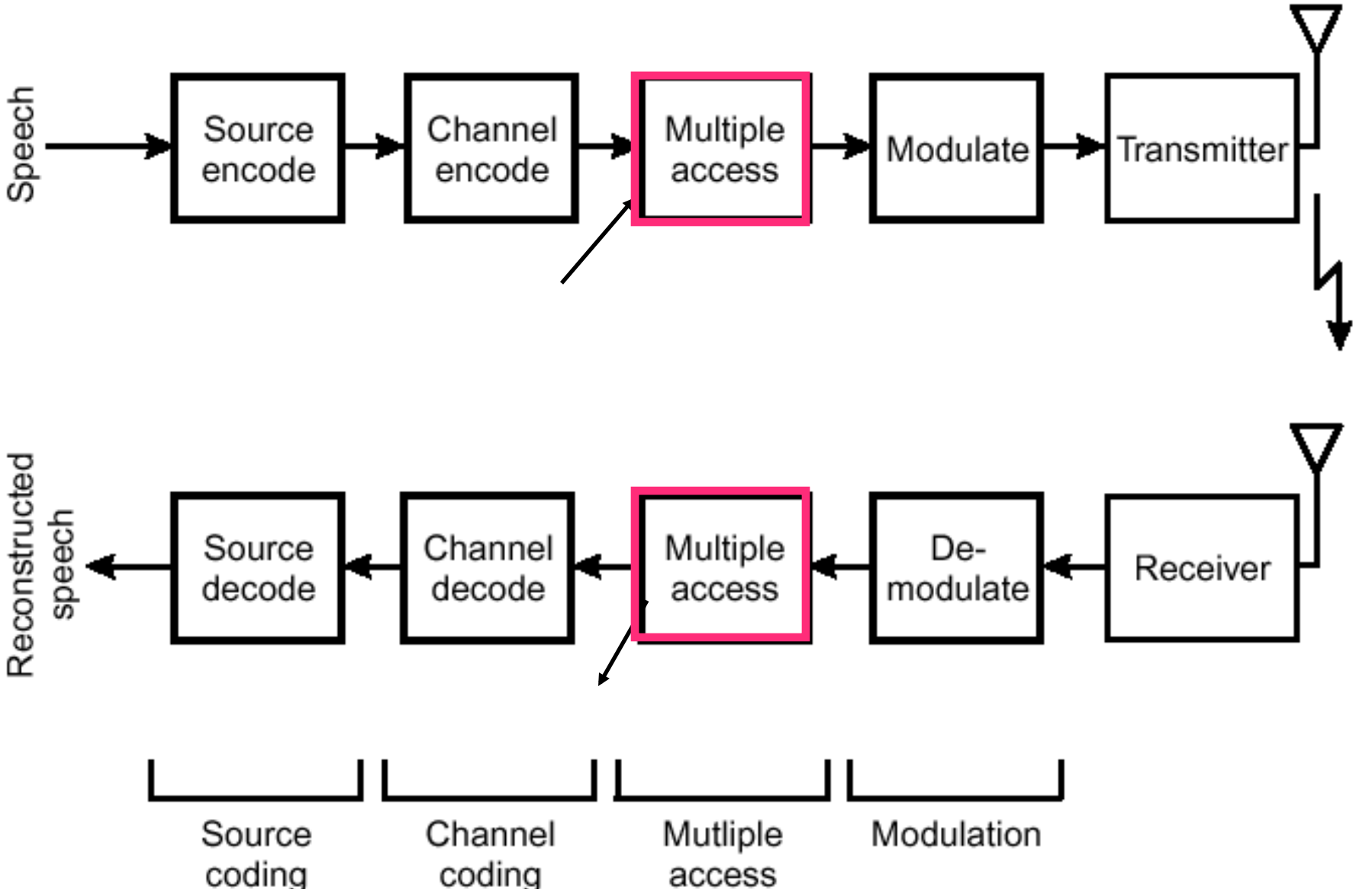
EENG473 Mobile Communications
Module 4 : Week # (14)

Spread spectrum and CDMA II



By: Dr.Mohab Mangoud

Codes for CDMA

Codes are classified into two families:

- **Orthogonal Codes.**
- **Pseudorandom Noise (PN) Codes.**

Kinds of Orthogonal Codes:

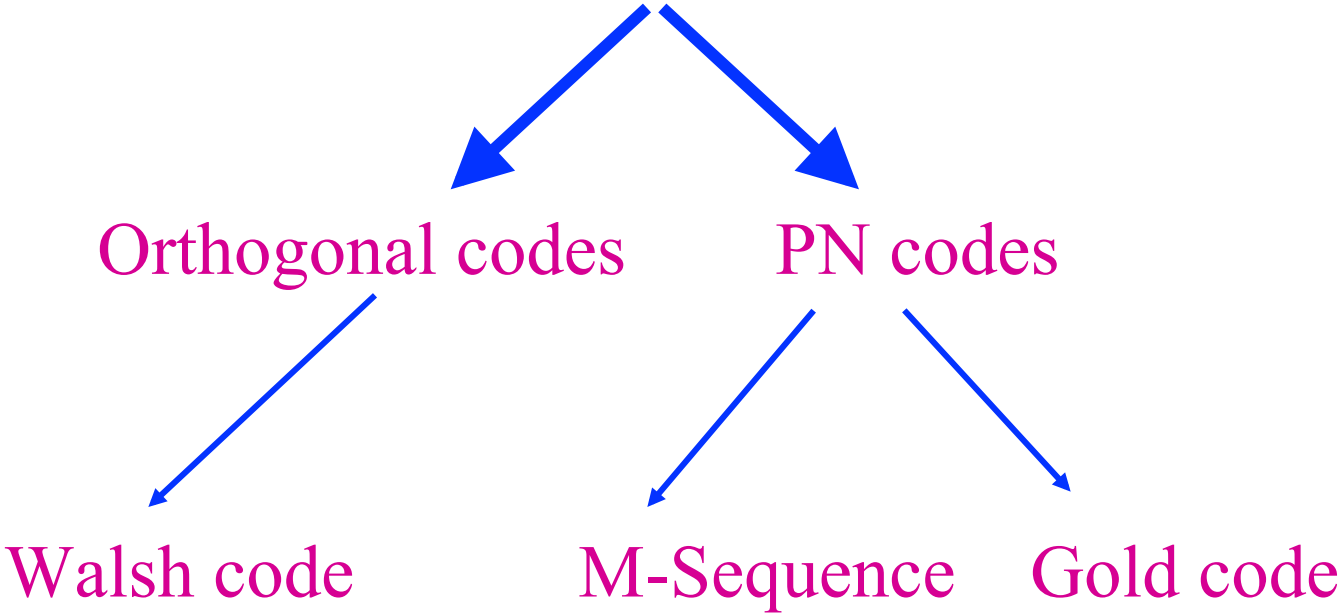
1-) Walsh Codes
2-) Orthogonal Gold Codes
3-) Multi-rate Orthogonal Gold Codes

Kinds of Pseudorandom Noise (PN) Codes:

1-) Maximal Length Sequences
2-) Gold Codes
3-) Kasami Sequences
4-) Barker Codes.

By: Dr.Mohab Mangoud

Codes in CDMA



By: Dr.Mohab Mangoud

3.5.2 PN Codes

Although the forward link of IS-95 CDMA has pilot and sync channels to aid synchronization, the reverse link does not have pilot and sync channels. The mobile stations transmit at will, and no attempt is made to synchronize their transmissions. Thus Walsh codes cannot be used on the reverse link. The incoherent nature of the reverse link calls for the use of another class of codes, PN codes, for channelization.

3.5.2.1 Generation of PN Codes

PN code sets can be generated from linear feedback shift registers. One such example (a three-stage register) is shown in Figure 3.10. Binary bits are shifted through the different stages of the register. The output of the last stage and the output of one intermediate stage are combined and fed as input to the first stage. The register starts with an initial sequence of bits, or initial state, stored in its stages. Then the register is clocked, and bits are moved through the stages. This way, the register continues to generate output bits and feed input bits to its first stage.

The output bits of the last stage form the PN code. We now demonstrate the code generation using the register shown in Figure 3.10. An initial state of [1, 0, 1] is used for the register. The output of stage 3 is the output of the register. After clocking the bits through the register, we obtain the results summarized in Table 3.2.

Note that at shift 7, the state of the register returns to that of the initial state, and further shifting of the bits yields another identical sequence of

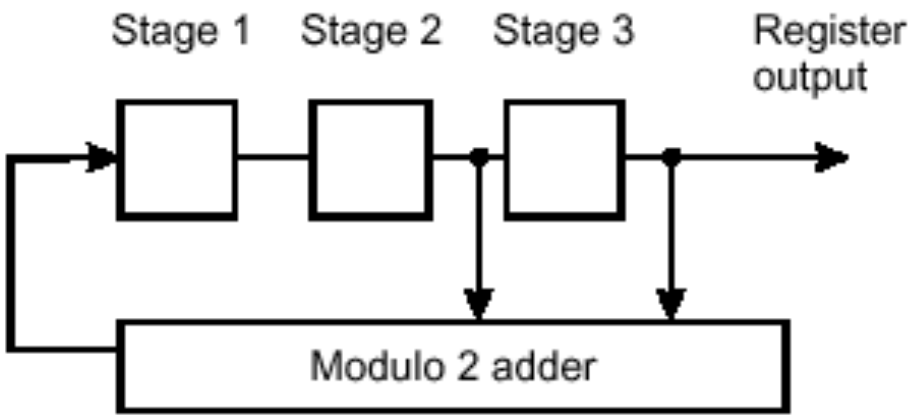


Figure 3.10 An example of linear feedback shift register for PN code generation.

outputs. Thus, the effective length of the periodic PN code generated is 7. The register output forms the PN code, which is

$$\mathbf{p} = [1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0]$$

The code generated in this manner is called a maximal-length shift register code, and the length L of a maximal-length code is

$$L = 2^N - 1 \tag{3.8}$$

where N is the number of stages, or order, of the register. In this case, $N = 3$, and the code length equals 7. The PN code structure is determined by the feedback logic (i.e., which stages are tapped for feedback) and the initial register

Table 3.2
Register States and Outputs

	Output	Output	Output	Output
Shift	Stage 1	Stage 2	Stage 3	Register
0	1	0	1	1
1	1	1	0	0
2	1	1	1	1
3	0	1	1	1
4	0	0	1	1
5	1	0	0	0
6	0	1	0	0
7	1	0	1	1

state. For example, if the initial state of the register is [0, 0, 0], then the different stages would get "stuck" in zeros; the register output then would be all zeros, and the code generated would not be maximal length.

A PN code set of seven codes can be generated by successively shifting $\mathbf{p}$, and by changing 0s to −1s we obtain

$$\mathbf{p}_0 = [+1 \quad -1 \quad +1 \quad +1 \quad +1 \quad -1 \quad -1]$$
$$\mathbf{p}_1 = [-1 \quad +1 \quad -1 \quad +1 \quad +1 \quad +1 \quad -1]$$
$$\mathbf{p}_2 = [-1 \quad -1 \quad +1 \quad -1 \quad +1 \quad +1 \quad +1]$$
$$\mathbf{p}_3 = [+1 \quad -1 \quad -1 \quad +1 \quad -1 \quad +1 \quad +1]$$
$$\mathbf{p}_4 = [+1 \quad +1 \quad -1 \quad -1 \quad +1 \quad -1 \quad +1]$$
$$\mathbf{p}_5 = [+1 \quad +1 \quad +1 \quad -1 \quad -1 \quad +1 \quad -1]$$
$$\mathbf{p}_6 = [-1 \quad +1 \quad +1 \quad +1 \quad -1 \quad -1 \quad +1]$$

Readers can easily verify that these codes satisfy the conditions outlined in Section 1.2 for DS-SS multiple access; that is,

1. The cross-correlation should be zero or very small.
2. Each sequence in the set has an equal number of 1s and −1s, or the number of 1s differs from the number of −1s by at most one.
3. The scaled dot product of each code should equal to 1.

Since the maximal PN code length is always an odd number (see (3.8)) and the code shown above has four +1s and three −1s, the code satisfies condition 2.

