where *m* is the number of bits in the register.
- If *m* is very large, that period is very long (Example: $m=32$, $2^{32}-1=4.294.967.295$)
- We can take any bit of the LFSR, for example the LSB, and get a sequence of bits that is pseudo-random with a period of 2^m-1. Such a sequence is called a “maximal length sequence” or "m-sequence"
- If the feedback is chosen “incorrectly”, non maximal sequence outputs will result (the period will be less than 2^m-1)



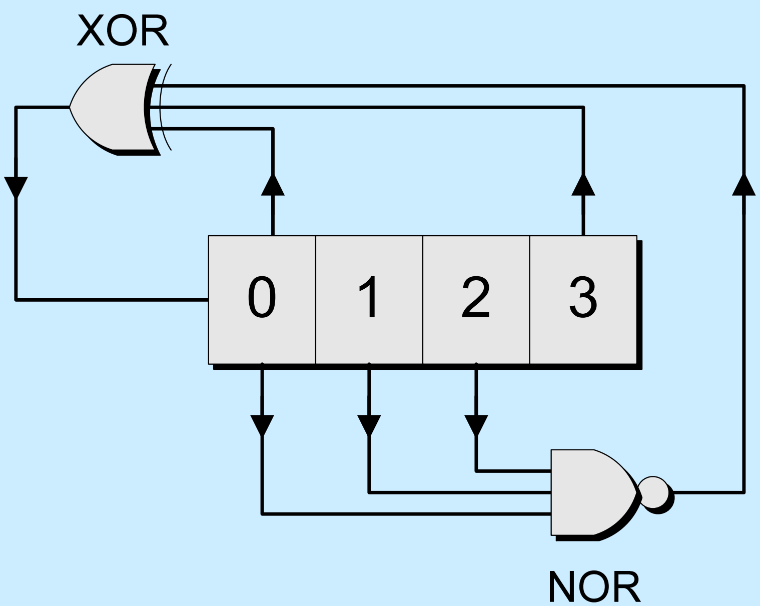
clock	q0	q1	q2	
---	0	0	1	← Start
1	1	0	0	
2	1	1	0	
3	1	1	1	← All 1s
4	0	1	1	
5	1	0	1	
6	0	1	0	
7	0	0	1	
8	1	0	0	
:	:	:	:	

(a) XOR with taps at [0,2]



clock	q0	q1	q2	
---	0	0	1	← Start
1	0	0	0	← All 0s
2	1	0	0	
3	0	1	0	
4	1	0	1	
5	1	1	0	
6	0	1	1	
7	0	0	1	
8	0	0	0	
:	:	:	:	

(b) XNOR with taps at [0,2]



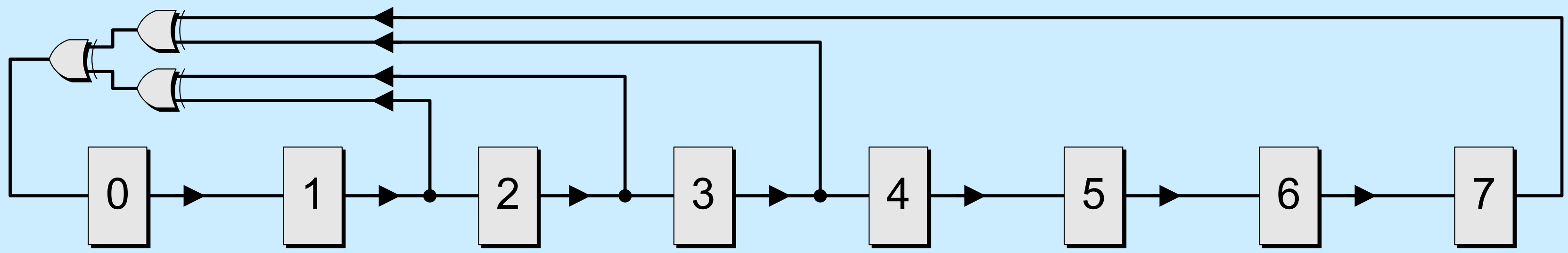
clock	q0	q1	q2	q3
--	0	0	0	1
1	0	0	0	0
2	1	0	0	0
3	1	1	0	0
4	1	1	1	0
5	1	1	1	1
6	0	1	1	1
7	1	0	1	1
8	0	1	0	1
9	1	0	1	0
10	1	1	0	1
11	0	1	1	0
12	0	0	1	1
13	1	0	0	1
14	0	1	0	0
15	0	0	1	0
16	0	0	0	1
17	0	0	0	0
:	:	:	:	:

All 0s

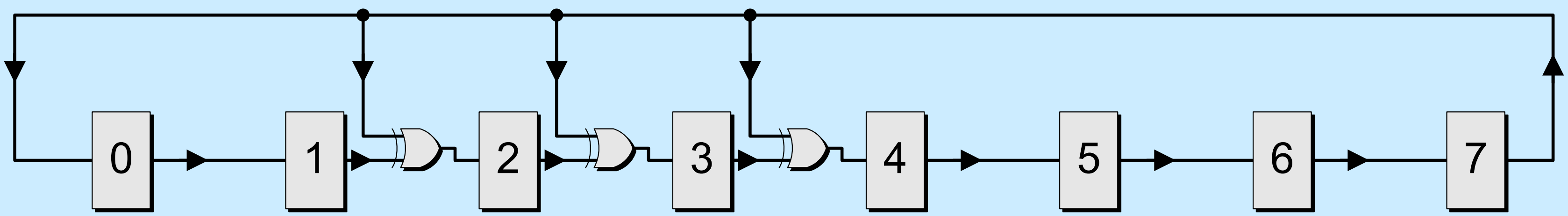
$2^4 = 16$ values

Figure C-09

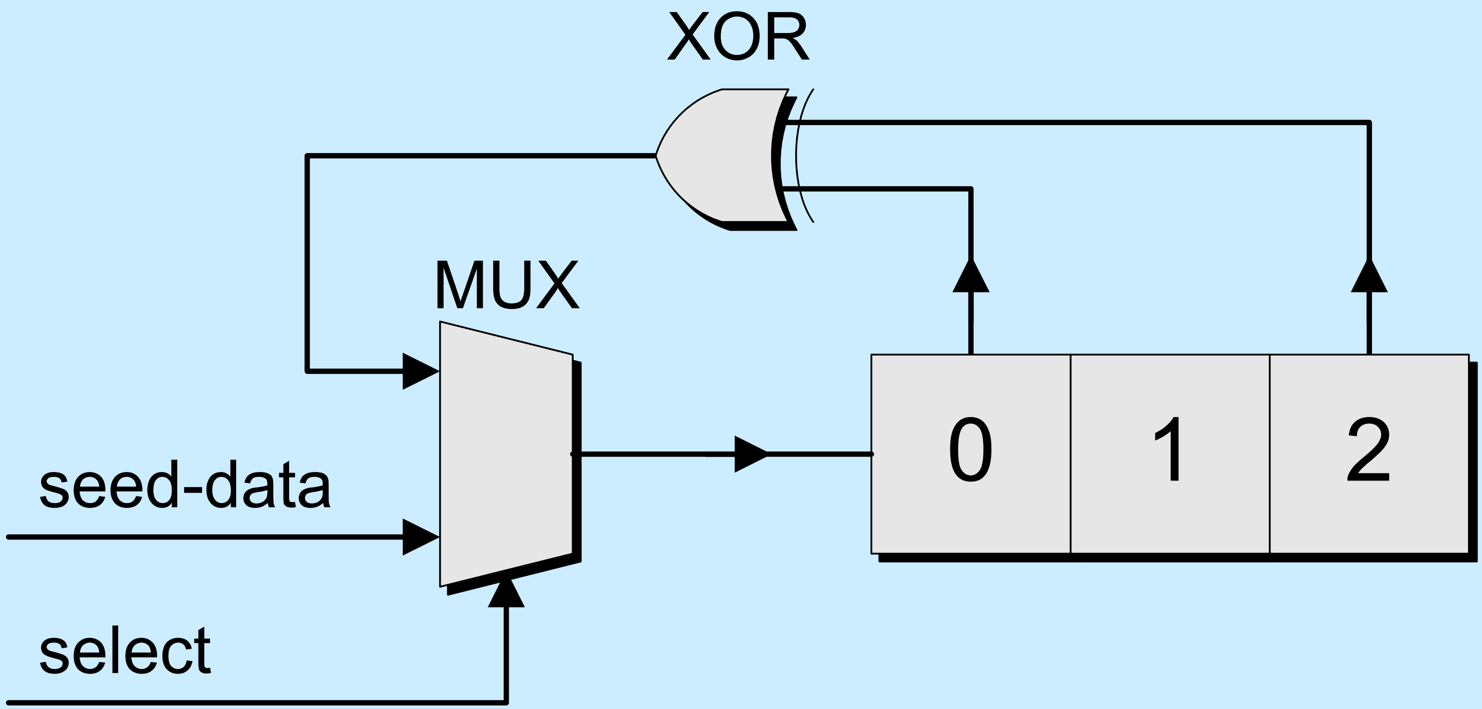
# Bits	Loop Length	Taps
2	3 *	[0,1]
3	7 *	[0,2]
4	15	[0,3]
5	31 *	[1,4]
6	63	[0,5]
7	127 *	[0,6]
8	255	[1,2,3,7]
9	511	[3,8]
10	1,023	[2,9]
11	2,047	[1,10]
12	4,095	[0,3,5,11]
13	8,191 *	[0,2,3,12]
14	16,383	[0,2,4,13]
15	32,767	[0,14]
16	65,535	[1,2,4,15]
17	131,071 *	[2,16]
18	262,143	[6,17]
19	524,287 *	[0,1,4,18]
20	1,048,575	[2,19]
21	2,097,151	[1,20]
22	4,194,303	[0,21]
23	8,388,607	[4,22]
24	16,777,215	[0,2,3,23]
25	33,554,431	[2,24]
26	67,108,863	[0,1,5,25]
27	134,217,727	[0,1,4,26]
28	268,435,455	[2,27]
29	536,870,911	[1,28]
30	1,073,741,823	[0,3,5,29]
31	2,147,483,647 *	[2,30]
32	4,294,967,295	[1,5,6,31]



(a) Many-to-one implementation



(b) One-to-many implementation


XOR
MUX
seed-data
select
0
1
2

Example - CRC

- CRC (Cyclic Redundancy Check) is used by many file system and networking algorithms to ensure data integrity.