Since the maximal PN code length is always an odd number (see (3.8)) and the code shown above has four +1s and three −1s, the code satisfies condition 2.

3.5.2.2 Channelization using PN Codes

We again use an example to illustrate how PN codes can be used for multiple access. Suppose the same three users wish to send three separate messages. These messages are

$$\mathbf{m}_1 = [+1 \quad -1 \quad +1]$$
$$\mathbf{m}_2 = [+1 \quad +1 \quad -1]$$
$$\mathbf{m}_3 = [-1 \quad +1 \quad +1]$$

Each of the three users is assigned a PN code, respectively:

$$\mathbf{P}_0 = [+1 \quad -1 \quad +1 \quad +1 \quad +1 \quad -1 \quad -1]$$
$$\mathbf{P}_3 = [+1 \quad -1 \quad -1 \quad +1 \quad -1 \quad +1 \quad +1]$$
$$\mathbf{P}_6 = [-1 \quad +1 \quad +1 \quad +1 \quad -1 \quad -1 \quad +1]$$

Message one is assigned PN code 0, message two is assigned PN code and message three is assigned PN code 6. Each message is spread by its assigned PN code. Note that the chip rate of the PN code is seven times the bit rate the message, contributing to a processing gain of 7. For message one:

$m_1(t)$	1							−1							1						
$m_1(t)$	1	1	1	1	1	1	1	−1	−1	−1	−1	−1	−1	−1	1	1	1	1	1	1	
$p_0(t)$	1	−1	1	1	1	−1	−1	1	−1	1	1	1	−1	−1	1	−1	1	1	1	−1	
$m_1(t)p_0(t)$	1	−1	1	1	1	−1	−1	−1	1	−1	−1	−1	1	1	1	−1	1	1	1	−1	

Note that $m_1(t)p_0(t)$ is the spread-spectrum signal of the first message. Similarly for message two:

$m_2(t)$	1							1							−1						
$m_2(t)$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	−1	−1	−1	−1	−1	−1	−1
$p_3(t)$	1	−1	−1	1	−1	1	1	1	−1	−1	1	−1	1	1	1	−1	−1	1	−1	1	1
$m_2(t)p_3(t)$	1	−1	−1	1	−1	1	1	1	−1	−1	1	−1	1	1	−1	1	1	−1	1	−1	−1

For message three:

$m_3(t)$	−1							1							1						
$m_3(t)$	−1	−1	−1	−1	−1	−1	−1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$p_6(t)$	−1	1	1	1	−1	−1	1	−1	1	1	1	−1	−1	1	−1	1	1	1	−1	−1	1
$m_3(t)p_6(t)$	1	−1	−1	−1	1	1	−1	−1	1	1	1	−1	−1	1	−1	1	1	1	−1	−1	1

The spread-spectrum signals for all three messages $m_1(t)p_0(t)$, $m_2(t)p_3(t)$, and $m_3(t)p_6(t)$ are combined to form a composite signal $C(t)$; that is,

$$C(t) = m_1(t)p_0(t) + m_2(t)p_3(t) + m_3(t)p_6(t)$$

The resulting $C(t)$ is

$C(t)$	3	−3	−1	1	1	1	−1	−1	1	−1	1	−3	1	3	−1	1	3	1	1	−3	−1

$C(t)$ is the composite signal that is transmitted in the RF band. If there are negligible errors during the transmission process, the receiver intercepts $C(t)$. In order to separate out the original messages $m_1(t)$, $m_2(t)$, and $m_3(t)$ from the composite signal $C(t)$, the receiver multiplies $C(t)$ by the assigned PN code for each message:

$C(t)p_0(t)$	3	3	−1	1	1	−1	1	−1	−1	−1	1	−3	−1	−3	−1	−1	3	1	1	3	1
$C(t)p_3(t)$	3	3	1	1	−1	1	−1	−1	−1	1	1	3	1	3	−1	−1	−3	1	−1	−3	−1
$C(t)p_6(t)$	−3	−3	−1	1	−1	−1	−1	1	1	−1	1	3	−1	3	1	1	3	1	−1	3	−1

Then the receiver integrates, or adds up, all the values over each bit period. The functions $M_1(t)$, $M_2(t)$, and $M_3(t)$ are the results:

$C(t)p_0(t)$	3	3	−1	1	1	−1	1	−1	−1	−1	1	−3	−1	−3	−1	−1	3	1	1	3	1
$M_1(t)$							7							−9							7

$C(t)p_0(t)$	3	3	−1	1	1	−1	1	−1	−1	−1	1	−3	−1	−3	−1	−1	3	1	1	3	1
$M_1(t)$							7							−9							7

$C(t)p_3(t)$	3	3	1	1	−1	1	−1	−1	−1	1	1	3	1	3	−1	−1	−3	1	−1	−3	−1
$M_2(t)$							7							7							−9

$C(t)p_6(t)$	−3	−3	−1	1	−1	−1	−1	1	1	−1	1	3	−1	3	1	1	3	1	−1	3	−1
$M_3(t)$							−9							7							7

A decision threshold looks at the integrated functions $M_1(t)$, $M_2(t)$, and $M_3(t)$. The decision rules used are

$$\tilde{m}(t) = 1 \quad \text{if } M(t) > 0$$
$$\tilde{m}(t) = -1 \quad \text{if } M(t) < 0$$

After applying the above decision rules, we obtain

$\tilde{m}_1(t)$	1	−1	1
$\tilde{m}_2(t)$	1	1	−1
$\tilde{m}_3(t)$	−1	1	1

3.5.2.3 Concluding Remarks

We define the discrete-time autocorrelation of a real-valued sequence **x** to be

$$R_{\mathbf{x}}(i) = \sum_{j=0}^{J-1} x_j x_{j-1} \tag{3.9}$$

In other words, for each successive shift i, we calculate the summation of the product of x_j and its shifted version x_{j-i}. We proceed to calculate the autocorrelation of the PN sequence $\mathbf{p}_0$. Table 3.3 calculates the autocorrelation $R_{\mathbf{p}_0}(i)$ of $\mathbf{p}_0$.

Note that $\mathbf{p}_0 = [+1 \quad -1 \quad +1 \quad +1 \quad +1 \quad -1 \quad -1]$ and the shifted sequence $\mathbf{p}_{0,j-i}$ are also shown for each shift i in the table. The resulting $R_{\mathbf{p}_0}(i)$ for each shift i is shown on the right side of the table. Figure 3.11 depicts the autocorrelation function $R_{\mathbf{p}_0}(i)$ as a function of time shift i.

In Figure 3.11, we see that the autocorrelation function reaches a peak at every seventh shift. For all other shifts (or time offsets), the autocorrelation stays at the minimum of −1. The autocorrelation property shown in

Table 3.3

Calculation of Autocorrelation for the Sequence $\mathbf{P}_0$

i	$\mathbf{p}_{0,j-i}$							$R_{\mathbf{p}_0}(i)$
0	1	−1	1	1	1	−1	−1	7
1	−1	1	−1	1	1	1	−1	−1
2	−1	−1	1	−1	1	1	1	−1
3	1	−1	−1	1	−1	1	1	−1
4	1	1	−1	−1	1	−1	1	−1
5	1	1	1	−1	−1	1	−1	−1
6	−1	1	1	1	−1	−1	1	−1
7	1	−1	1	1	1	−1	−1	7
8	−1	1	−1	1	1	1	−1	−1
9	−1	−1	1	−1	1	1	1	−1
10	1	−1	−1	1	−1	1	1	−1
11	1	1	−1	−1	1	−1	1	−1
12	1	1	1	−1	−1	1	−1	−1
13	−1	1	1	1	−1	−1	1	−1
14	1	−1	1	1	1	−1	−1	7
15	−1	1	−1	1	1	1	−1	−1

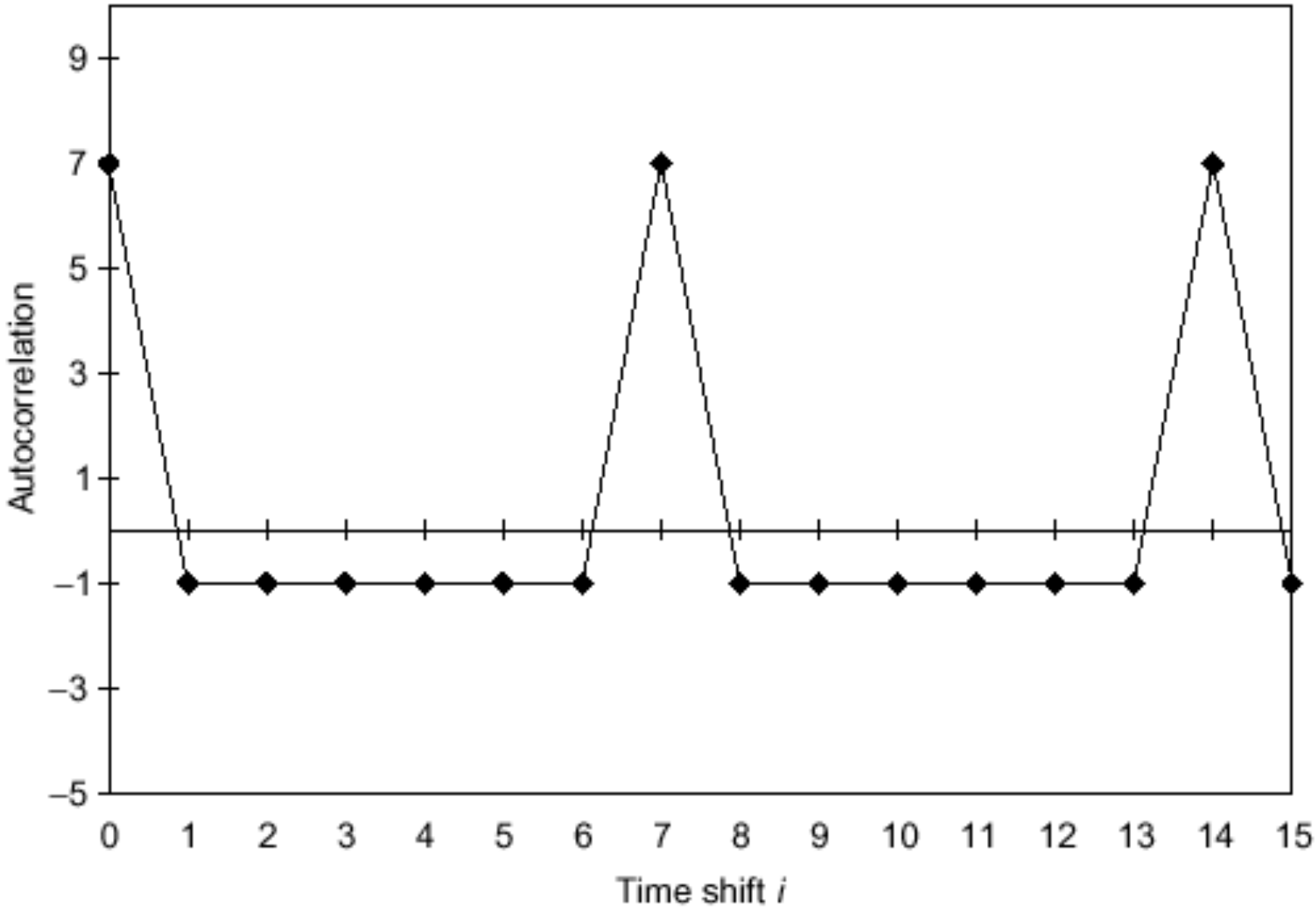


Figure 3.11 Autocorrelation function of PN sequence $\mathbf{p}_0$.

Figure 3.11 is important because it helps the initial acquisition and synchronization of the PN code at the receiver. A high correlation occurs only when the codes are aligned (i.e., when time shift i is zero); if the codes are not aligned, a low correlation results.

In practice, the receiver possesses an original copy of the PN code (i.e., $\mathbf{p}_{0,j}$). The receiver would like to acquire an incoming sequence $\mathbf{p}_{0,j-i}$ at an arbitrary phase. The receiver only has to "slide" the incoming sequence and calculate the autocorrelation. When the autocorrelation reaches a maximum, then the two codes are in-phase and have a time shift of zero. In an IS-95 CDMA system, this is in fact done by a mobile station to acquire the unmodulated pilot channel. This acquisition scheme can also be used when the spreading code length is equal to the data bit period.

In IS-95 CDMA, the reverse link uses the "long" PN code for channelization. The long code is so called because its length is literally very long. The long code has a length of $2^{42} - 1$ chips and is generated using a 42-stage register.

We saw in Section 3.5.1 that the forward link uses the Walsh code for channelizing individual users of a particular base station. However, the forward link also uses the PN code. Each base station is assigned a unique PN code that is superimposed on top of the Walsh code. This is done to provide isolation among the different base stations (or sectors); the isolation is necessary because each base station uses the same 64 Walsh code set. The PN code used on the forward link is called the "short" code. It is so called because its length is relatively short. The short code is generated using a 15-stage register and has a length of $2^{15} - 1$ chips.

propagation delay, which is inherent in a mobile channel. For example, the two codes mentioned previously are orthogonal when they are perfectly aligned:

x_i	−1	−1	1	1	−1	−1	1	1
y_i	−1	1	1	−1	−1	1	1	−1

However, if y_i suffers a delay of one chip as a result of propagation delay in a mobile channel, then

x_i	−1	−1	1	1	−1	−1	1	1
y_{i-1}	1	1	−1	−1	1	1	−1	−1

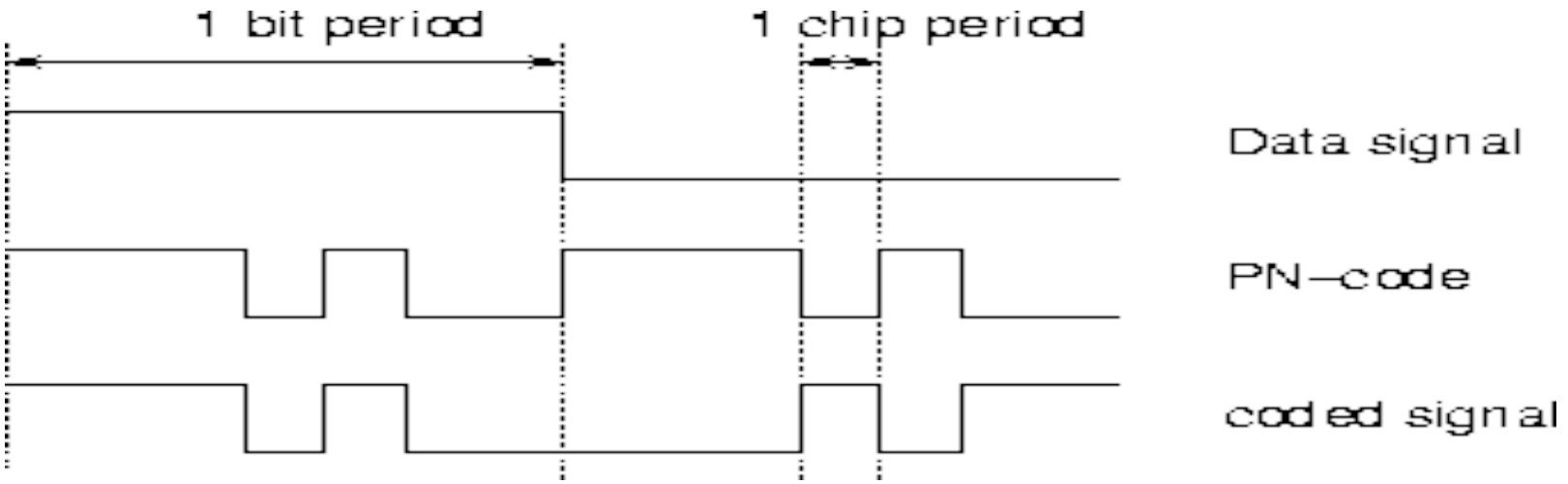
Readers can easily verify that these two sequences are no longer orthogonal. If the codes are not orthogonal due to synchronization or channel impairment, then multiple access messages in the same band can no longer be separated from one another via code-orthogonality. The results are correlation crosstalk and mutual interference. In essence, an additional condition needs to be adhered to; that is,

$$\int_0^{T-\tau} x(t)\,y(t+\tau)\,dt = 0$$

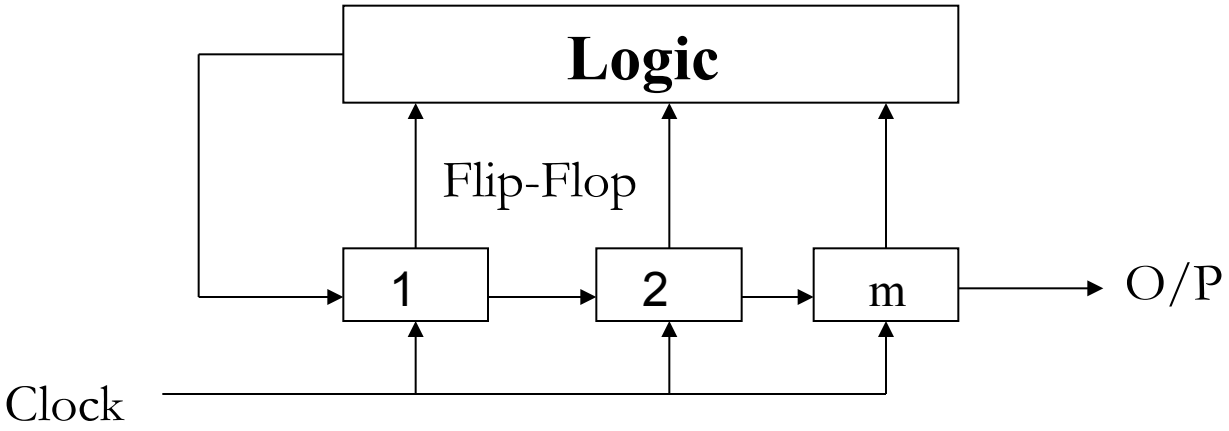
$$\int_{T-\tau}^{T} x(t)\,y(t+\tau-T)\,dt = 0$$

Therefore, simple orthogonality between two aligned codes is not enough—the above two partial correlations must also be zero, or at least small, for any value of τ likely encountered in the system [2]. Radio propagation is treated in

Pseudo-Random Noise Codes:

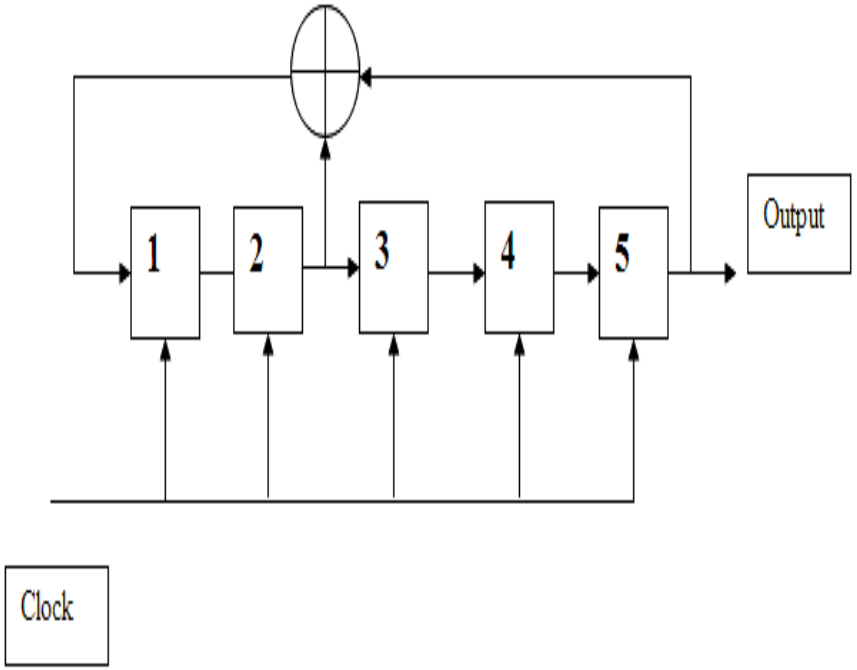


Feedback Shift Register:



By: Dr.Mohab Mangoud

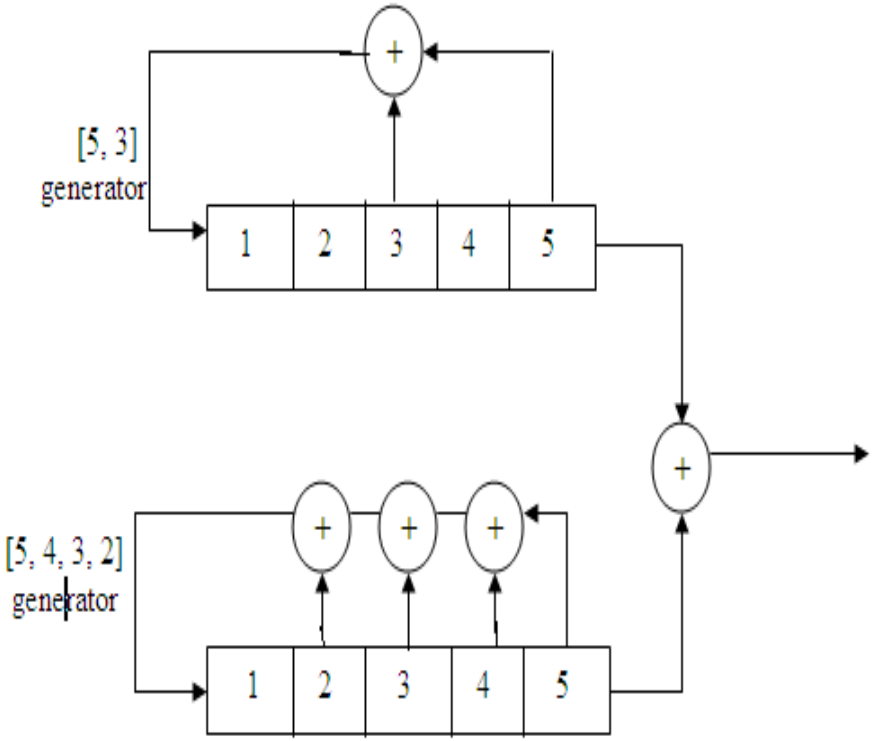
Maximal Length Sequences (M-Sequences)

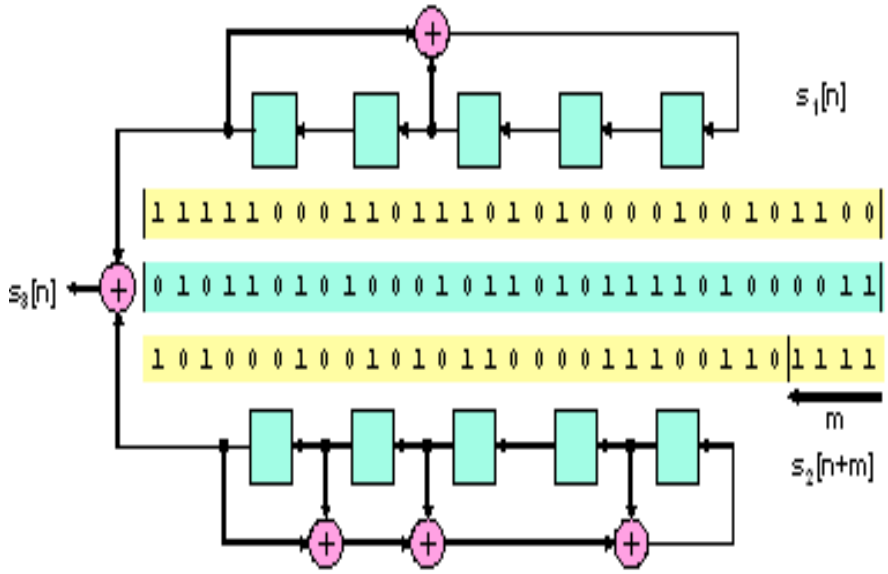


Feed-Back symbol	1	0	0	0	0	O/P Symbol
0	0	1	0	0	0	0
1	1	0	1	0	0	0
0	0	1	0	1	0	0
1	1	0	1	0	1	0
1	1	1	0	1	0	1
1	1	1	1	0	1	0
0	0	1	1	1	0	1
0	1	0	1	1	1	0
0	1	1	0	1	1	1