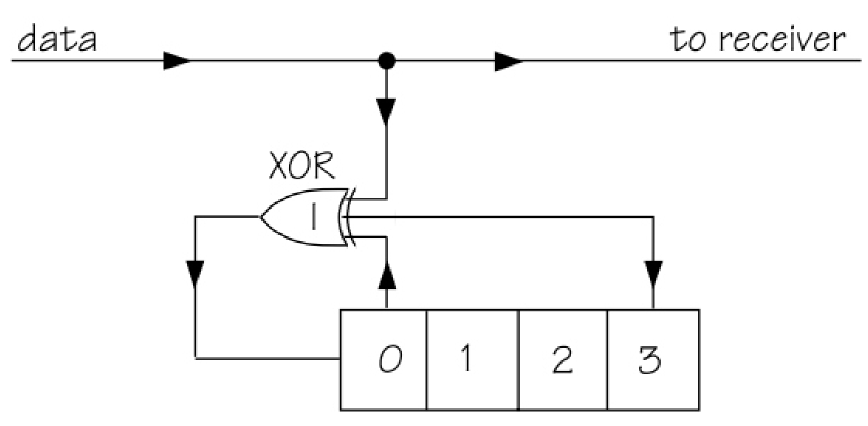
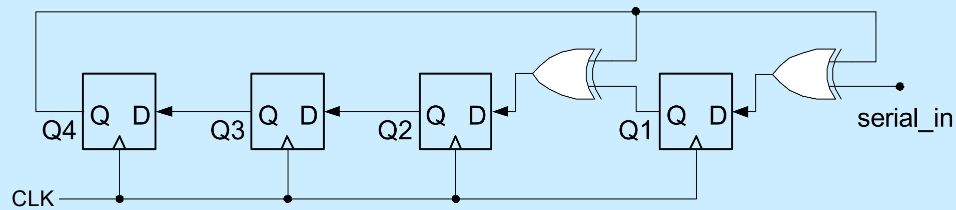


Figure F-12. Cyclic redundancy check (CRC) calculations

CRC encoding



1 0 1 1 0 0 1 0 0 0 0

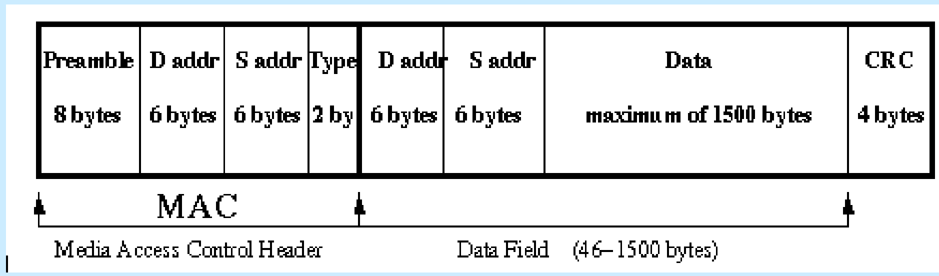
0	0	0	0	
0	0	0	1	0 1 1 0 0 1 0 0 0 0
0	0	1	0	1 1 0 0 1 0 0 0 0
0	1	0	1	1 0 0 1 0 0 0 0
1	0	1	1	0 0 1 0 0 0 0
0	1	0	1	0 1 0 0 0 0
1	0	1	0	1 0 0 0 0
0	1	1	0	0 0 0 0
1	1	0	0	0 0 0
1	0	1	1	0 0
0	1	0	1	0
1	0	1	0	

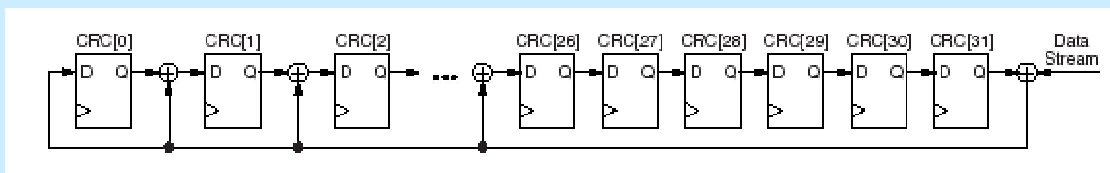
Message sent:

1 0 1 1 0 0 1 1 0 1 0

Example: Ethernet CRC-32

7	Application (HTTP,FTP, DNS)
4	Transport (TCP, UDP)
3	Network (IP)
2	Data Link (Ethernet, 802.11b)
1	Physical





CRC – basic principle

- A CRC of length N will detect any single error burst not longer than N bits and will detect a single error in 2^N bits
- Thus, appending just 32 CRC bits (4 bytes) to the transmitted data and checking to see that the CRC matches at the receiver will ensure less than 1 error in 2^{32} bits – but only a single error can be detected this way
- For low error rates, this is great
- Even for higher error rates CRC is good but not perfect
- Good data integrity check for files

Example – Cryptology

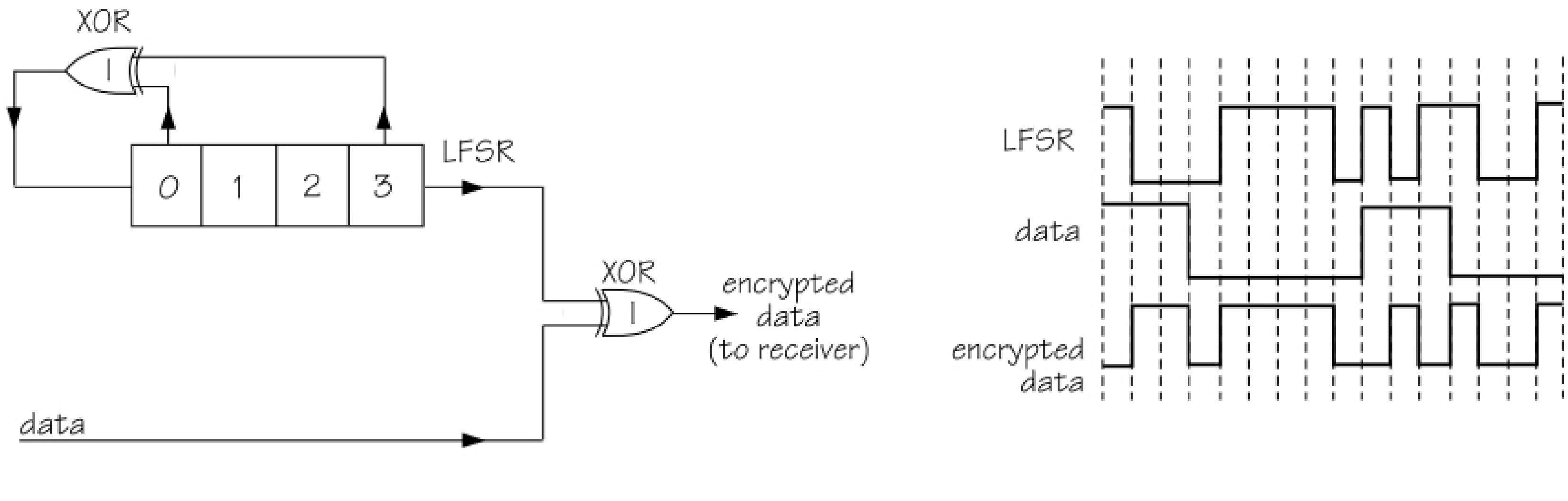
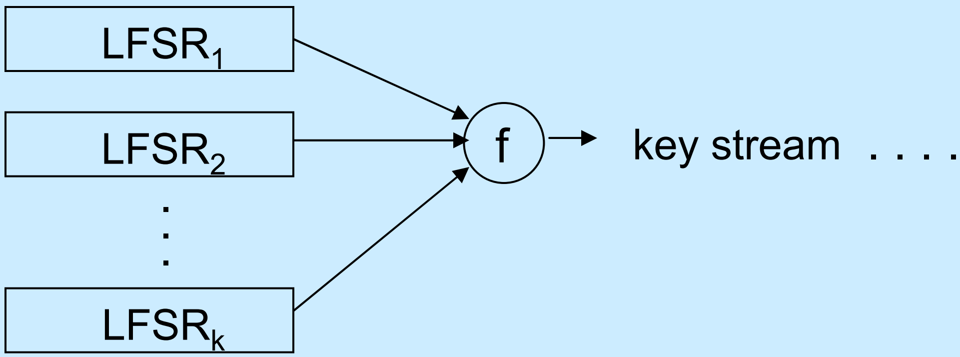


Figure F-11. Data encryption using an LFSR

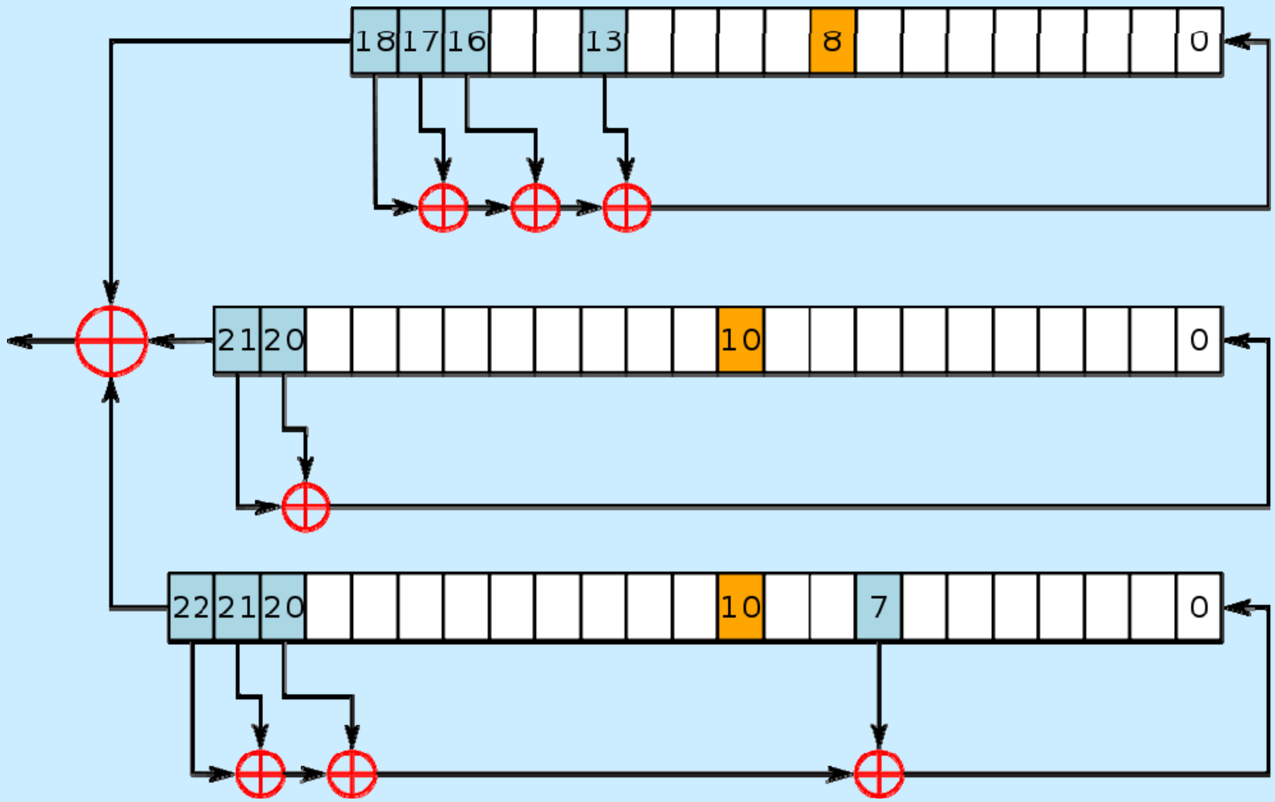
Stream Ciphers from LFSRs



Desirable properties of f:

- high non-linearity
- long "cycle period" ($\sim 2^{n1+n2+\ldots+nk}$)
- low correlation with the input bits

GSM A5/1

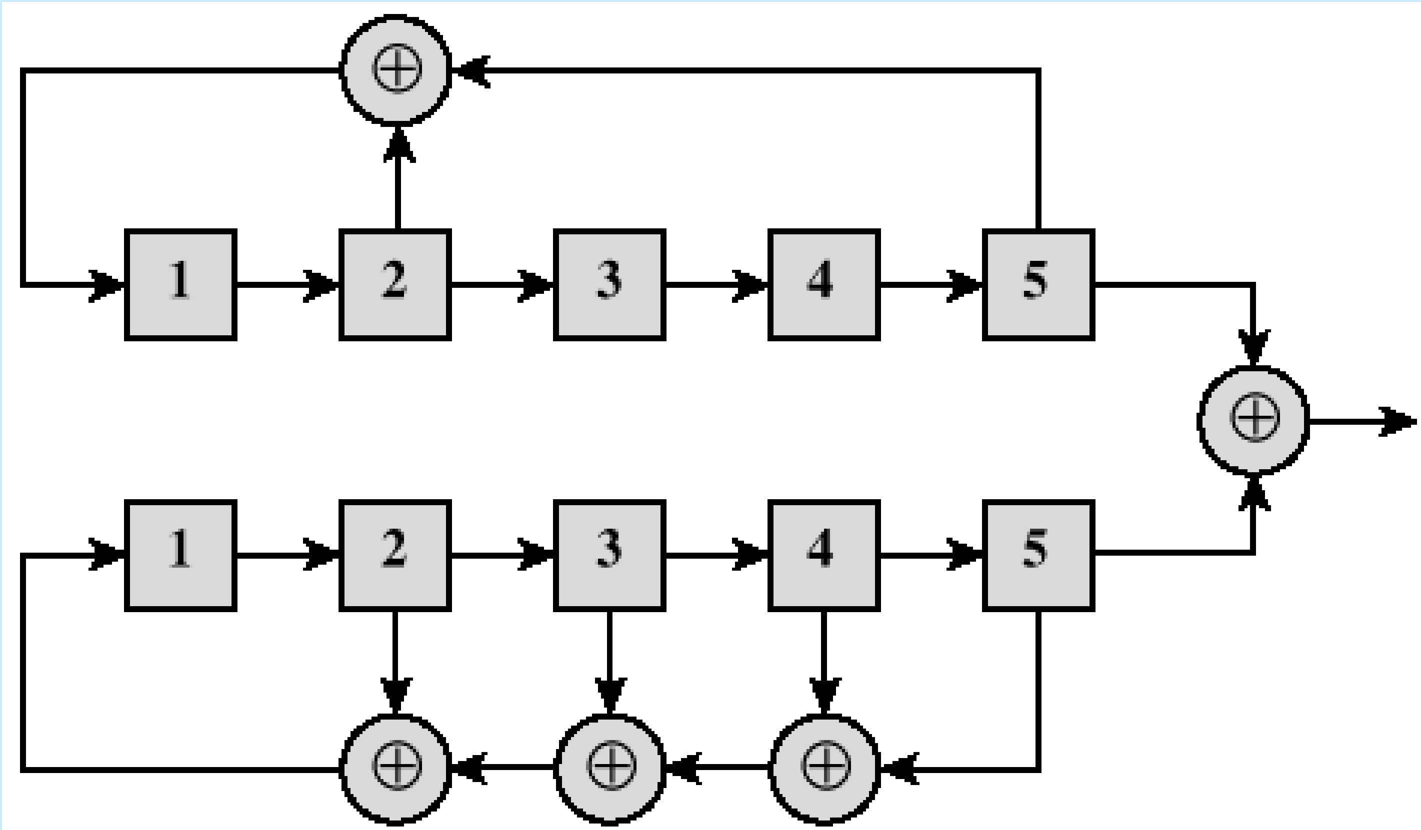


- The A5/1 stream cipher uses three LFSRs.

Gold codes

- Gold sequences constructed by the XOR of two PN sequences with the same clocking
- Codes have well-defined cross correlation properties
- Only simple circuitry needed to generate large number of unique codes
- This type of code is used in deep-space communications
- In following example two shift registers generate the two m-sequences and these are then bitwise XORed

Gold Codes – used in deep-space communications



(a) Shift-register implementation

Example – Cryptology and Spread Spectrum Communications

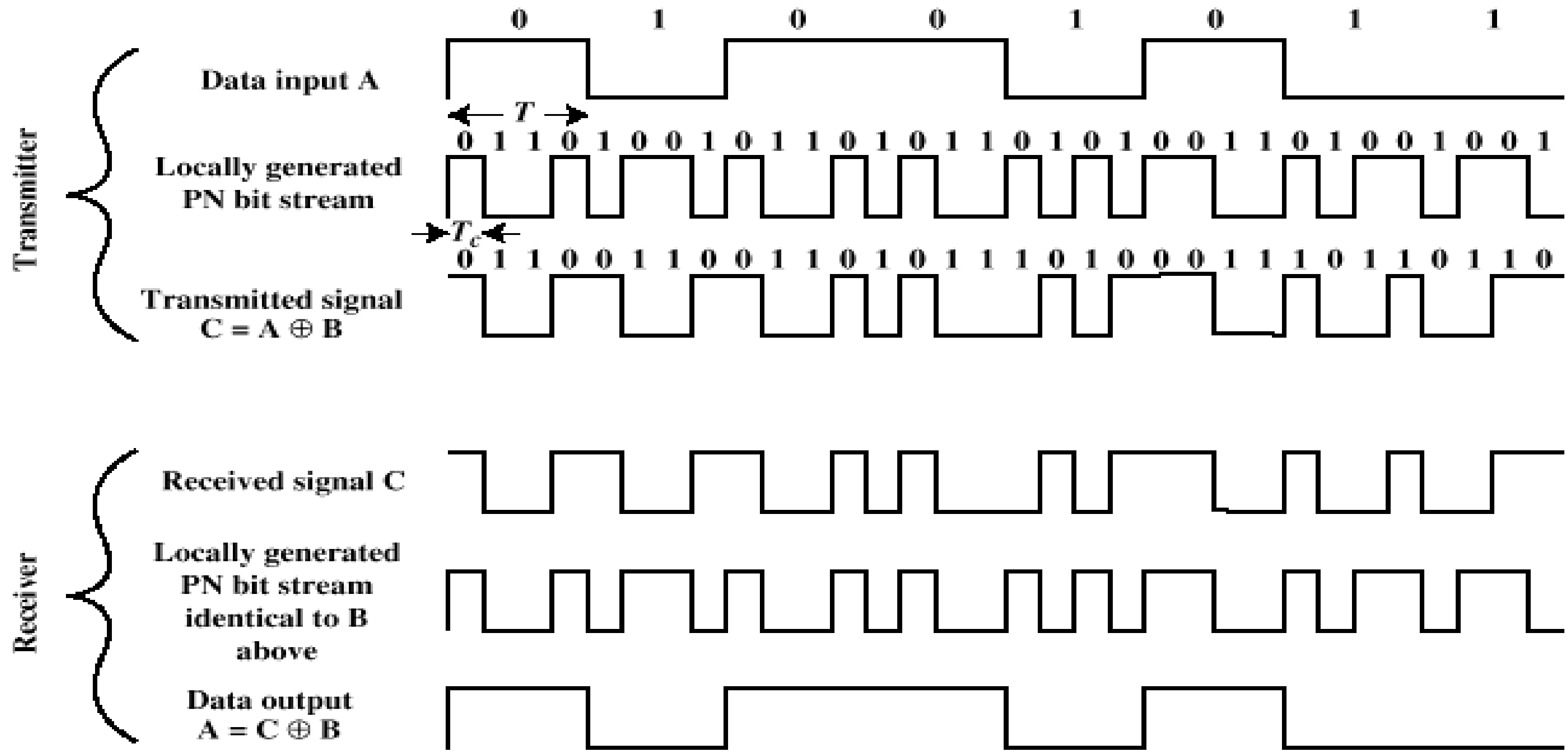


Figure 7.6 Example of Direct Sequence Spread Spectrum

Example – Spread Spectrum and Secure Communications

- If we modulate the signal, transmit it, receive it, and then “XOR” it with the same PN sequence, we have a secure wireless communications system!

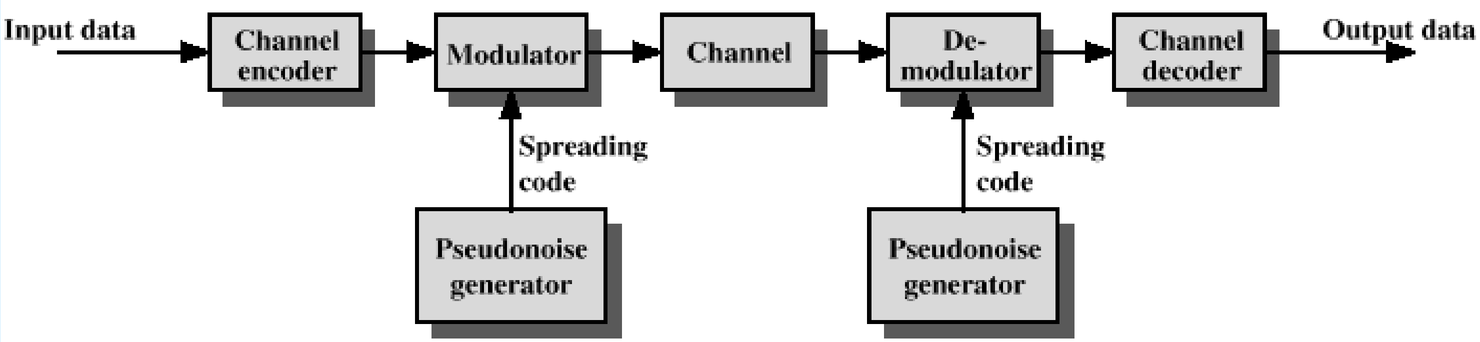


Figure 7.1 General Model of Spread Spectrum Digital Communication System

Spread Spectrum

- Input is fed into a channel encoder
  - Produces analog signal with narrow bandwidth
- Signal is further modulated using sequence of digits
  - Spreading code or spreading sequence
  - Generated by pseudonoise, or pseudo-random number generator
- Effect of modulation is to increase bandwidth of signal to be transmitted