By: Dr.Mohab Mangoud

Gold Codes:





k = 5 Register ⇒ N = 2^k + 1 = 33 Gold-Codes:
$s_3[n]$ für m = 0 .. N-1, sowie $s_1[n]$ und $s_2[n]$

By: Dr.Mohab Mangoud

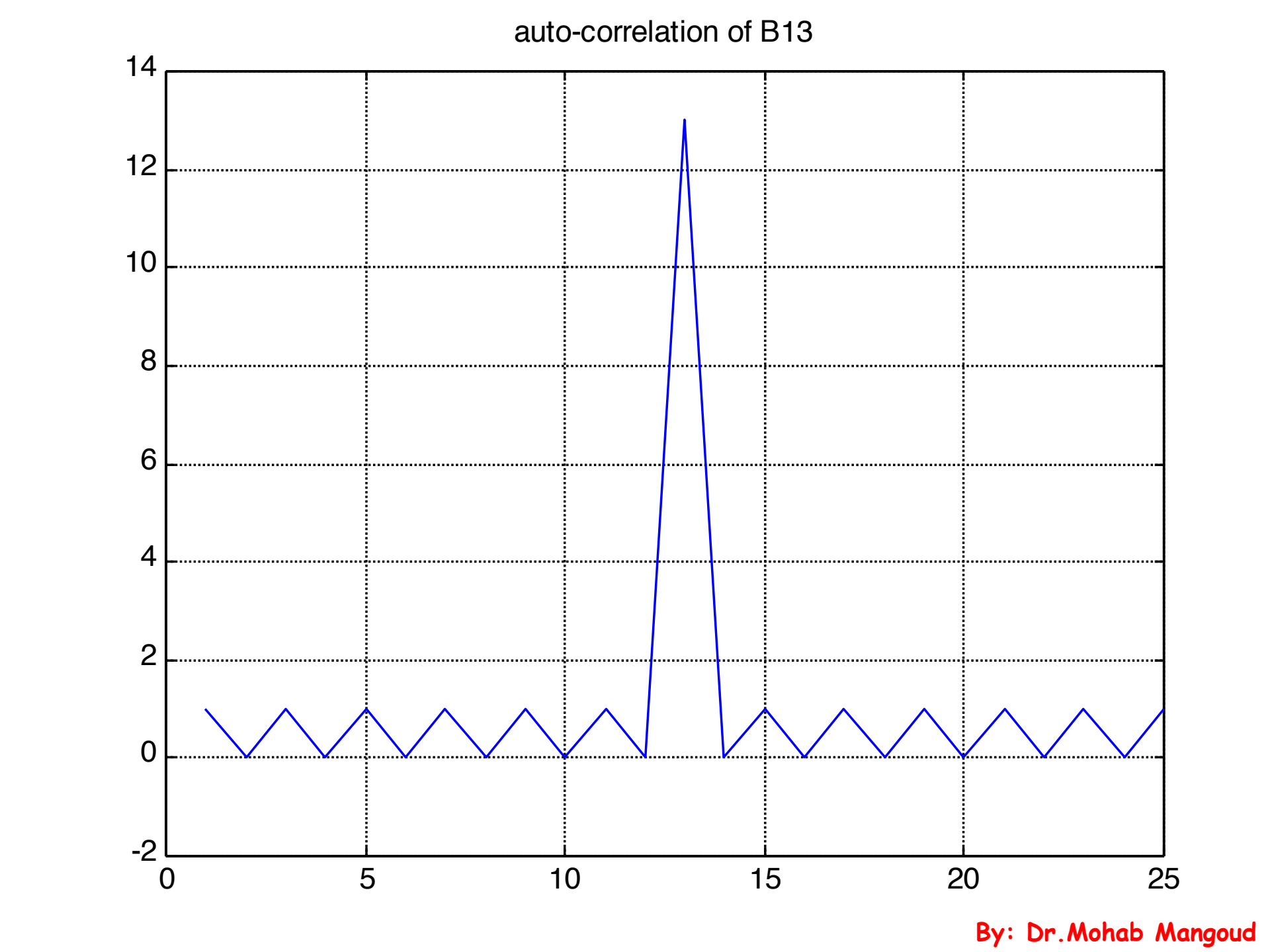
1) Barker Codes

Examples of Barker codes are:

Barker 11: [1,1,1,1,-1,
-1,1,1,-1,1,-1]

Barker 13: [1,1,1,1,1,-1,
-1,1,1,-1,1,-1,1]

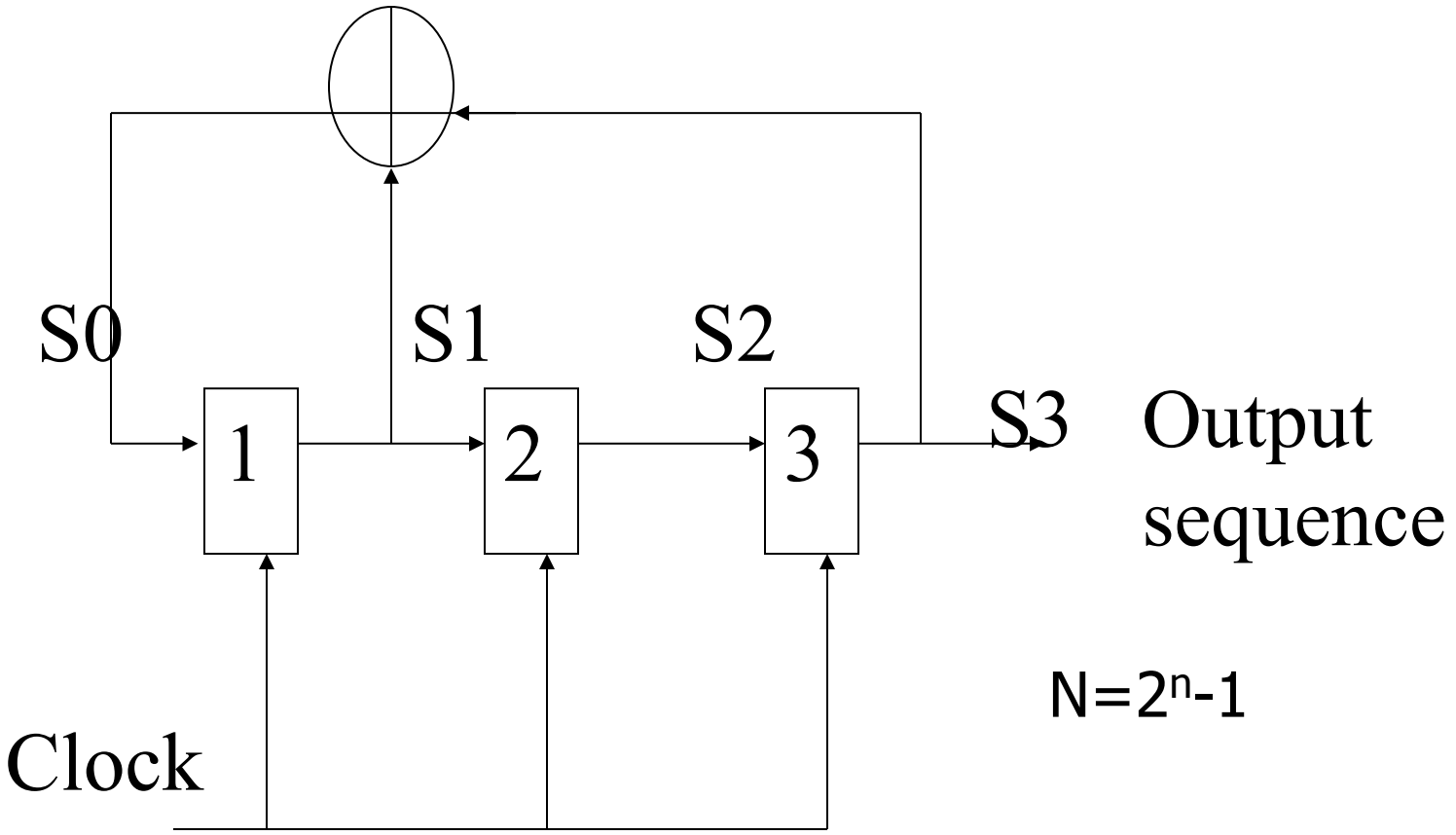
By: Dr.Mohab Mangoud



By: Dr.Mohab Mangoud

2) Maximal Length Sequences

(M-Sequences)

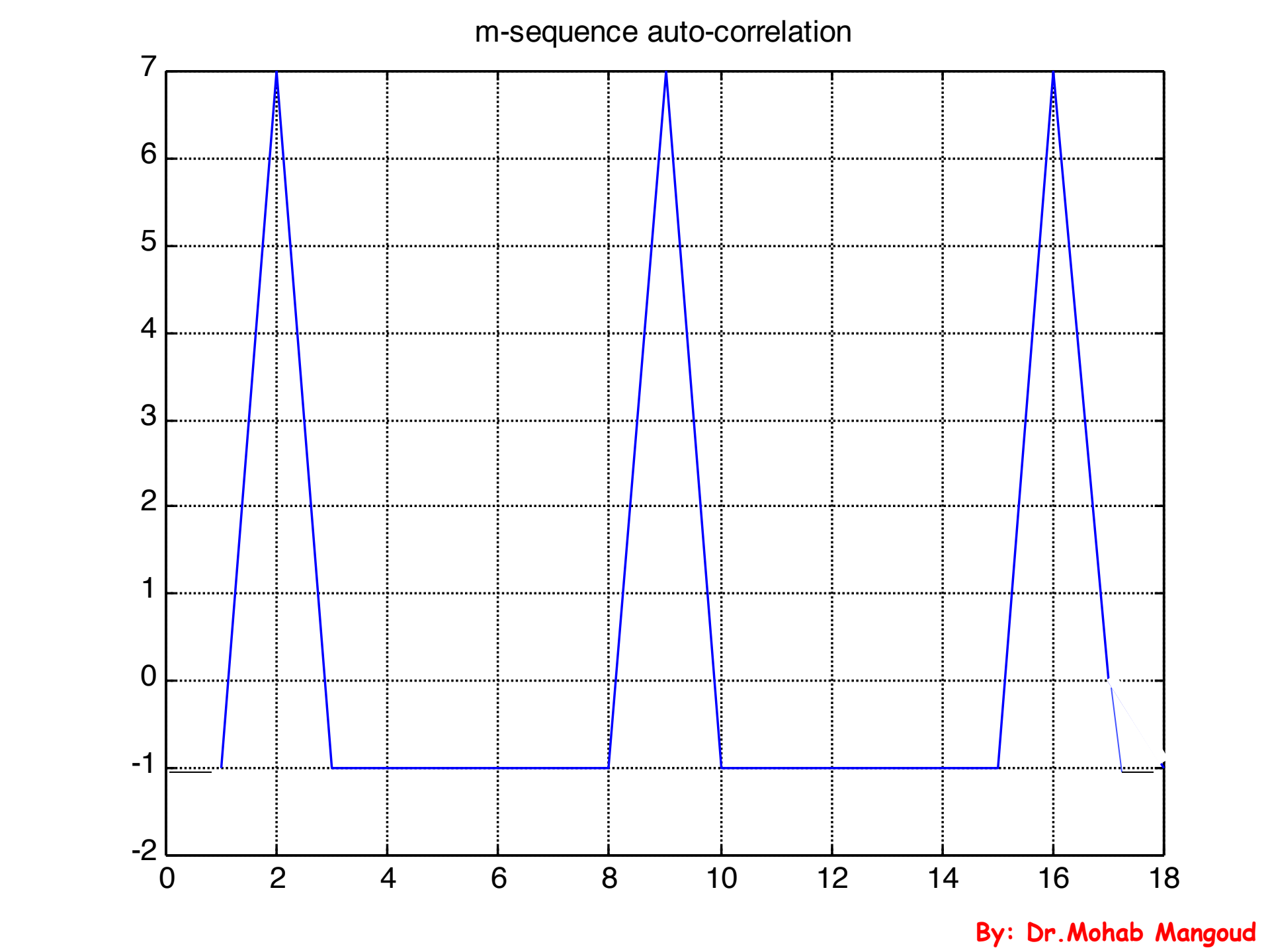


$N=2^n-1$

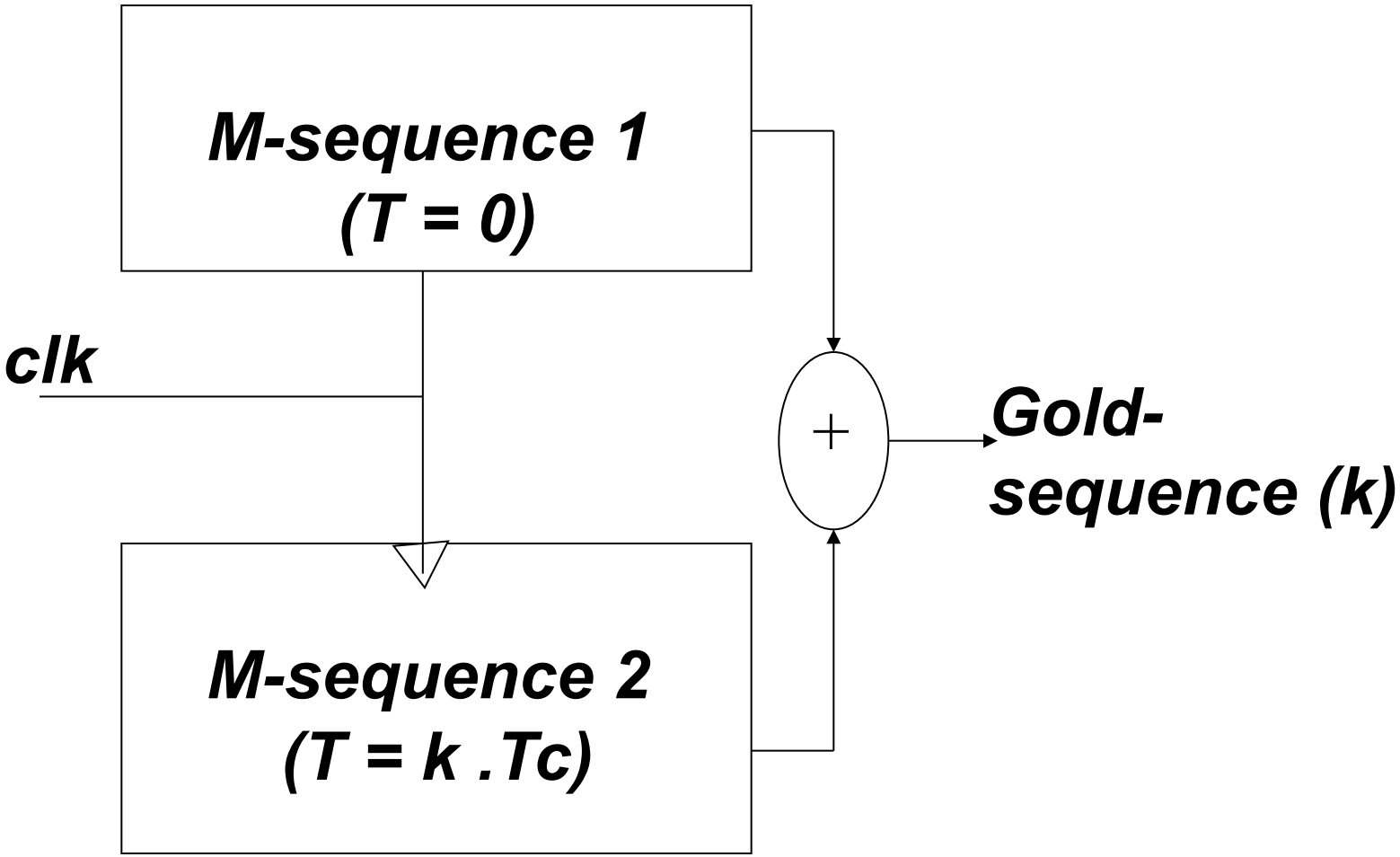
By: Dr.Mohab Mangoud

Choosing a maximum length sequence

t register ngth, m	feedback taps	#m- sequences
2*	[2,1]	2
3*	[3,1]	2
4	[4,1]	2
5*	[5,2],[5,4,3,2],[5,4,2,1]	6
6	[6,1],[6,5,2,1],[6,5,3,2]	6
7*	[7,1],[7,3],[7,3,2,1],[7,4,3,2],[7,6,4,2],[7,6,3,1], [7,6,5,2],[7,6,5,4,2,1],[7,5,4,3,2,1]	18
8	[8,4,3,2],[8,6,5,3],[8,6,5,2],[8,5,3,1],[8,6,5,1], [8,7,6,1],[8,7,6,5,2,1],[8,6,4,3,2,1]	16

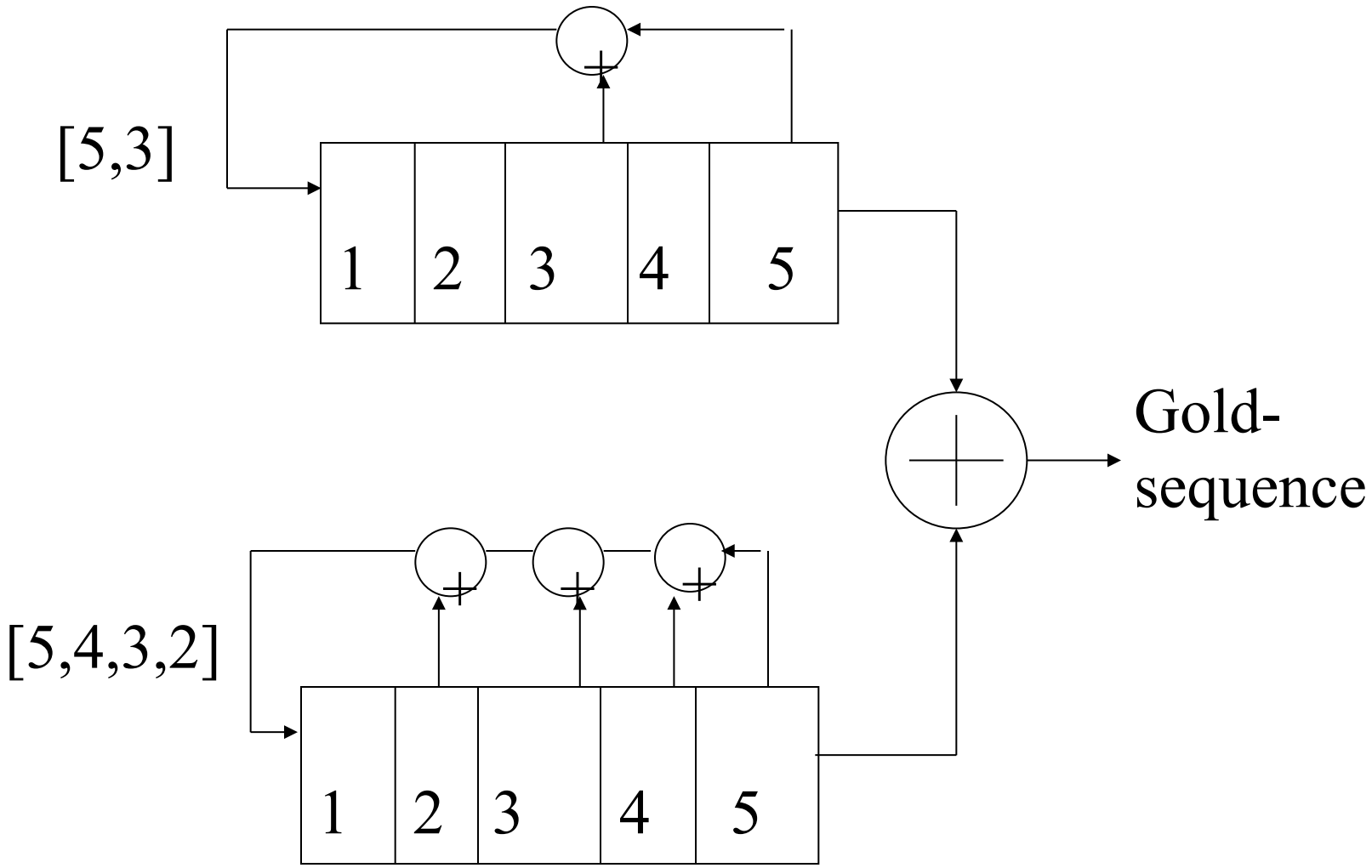

m-sequence auto-correlation
7
6
5
4
3
2
1
0
-1
-2
0
2
4
6
8
10
12
14
16
18
By: Dr.Mohab Mangoud