Direct Sequence Spread Spectrum (DSSS)

- Each bit in original signal is represented by multiple bits in the transmitted signal
- Spreading code spreads signal across a wider frequency band
  - Spread is in direct proportion to number of bits used
- One technique combines digital information stream with the spreading code bit stream using XOR

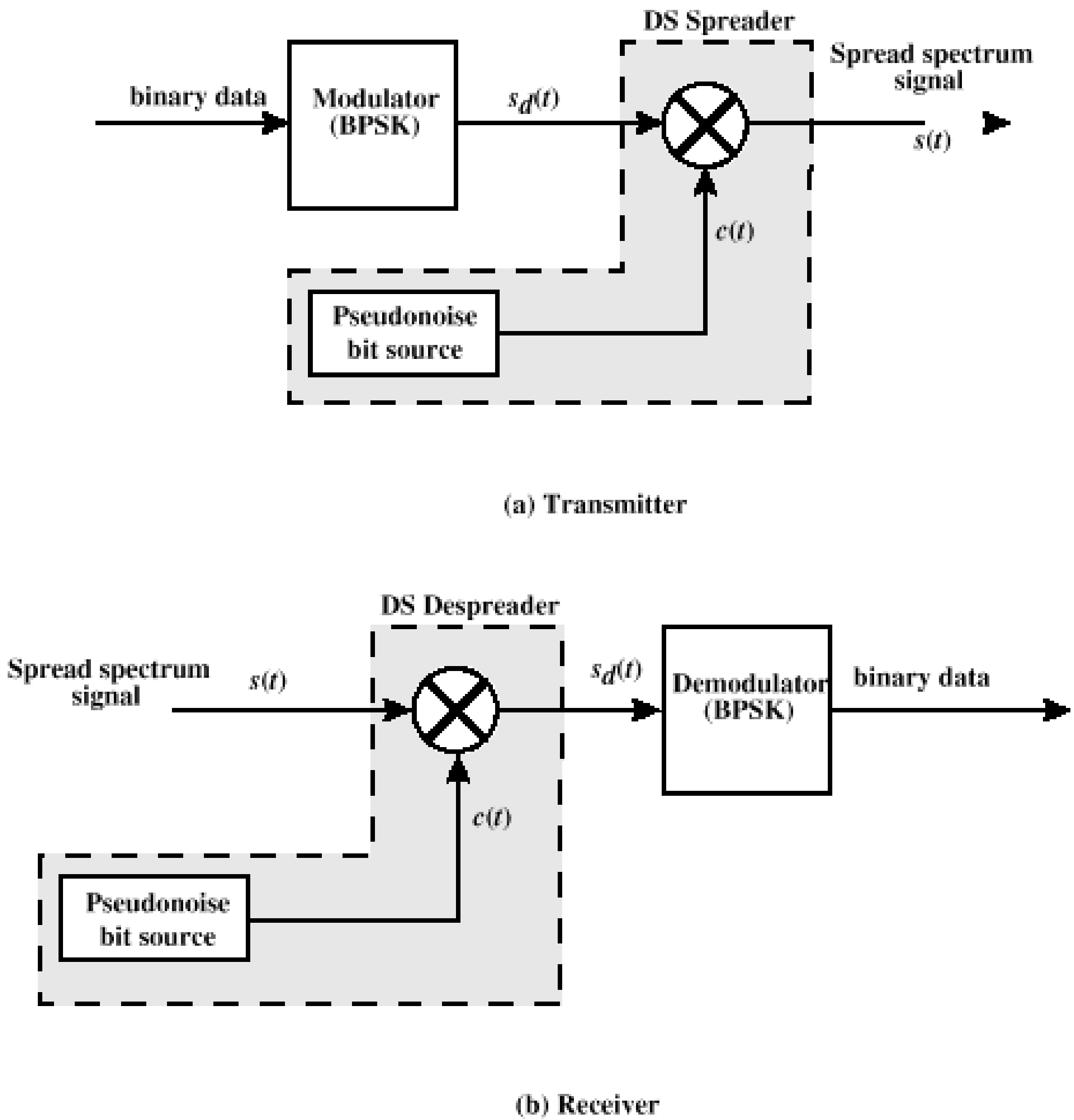


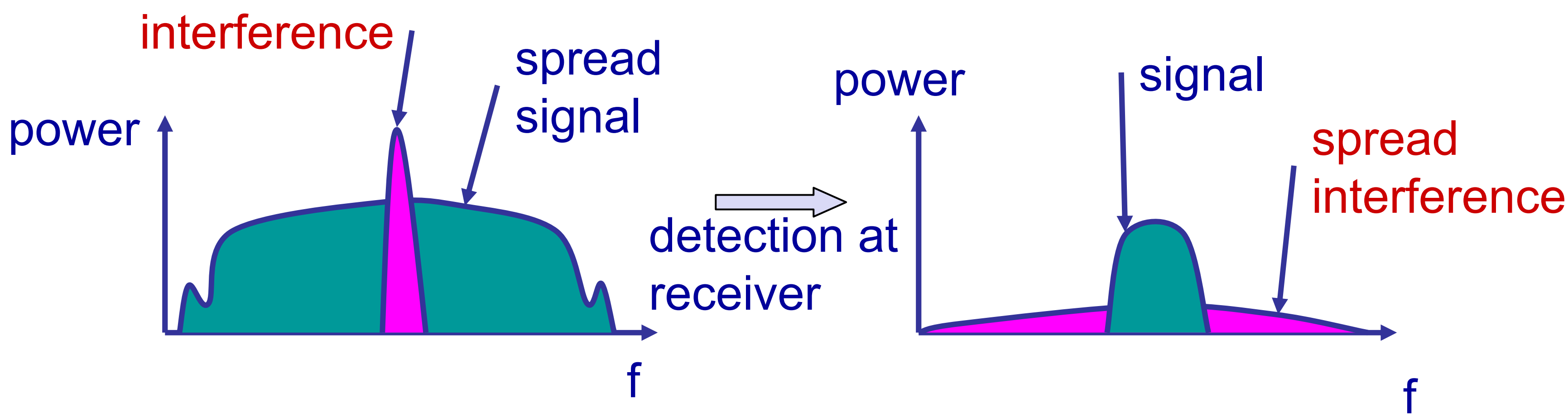
Figure 7.7 Direct Sequence Spread Spectrum System

Spread Spectrum

- What can be gained from apparent waste of spectrum?
  - Immunity from various kinds of noise and multipath distortion
  - Can be used for hiding and encrypting signals
  - Several users can independently use the same higher bandwidth with very little interference
- Used in GPS, radar, military communications

Spread Spectrum Technology

- Problem of radio transmission: frequency dependent fading can wipe out narrow band signals for duration of the interference
- Solution: spread the narrow band signal into a broad band signal using a special code