3) Gold Codes Families

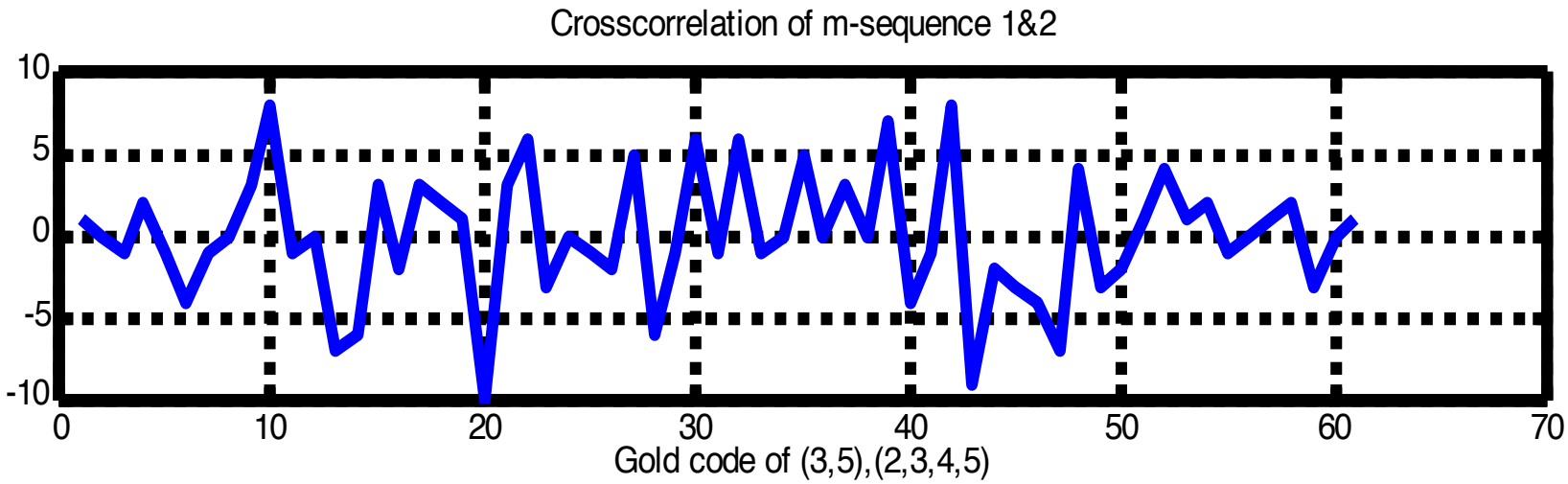


By: Dr.Mohab Mangoud

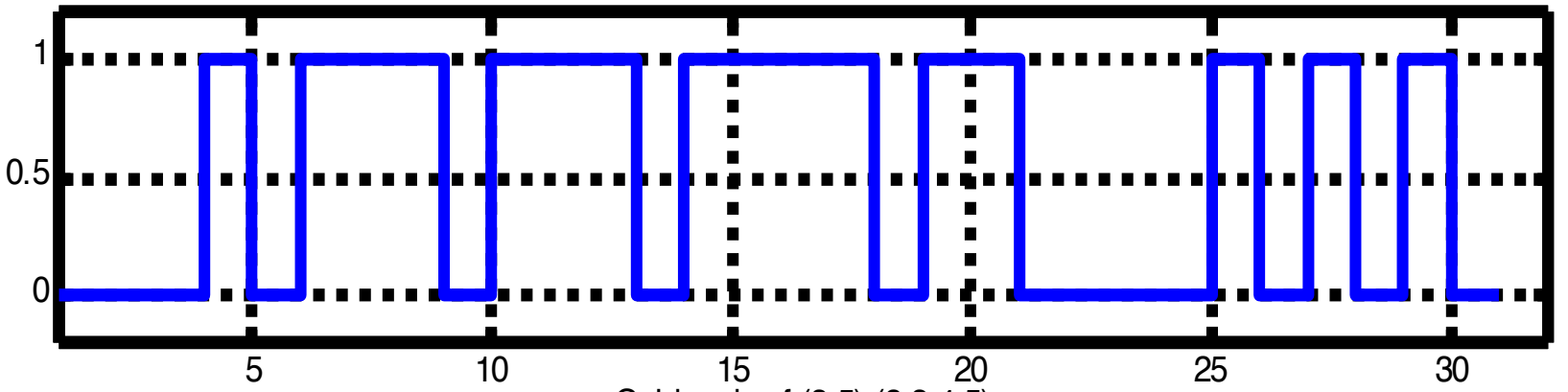
Example



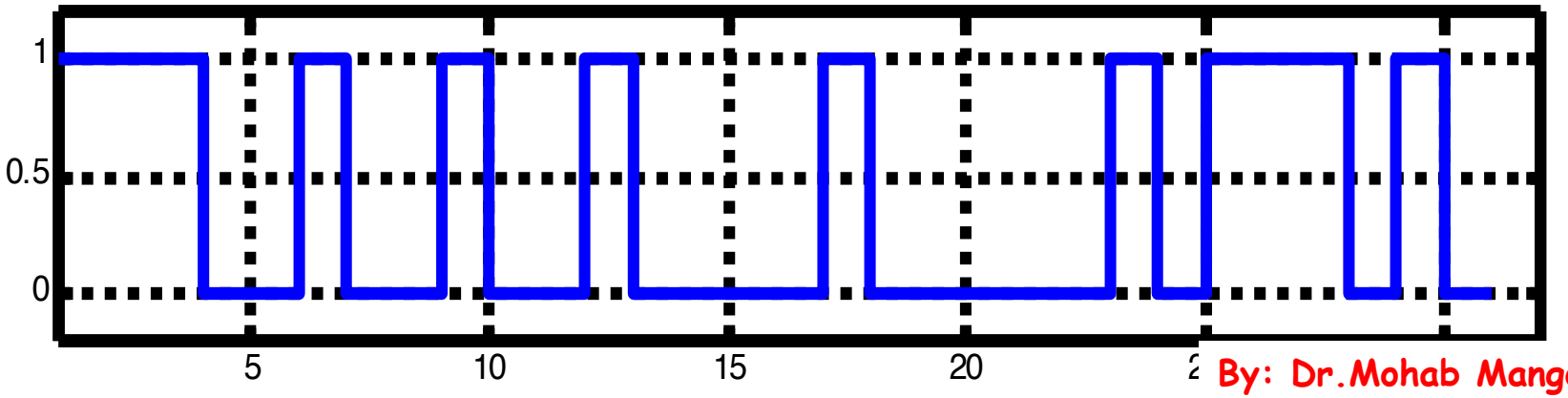
By: Dr.Mohab Mangoud

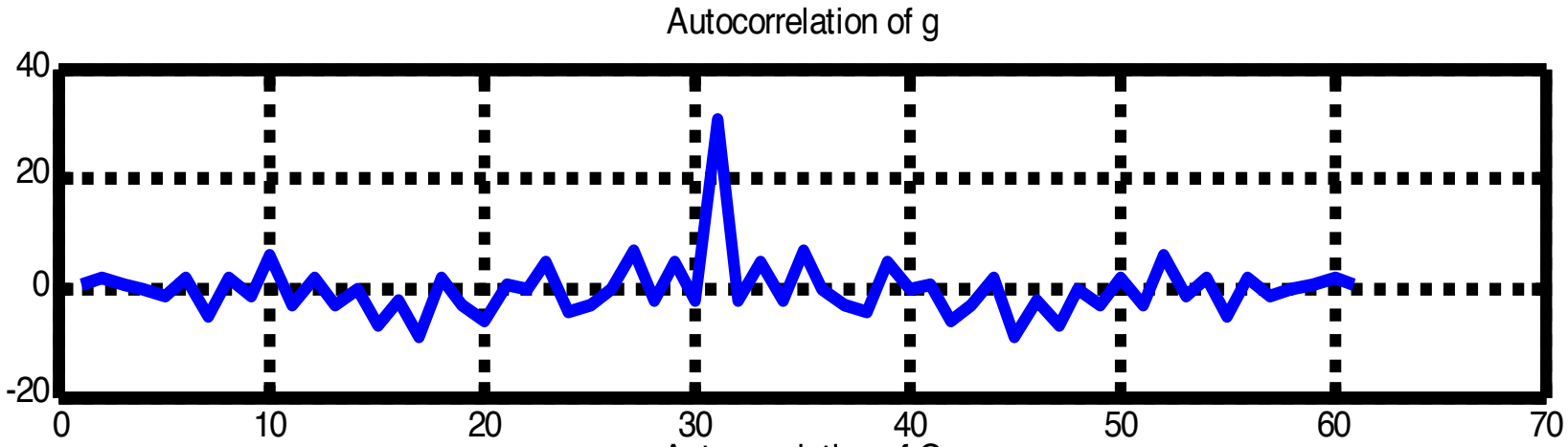


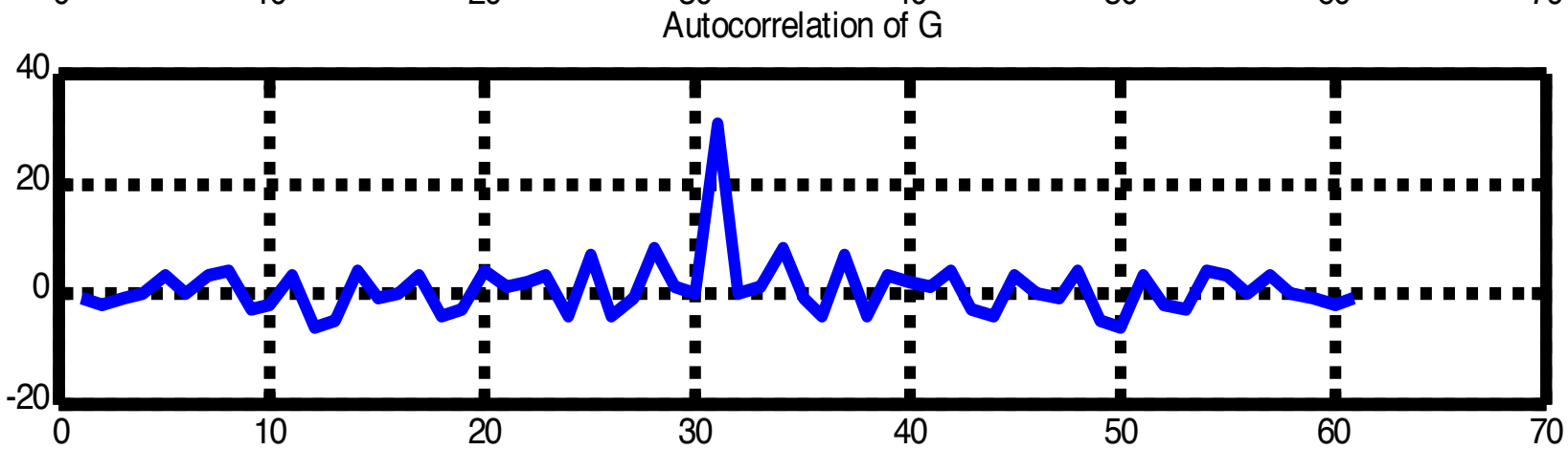
Gold code of (3,5),(2,3,4,5)

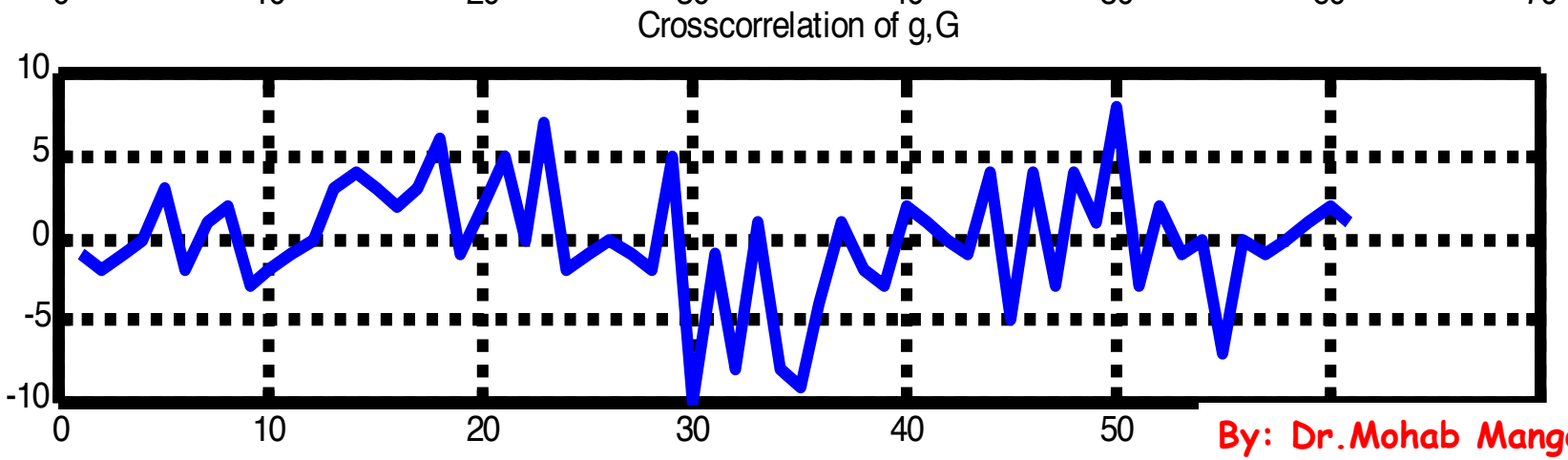


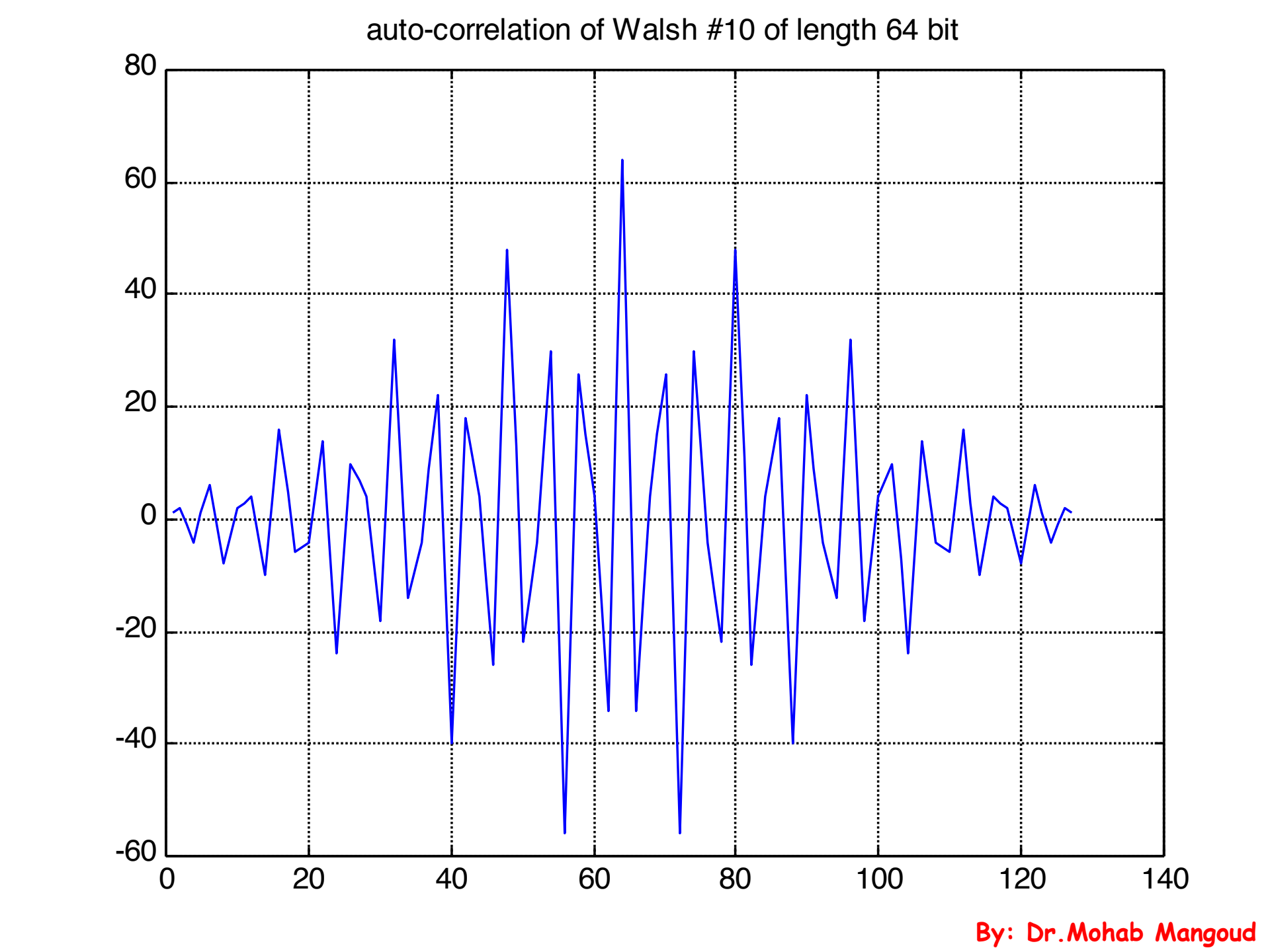
Gold code of (3,5),(2,3,4,5)

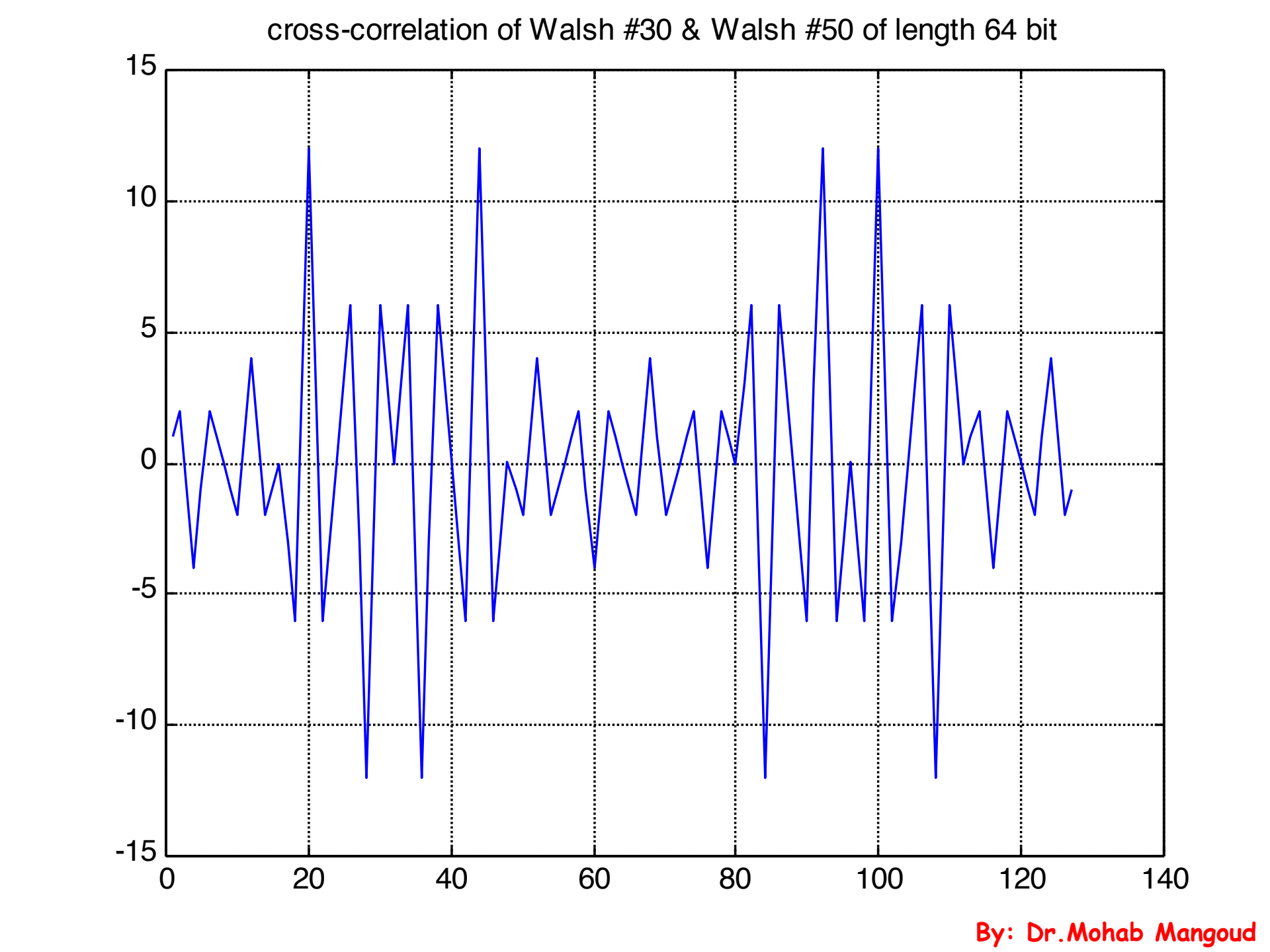




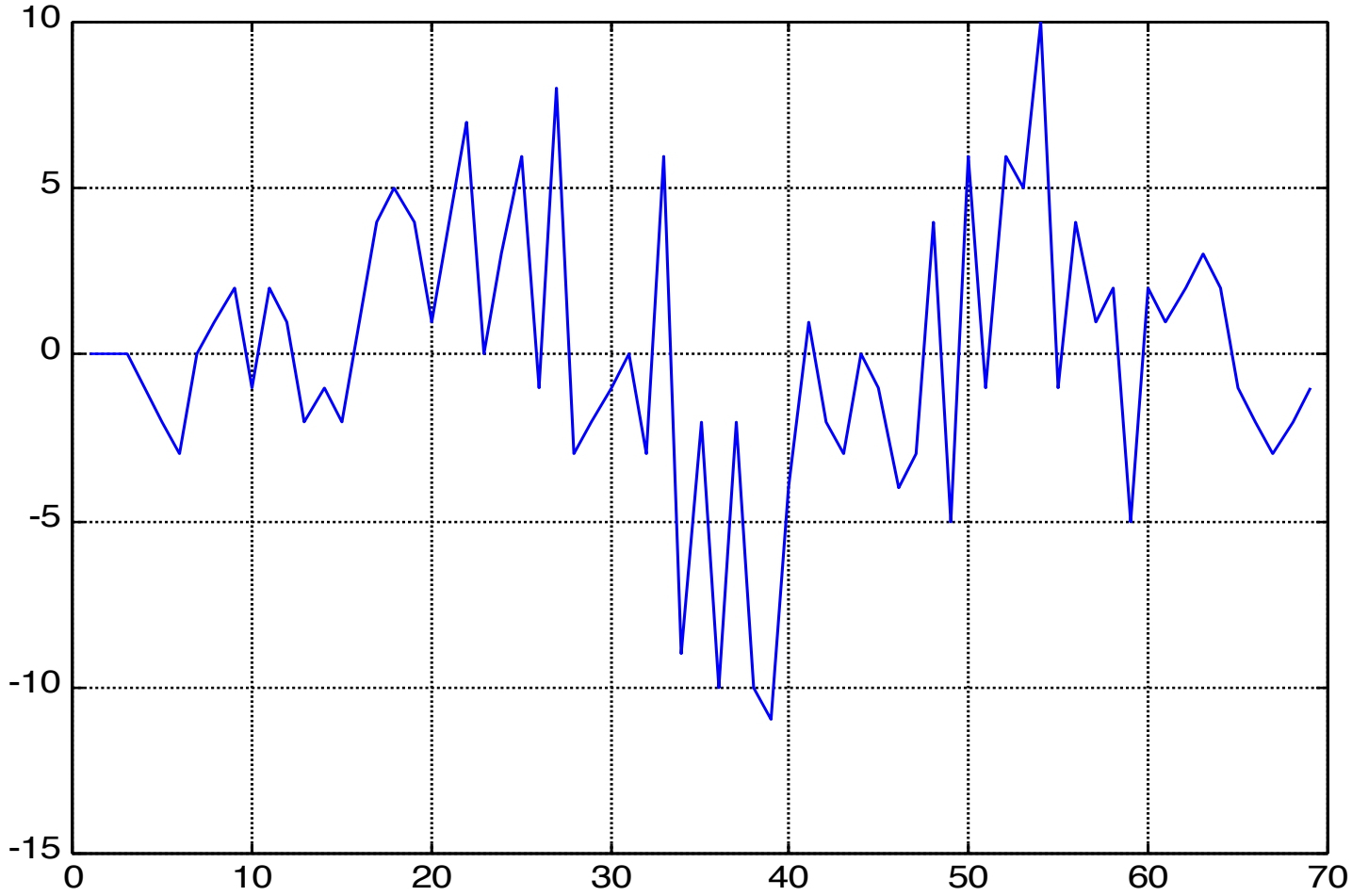





auto-correlation of Walsh #10 of length 64 bit
80
60
40
20
0
-20
-40
-60
0
20
40
60
80
100
120
140
By: Dr.Mohab Mangoud


cross-correlation of Walsh #30 & Walsh #50 of length 64 bit
15
10
5
0
-5
-10
-15
0
20
40
60
80
100
120
140
By: Dr.Mohab Mangoud

Orthogonal Gold Codes:



By: Dr.Mohab Mangoud

m-sequences

- Using a length *n* linear feedback shift register, the longest possible sequence is $N=2^n-1$.
  - This corresponds to registers cycling through all states except the all zeros state.
  - Once in the all zeros state, the state will never change. Thus this state must be avoided.
- *m*-sequences have length $N=2^n-1$ and are thus called "maximal length" sequences or *m*-sequences.
- These sequences also have very good autocorrelation properties and other noise-like characteristics

Properties of *m*-sequences

Property 1 *The number of ones in the code minus 1 equals the number of zeros in the code. That is each code is of length* $2^k - 1$ *and has* 2^{k-1} *1's and* $(2^{k-1} - 1)$ *0's.*

Property 2 *If a window of length k is slid across the sequence, every k-tuple of values will appear exactly once except the all zeros k-tuple.*

Property 3 *The modulo-2 sum of an m-sequence with a phase shifted version of itself will result in another shifted version of itself.*

Property 4 *The periodic auto-correlation function is two-valued taking on the values*

$$C_{k,k}(b) = \begin{cases} 1 & b = lN \\ -\frac{1}{N} & b \neq lN \end{cases} \tag{5.30}$$

Properties of *m*-sequences

- **Property:** Define a run as a subsequence of identical symbols within the *m*-sequence. The length of the subsequence is the length of the run. Then for any *m*-sequence there is
  - 1 run of ones of length *n*
  - 1 run of zeros of length *n-1*
  - 1 run of ones and 1 run of zeros of length *n-2*
  - : :
  - 2^{n-3} runs of ones and 2^{n-3} runs of zeros of length 1