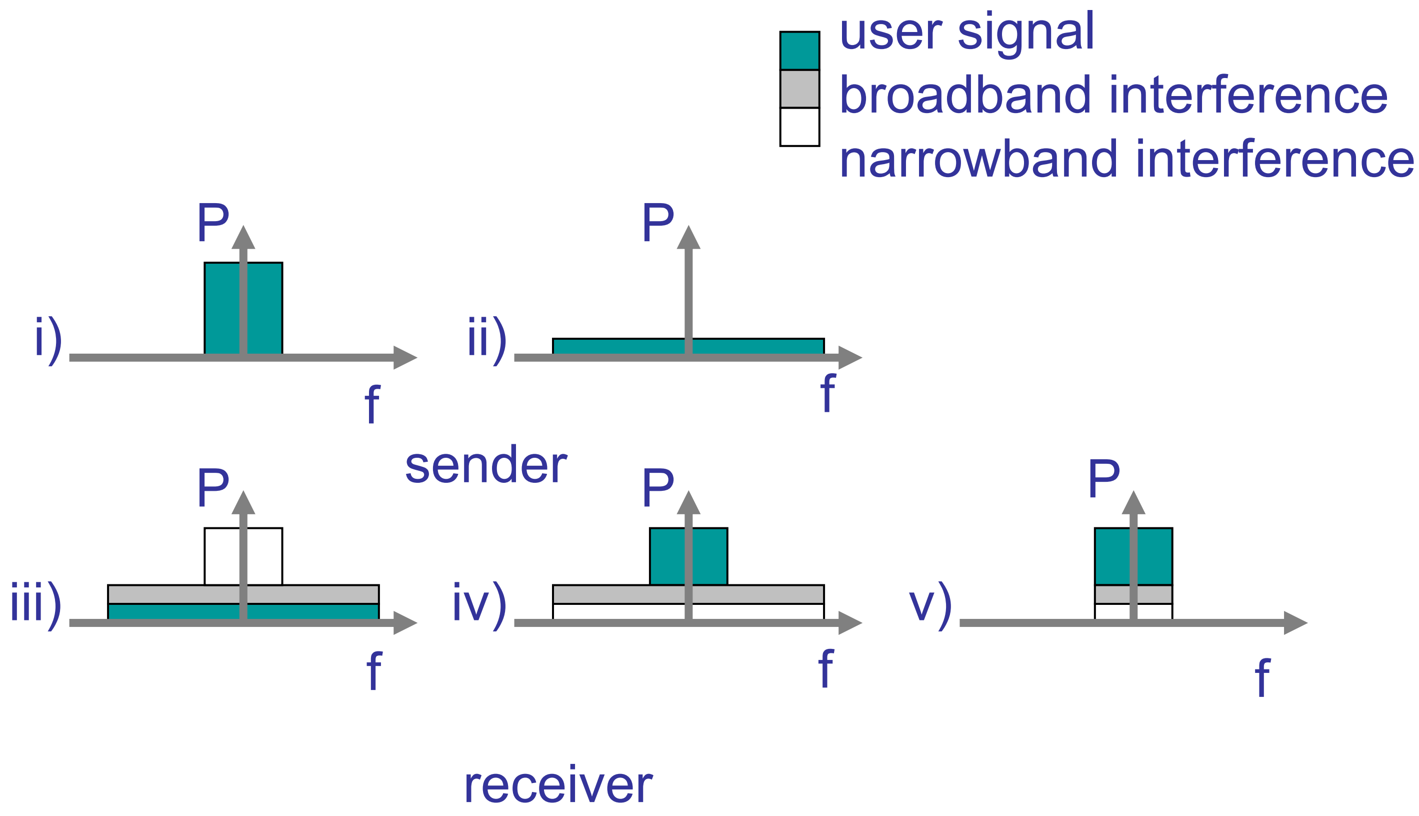


Spread Spectrum Technology

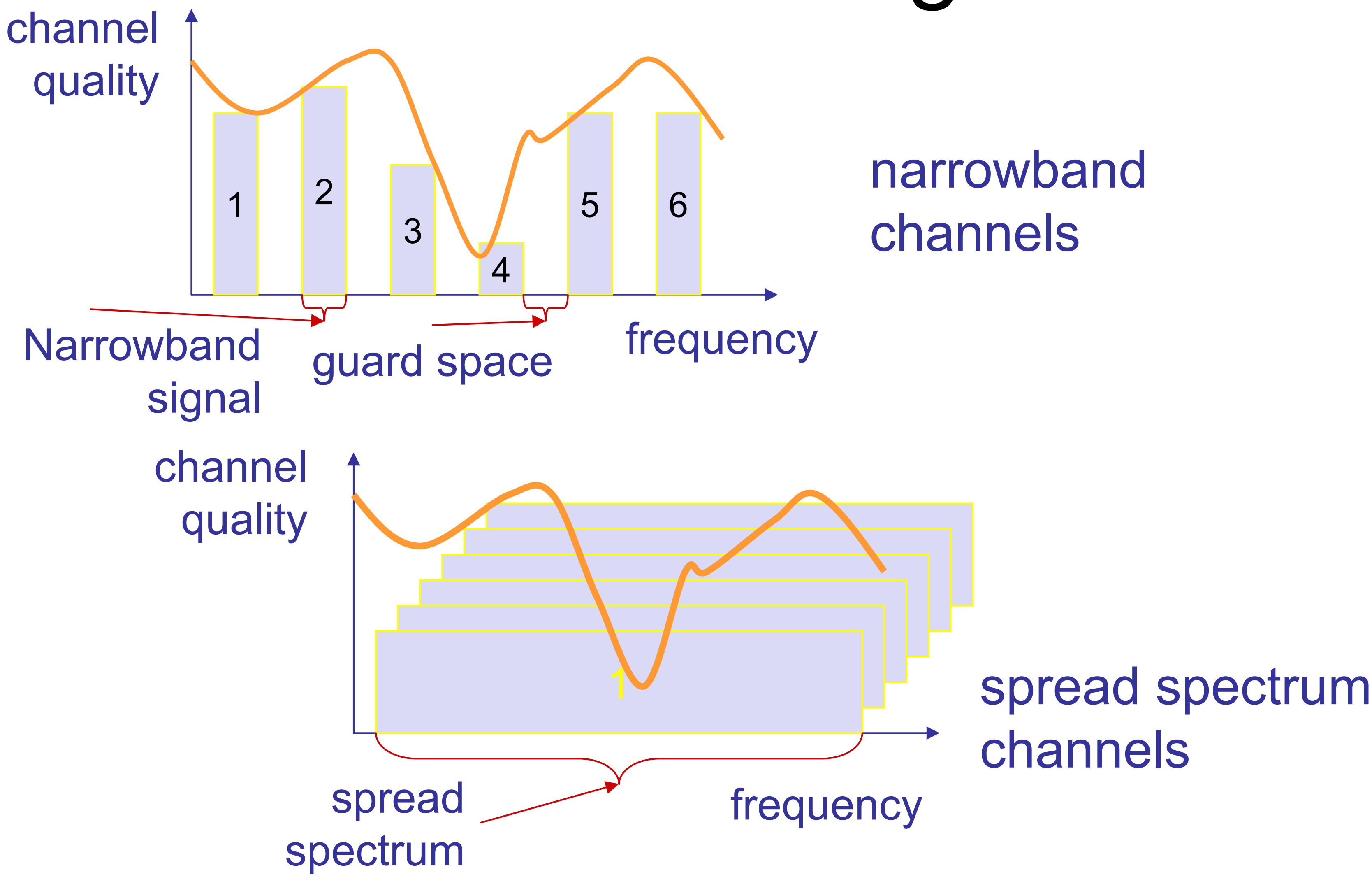
- Side effects:
  - coexistence of several signals without dynamic coordination
  - tap-proof
- Alternatives: Direct Sequence (DS/SS), Frequency Hopping (FH/SS)
- Spread spectrum increases BW of message signal by a factor *N*, Processing Gain

$$\text{Processing Gain } N = \frac{B_{ss}}{B} = 10\log_{10}\left(\frac{B_{ss}}{B}\right)$$

Effects of spreading and interference



Spreading and frequency selective fading



Frequency Hoping Spread Spectrum (FHSS)

- Signal is broadcast over seemingly random series of radio frequencies
  - A number of channels allocated for the FH signal
  - Width of each channel corresponds to bandwidth of input signal
- Signal hops from frequency to frequency at fixed intervals
  - Transmitter operates in one channel at a time
  - Bits are transmitted using some encoding scheme
  - At each successive interval, a new carrier frequency is selected
- Used in Bluetooth, Wifi, Celular

Frequency Hoping Spread Spectrum

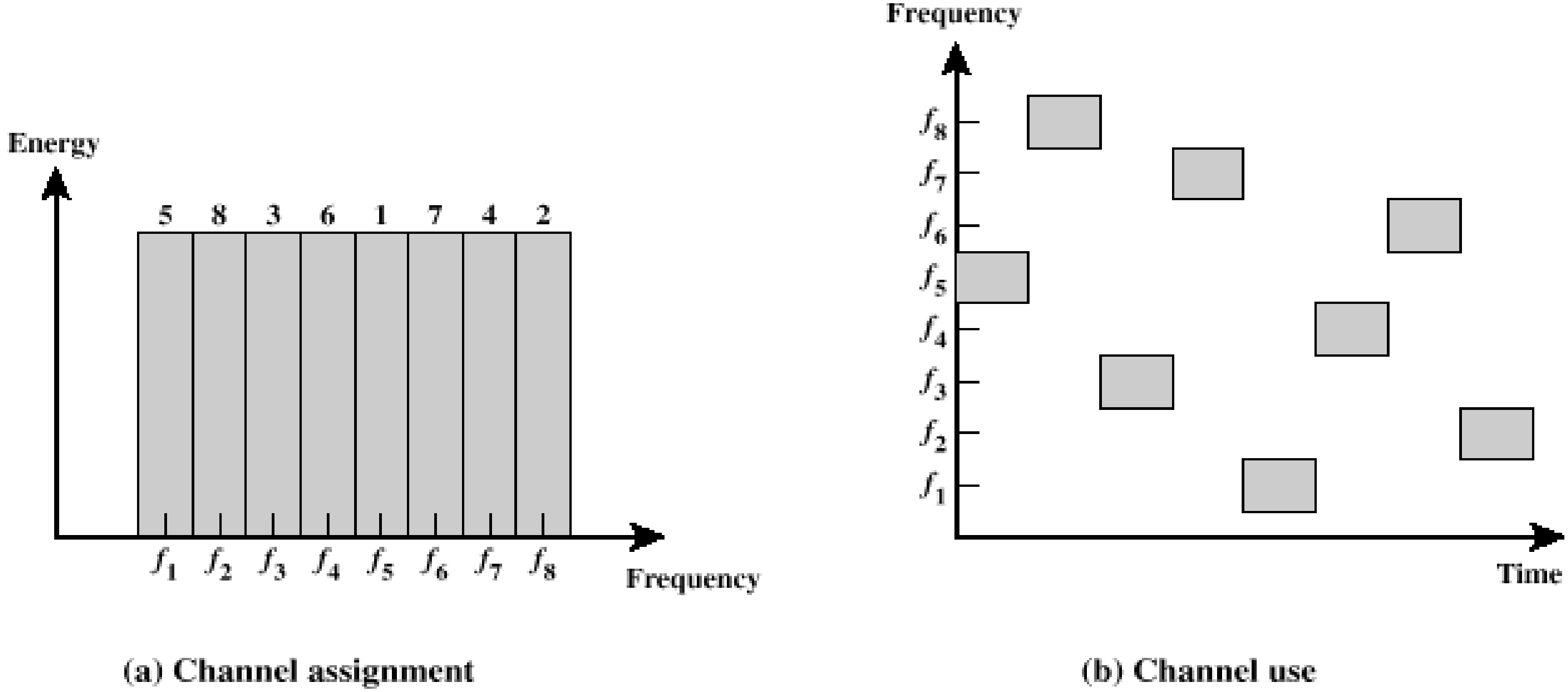


Figure 7.2 Frequency Hopping Example

Frequency Hoping Spread Spectrum

- Channel sequence dictated by spreading code
- Receiver, hopping between frequencies in synchronization with transmitter, picks up message
- Advantages
  - Eavesdroppers hear only unintelligible blips
  - Attempts to jam signal on one frequency succeed only at knocking out a few bits
  - Good to avoid interference by other transmitters – e.g. phones in other mobile network cells

FHSS Performance Considerations

- Large number of frequencies used
- Results in a system that is quite resistant to jamming
  - Jammer must jam all frequencies
  - With fixed power, this reduces the jamming power in any one frequency band
  - Obvious military applications

Properties of M-Sequences

- Property 1:
  - Has 2^{n-1} ones and 2^{n-1}-1 zeros
- Property 2:
  - For a window of length n slid 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