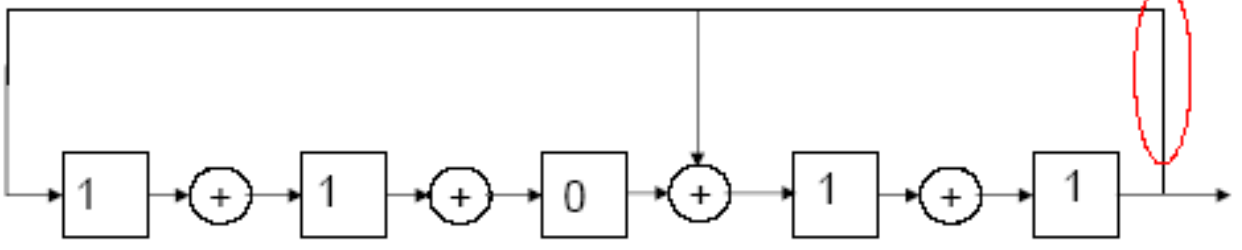
Properties of *m*-sequences

- **Proof:**
  - Again consider the shift register. Clearly there can be no run of *n* zeros since that would require the all zeros state to appear, which it cannot. Thus we cannot have a string of *n* zeros.
  - Similarly, we know that the all ones state occurs and occurs only once. This means that a run of ones of length *n* will occur but not more than once. Further, any run of ones longer than *n* would require consecutive appearances of the all ones state which cannot happen.
  - We also know that a run of *n-1* ones cannot occur. This is because a run of *n-1* ones requires a zero on either side of it and that requires the states 0111...1 and 111..110 to occur. However, we know that both of these states must occur before and after the run of *n* ones. Since the states cannot occur more than once, a string of *n-1* ones must not occur.
  - Consider a run of *k* ones with $0 \le k \le n-1$. The shift register must pass through the state $0\underbrace{11\ldots1}_{k}0$. There are *n-k-2* remaining values in the shift register. Thus, there are 2^{n-k-2} ways this run can occur. Similarly for *k* zeros.

Example: $N = 31$, $\mathbf{g} = 01001$

g_0 is always 1 thus not always represented

- Initially load register with 11011

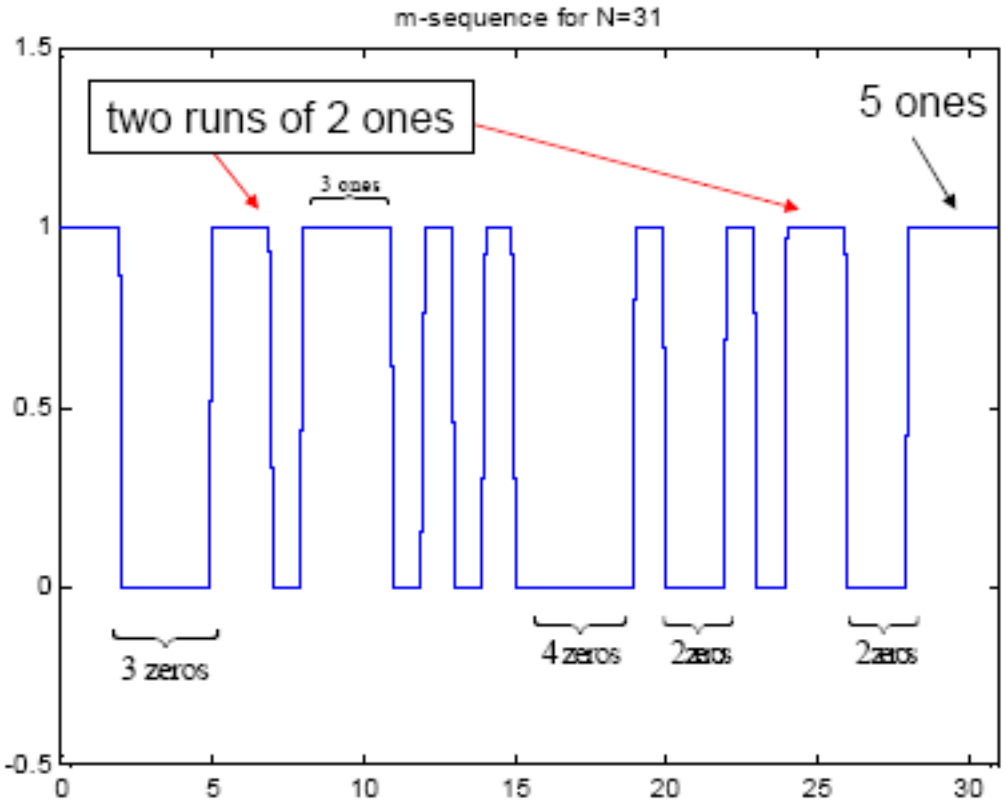


State	Output	Next State
11011	1	11111
11111	1	11101
11101	0	11100
11100	0	01110
01110	1	00111
00111	1	10001

$\mathbf{g} = [g_1, g_2, g_3, g_4, g_5]$
$= [0\ 1\ 0\ 0\ 1]$

$g(D) = 1 + g_1 D + g_2 D^2 + g_3 D^3 + g_4 D^4 + g_5 D^5$
$= 1 + D^3 + D^5$

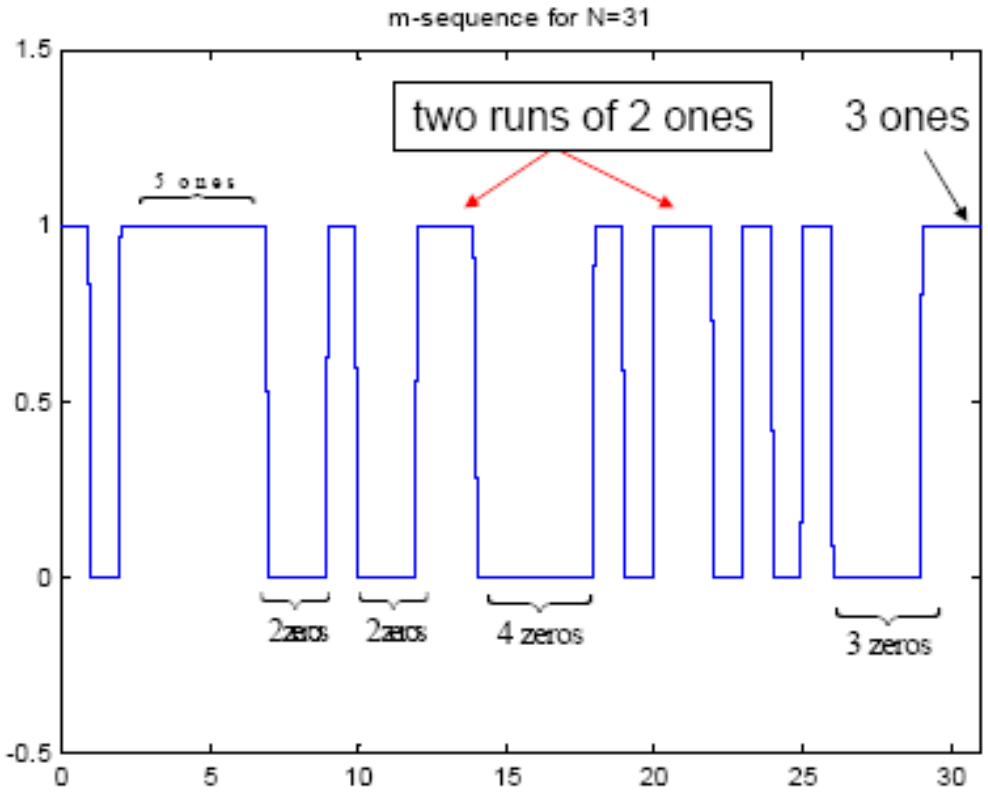
Example: N = 31, **g** = 00101



1 run of five 1's
1 run of three 1's
2 runs of two 1's
4 runs of one 1
1 run of four 0's
1 run of three 0's
2 runs of two 0's
4 runs of one 0

Note that we plot **d** rather than **a**

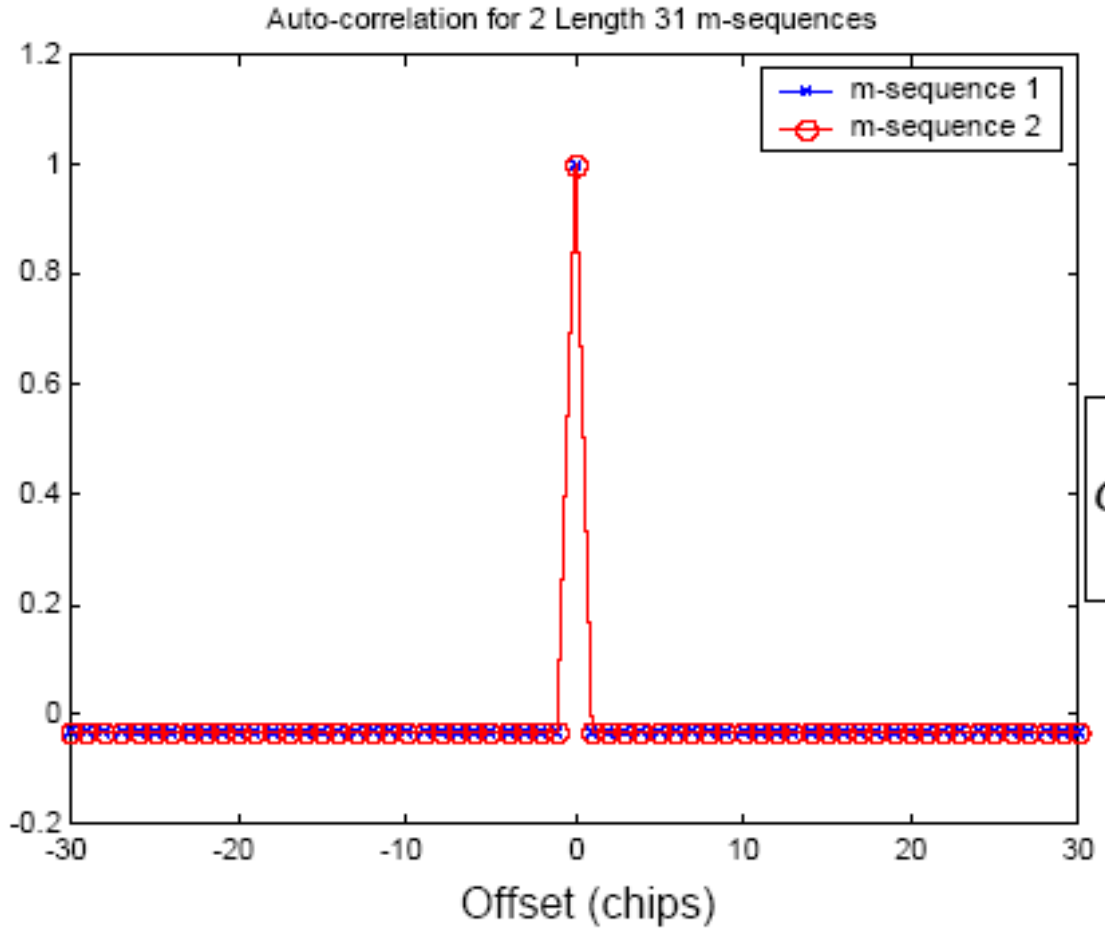
Example: N = 31, g = 01111



1 run of five 1's
1 run of three 1's
2 runs of two 1's
4 runs of one 1
1 run of four 0's
1 run of three 0's
2 runs of two 0's
4 runs of one 0

Note that we plot **d** rather than **a**

Example: Auto-correlation



$n = 5$
$N = 31$

$$C_{k,k}(m) = \begin{cases} 1 & m = l * 31 \\ -\frac{1}{31} & m \neq l * 31 \end{cases}$$

Recall that $C_{kk}(m)$ is defined on **a** rather than **d**

Thank you for attending this course

- **End of the course**

Feedback survey

- Tell me what you think about this course
  - STOP
  - Continue
  - I suggest

- Please email me your opinion and suggestions on:

mangoud@ eng.uob.bh
mangoud@vt.edu