Properties of M-Sequences

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

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

PN Autocorrelation Function

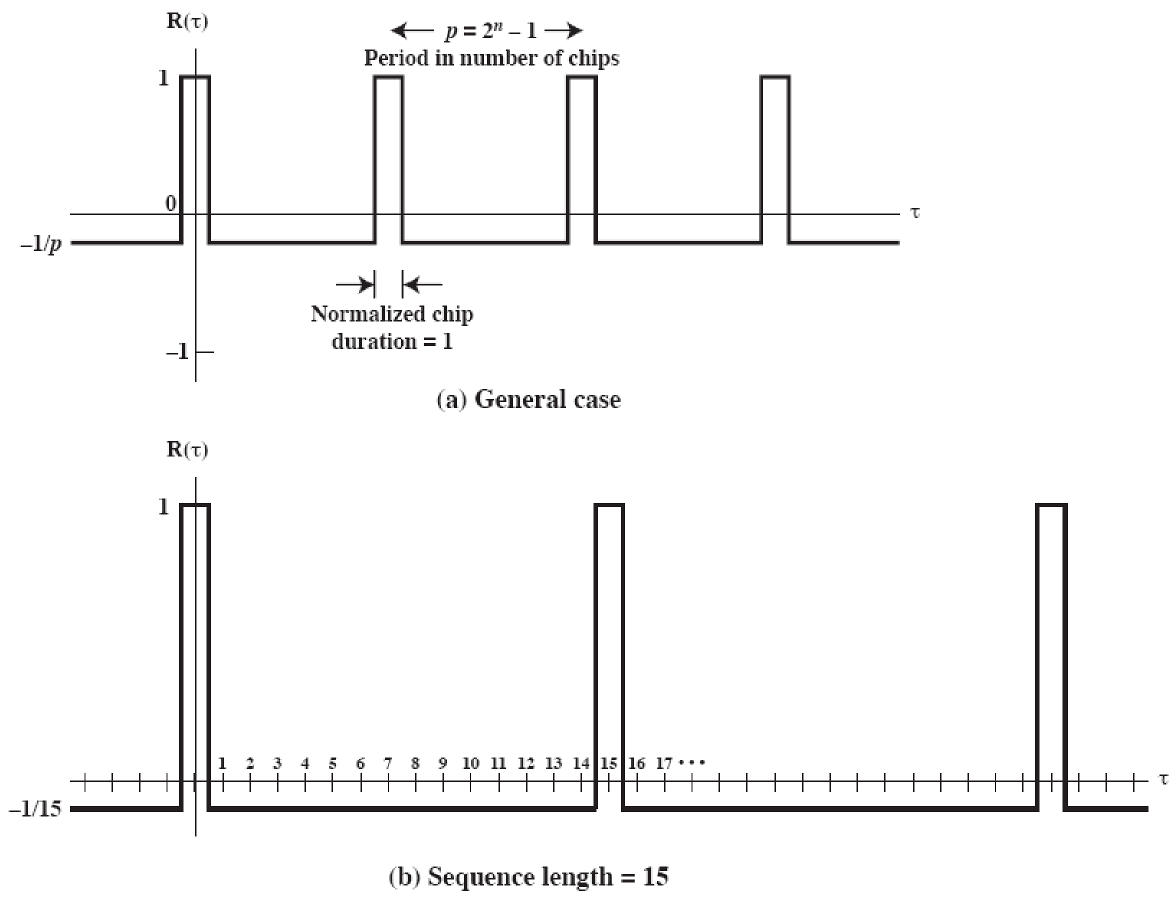
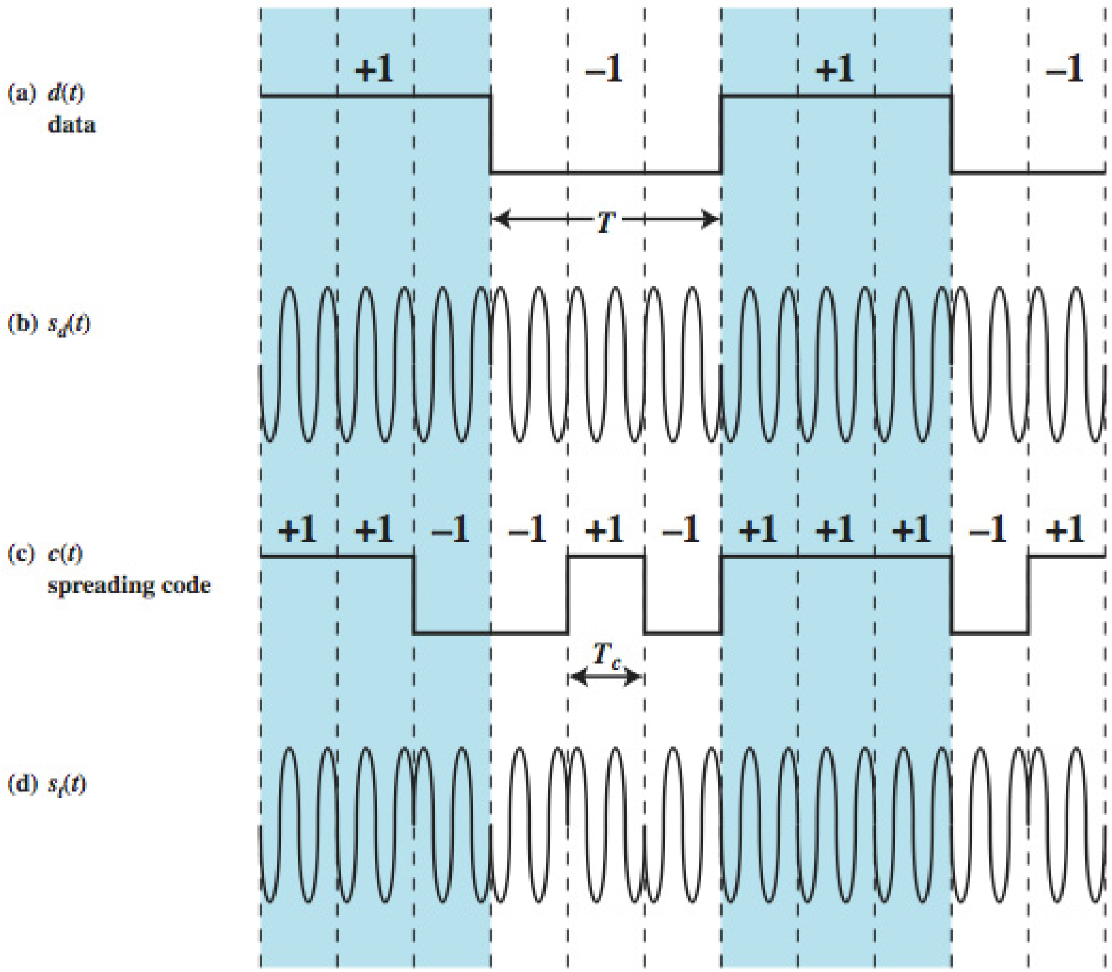


Figure 7.15 PN Autocorrelation Function

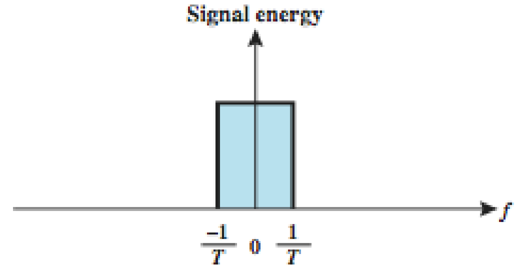
Definitions

- Correlation
  - The concept of determining how much similarity one set of data has with another
  - 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

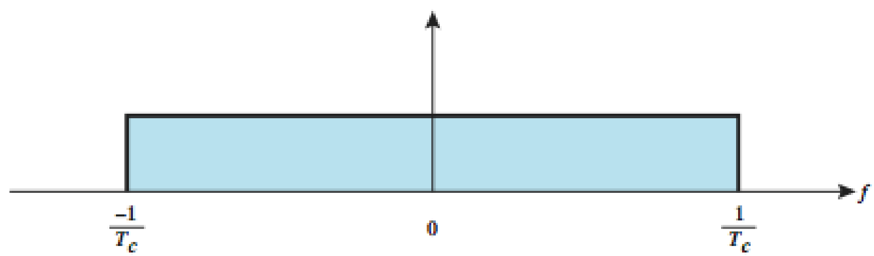
DSSS Example Using BPSK



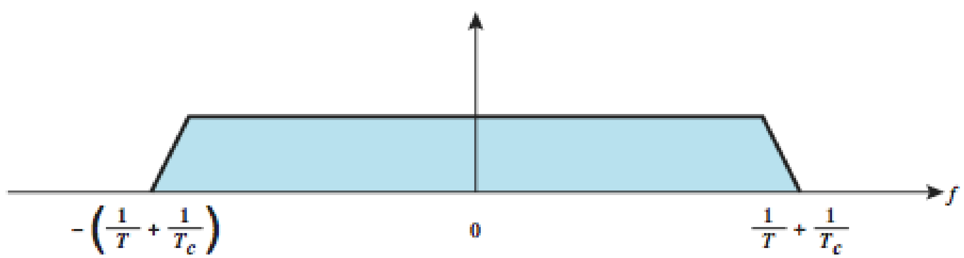
Approximate Spectrum of DSSS Signal



(a) Spectrum of data signal



(b) Spectrum of pseudonoise signal



(c) Spectrum of combined signal

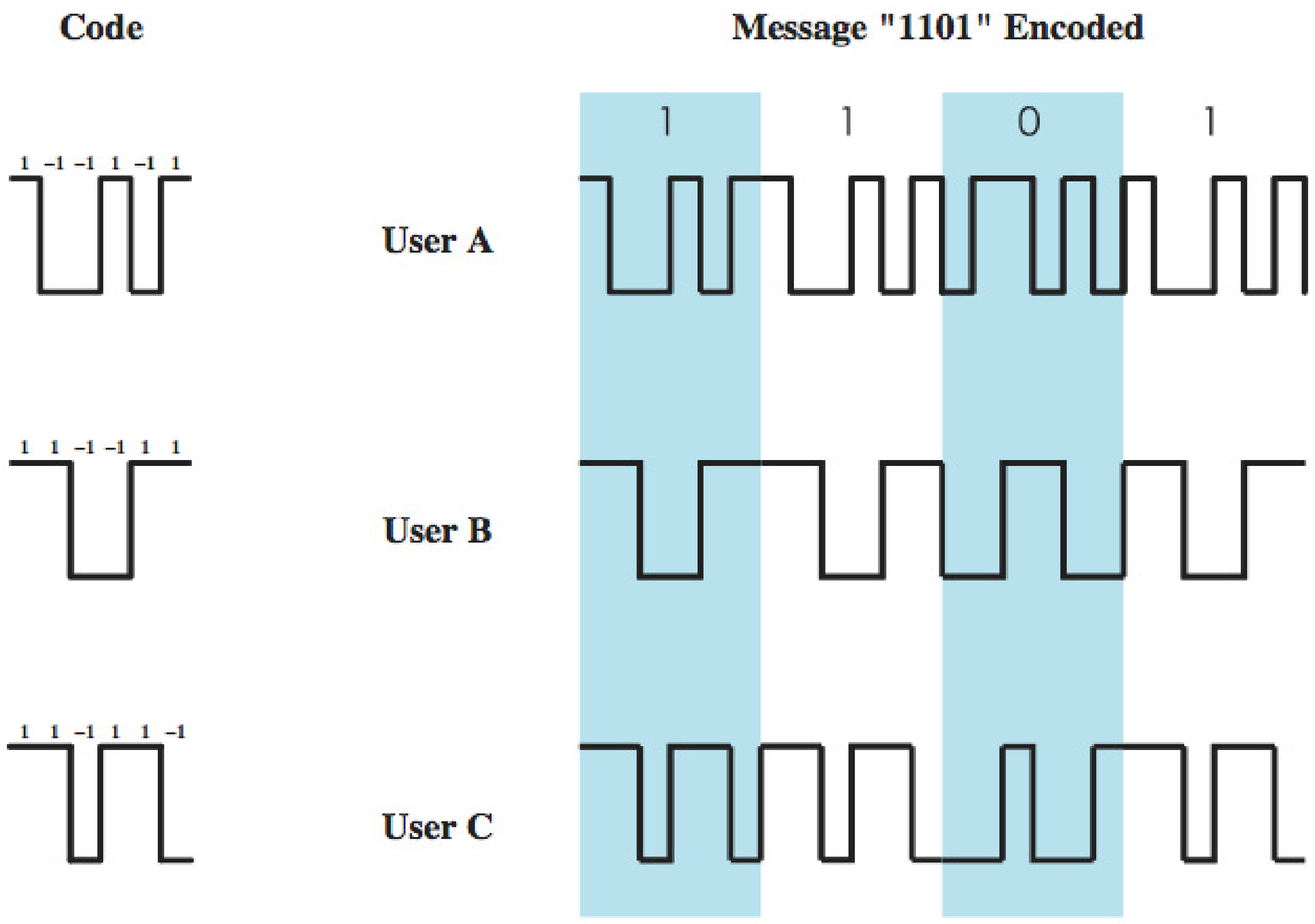
Code-Division Multiple Access (CDMA)

- Basic Principles of CDMA
  - D = rate of data signal
  - Break each bit into k *chips*
    - Chips are a user-specific fixed pattern
  - Chip data rate of new channel = kD
- Used in a lot of cellular phone standards

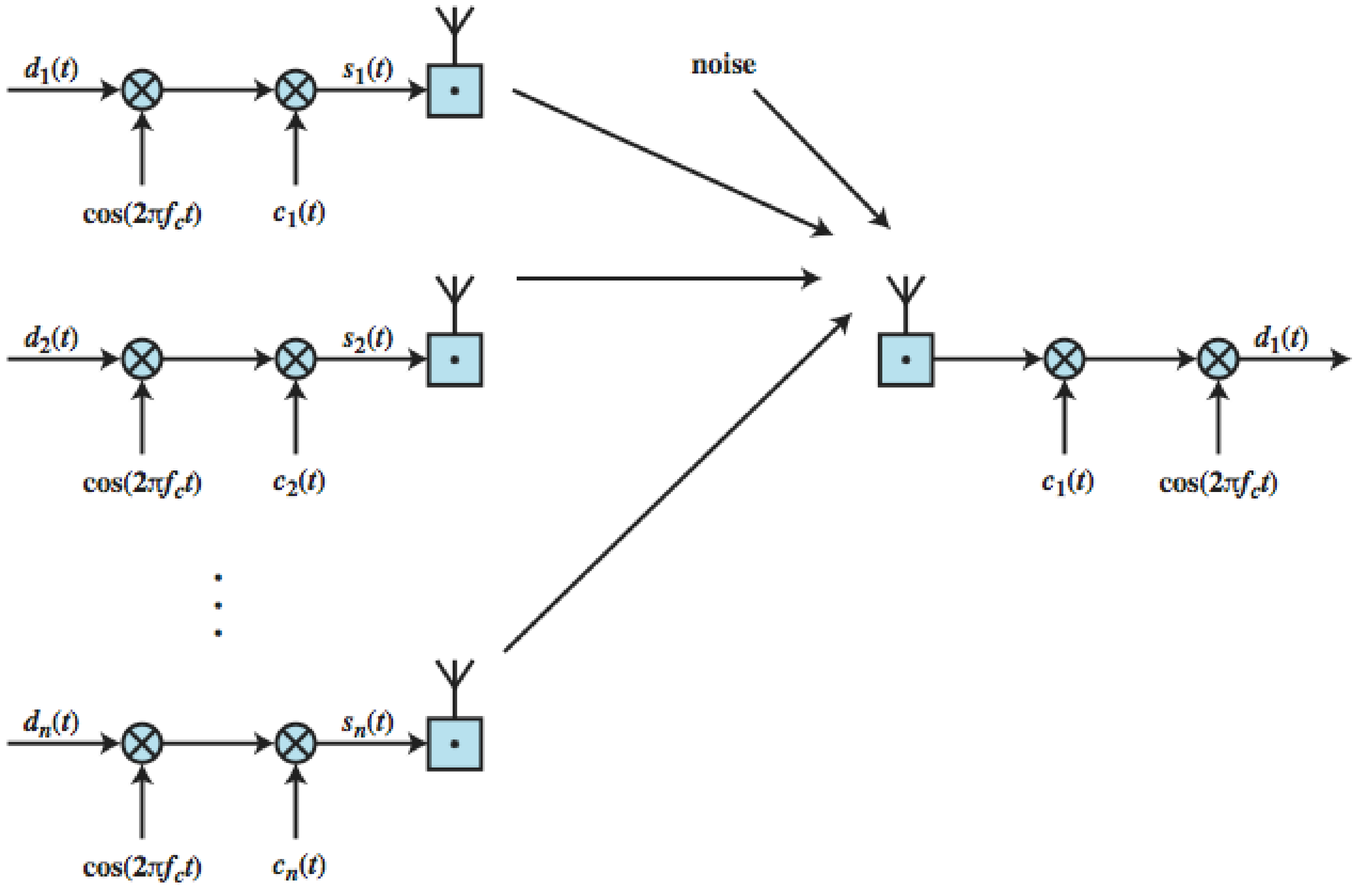
Code Division Multiple Access (CDMA)

- A multiplexing technique used with spread spectrum
- Given a data signal rate D
- Break each bit into k chips according to a fixed chipping code specific to each user
- Resulting new channel has chip data rate kD chips per second
- Can have multiple channels superimposed

CDMA Example



CDMA for DSSS



Advantages of Cross Correlation

- The cross correlation between an m-sequence and noise is low
  - This property is useful to the receiver in filtering out noise
- The cross correlation between two different m-sequences is low
  - This property is useful for CDMA applications
  - Enables a receiver to discriminate among spread spectrum signals generated by different m-sequences

Categories of Spreading Sequences

- Spreading Sequence Categories
  - PN sequences
  - Orthogonal codes
- For FHSS systems
  - PN sequences most common
- For DSSS systems not employing CDMA
  - PN sequences most common
- For DSSS CDMA systems
  - PN sequences
  - Orthogonal codes (which we didn't discuss)

Why use a counter when you can use an LFSR!

Logic structure of LFSR is the same as a binary counter
but the next state logic is much simpler => faster clock possible

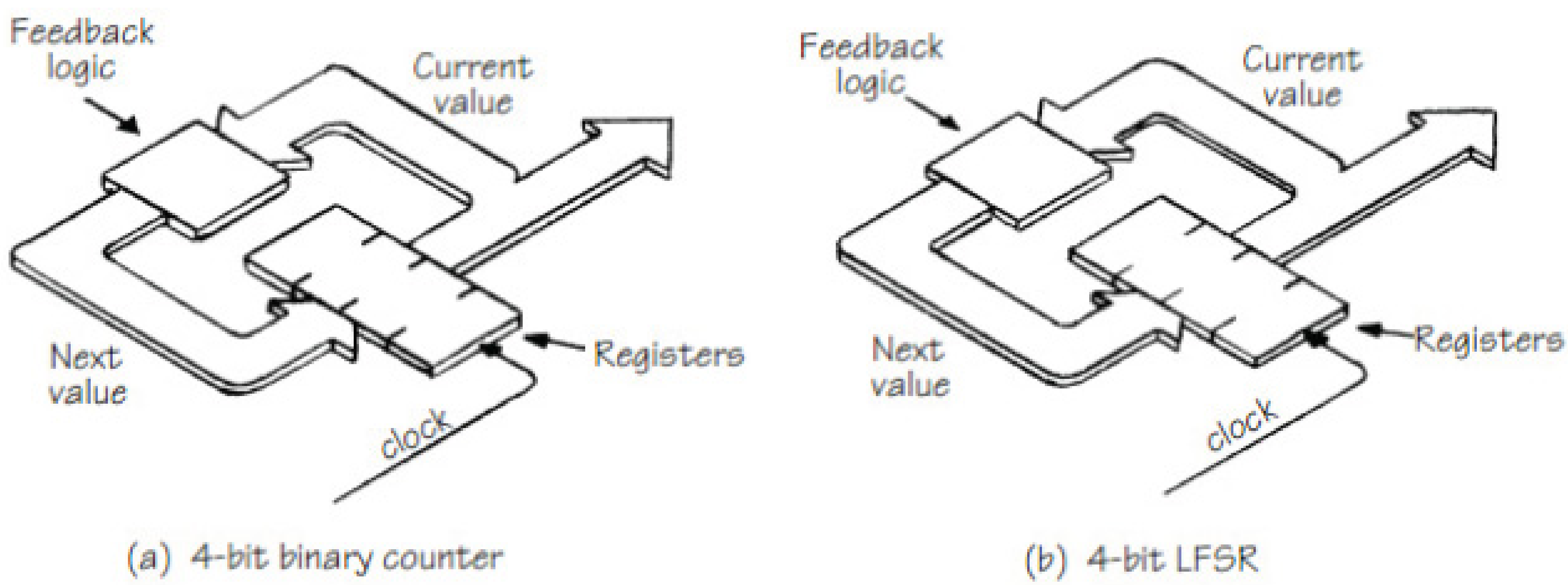
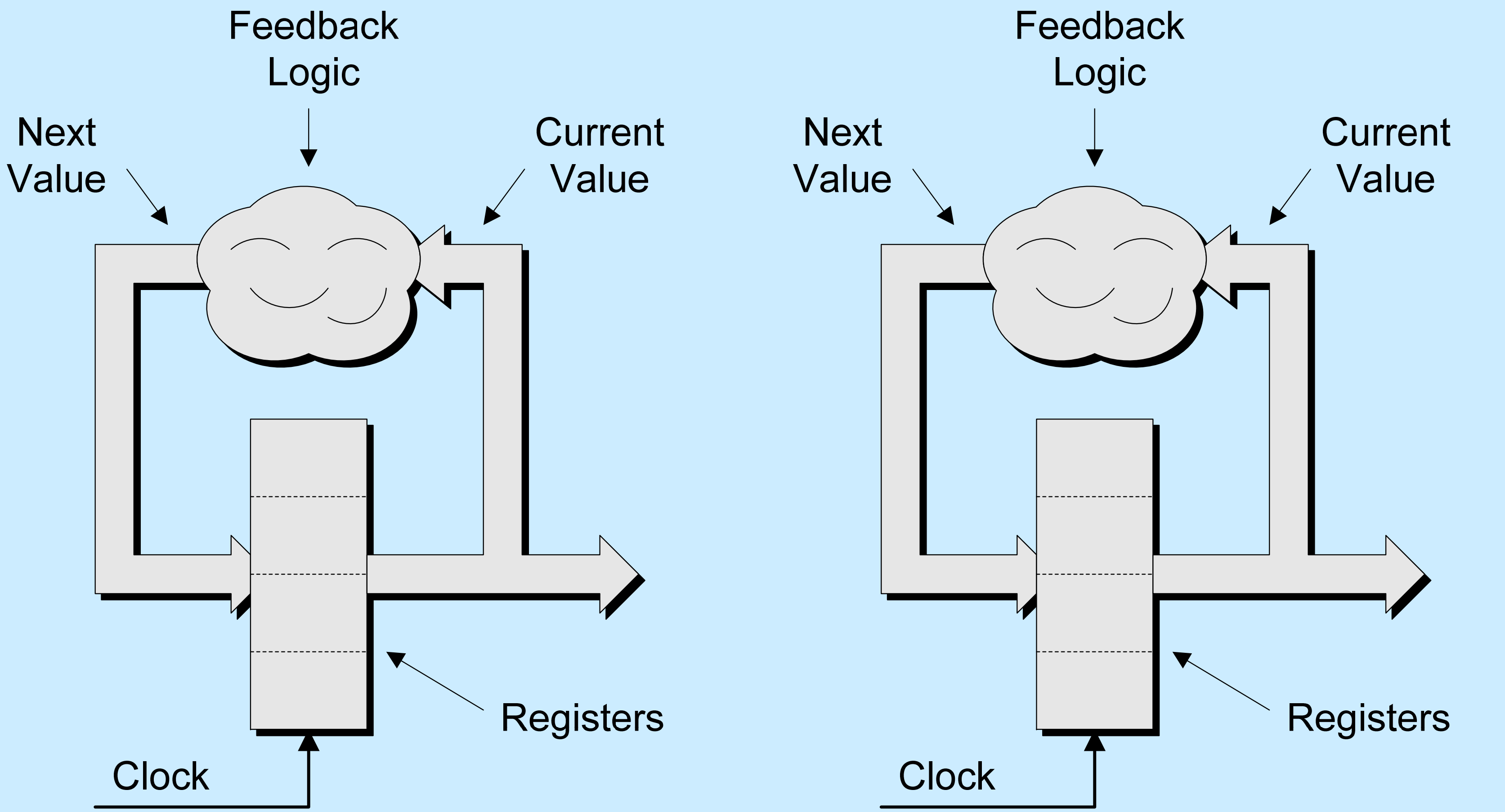


Figure F-8. Binary counter versus LFSR

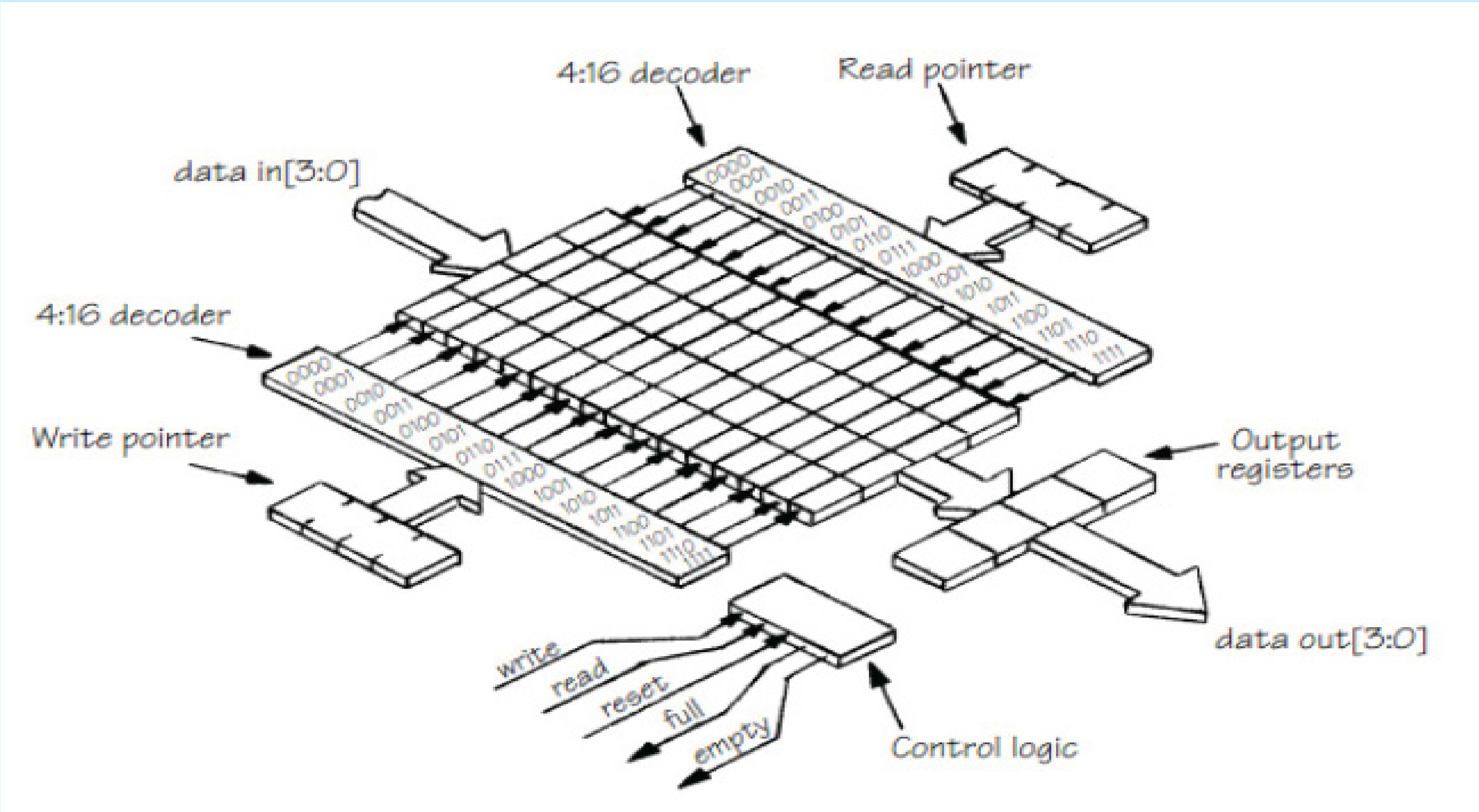


(a) 4-bit binary counter

(b) 4-bit LFSR

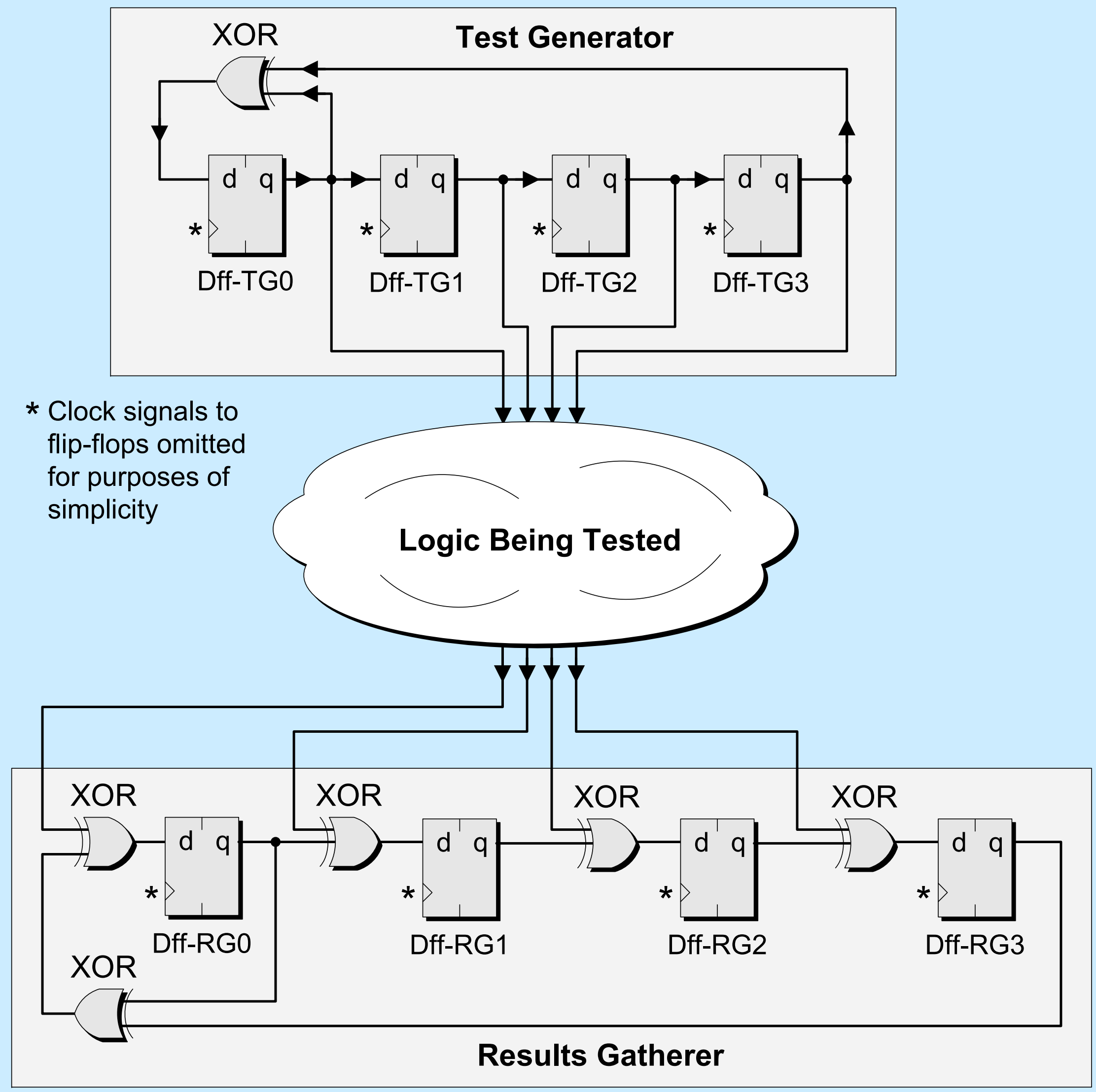
Figure C-08

Usage as FIFO pointers



Built-In Self Test (BIST)

- Devices can be self-tested (at speed) by incorporating LFSRs circuits into the design. Testing can occur while the device is operating or while in an idle mode.
- An LFSR generates a Pseudo-Random Test Pattern. A small LFSR with the appropriate feedback can generate very long sequences of apparently random data.



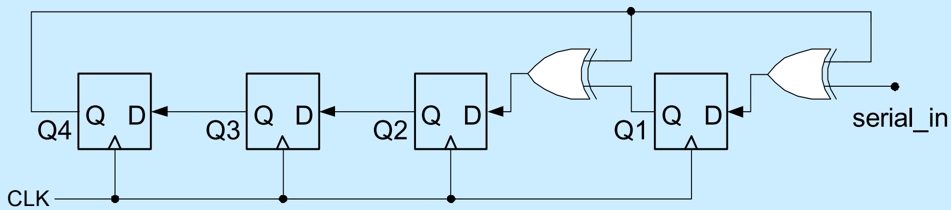
Test Generator
XOR
d q
Dff-TG0
Dff-TG1
Dff-TG2
Dff-TG3
* Clock signals to flip-flops omitted for purposes of simplicity
Logic Being Tested
XOR
Dff-RG0
Dff-RG1
Dff-RG2
Dff-RG3
Results Gatherer

Figure C-13

Conclusion

- LFSRs are easy to construct and require very few hardware resources
- It is easy to generate pseudo-random sequences and psuedo random numbers with LFSRs
- There are many uses in LFSRs in a wide variety of applications including chip design and testing, networking, cryptology, and communications

CRC decoding



1 0 1 1 0 0 1 1 0 1 0

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