CDMA TECHNOLOGY

INSTRCTOR: NAZILA SAFAVI

SECTION 2:
CDMA Forward Channels

Objectives

- **To review types and methods of generation of”**
  - Pilot Channel
  - Paging Channel
  - Sync Channel

Pilot Channel Generation



- The Walsh function zero spreading sequence is applied to the Pilot
- The use of short PN sequence offsets allows for up to 512 distinct Pilots per CDMA channel
- The PN offset *index* value (0-511 inclusive) for a given pilot PN sequence is multiplied by 64 to determine the actual offset
  - Example: 15 (offset index) x 64 = 960 PN chips
  - Result: The start of the pilot PN sequence will be delayed 960 chips x 813.8 nanoseconds per chip = 781.25 µs
- The quadrature spreading and baseband filtering (not shown), which are performed as with all the other forward and reverse code channels, will be discussed later

Walsh Codes Generation

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

$W_1 = 0$ $\quad$ $W_2 = \begin{matrix} 0 & 0 \\ 0 & 1 \end{matrix}$ $\quad$ $W_4 = \begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{matrix}$

$W_1 = 1$ $\quad$ $W_2 = \begin{matrix} 1 & 1 \\ 1 & 0 \end{matrix}$ $\quad$ $W_4 = \begin{matrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{matrix}$

Properties of the Walsh Codes

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

- When a Walsh code is XORed chip by chip with itself, the result is all 0's (100% correlation)
- When a Walsh code is XORed chip by chip with its logical negation, the result is all 1's (–100% correlation)
- When a Walsh code is XORed chip by chip with any other code or its logical negation, the result is half 0's and half 1's (0% correlation)

0 1 0 1	**0 1 0 1**	**0 1 0 1**	**0 1 0 1**
0 0 0 0	**0 1 0 1**	**0 0 1 1**	**0 1 1 0**
0 1 0 1	**0 0 0 0**	**0 1 1 0**	**0 0 1 1**
0 1 0 1	**0 1 0 1**	**0 1 0 1**	**0 1 0 1**
1 1 1 1	**1 0 1 0**	**1 1 0 0**	**1 0 0 1**
1 0 1 0	**1 1 1 1**	**1 0 0 1**	**1 1 0 0**

Orthogonality (Bit Strings)

a = (0, 1, 1, 0) b = (0, 1, 0, 1) -b = (1, 0, 1, 0)

+1 +1 +1

-1 -1 -1

a = (+1, -1, -1, +1) b = (+1, -1, +1, -1) -b = (-1, +1, -1, +1)

$a \bullet a = (+1)(+1) + (-1)(-1) + (-1)(-1) + (+1)(+1) = 1 + 1 + 1 + 1 = 4$

$a \bullet b = (+1)(+1) + (-1)(-1) + (-1)(+1) + (+1)(-1) = 1 + 1 - 1 - 1 = 0$

$b \bullet (-b) = (+1)(-1) + (-1)(+1) + (+1)(-1) + (-1)(+1) = -1 - 1 - 1 - 1 = -4$

a XOR a = (0, 0, 0, 0) = 100% match

a XOR b = (0, 0, 1, 1) = 50% match & 50% no-match

b XOR -b = (1, 1, 1, 1) = 100% no-match

Walsh Codes: Spreading a Signal

N = 4

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

- Select a Walsh code
- If the bit is 0, send the Walsh code
- If the bit is 1, send the logical negation of the Walsh code

Selected Walsh code: **0 1 0 1** (used to represent code symbols “0”)

Negated Walsh code: **1 0 1 0** (used to represent code symbols “1”)

Original sequence: **1 0 0 1 1 0**

Spread sequence: **1 0 1 0 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 0 1 0 1**

Walsh Codes: De-spreading a Signal (Using the Right Code)

N = 4

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

- XOR the received sequence with the same Walsh code used for spreading it
- Perfect synchronization is a must
- If a Walsh code produces “N” 0’s, the original bit was a 0
- If a Walsh code produces “N” 1’s, the original bit was a 1

(Original sequence):	1				0				0				1				1				0			
Received sequence:	1	0	1	0	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0	0	1	0	1
Walsh code:	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
XOR:	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
					1				0				1				1				0			

Walsh Codes: De-spreading a Signal (Using the Wrong Code)

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

- If the received sequencel is XORed with a Walsh code other than the one used for spreading it, the result is neither groups of all 0's or groups of all 1's, but groups containing half the number of 0's and half the number of 1's pseudorandomly distributed
- This indicates that nothing was coded using this Walsh code

(Original sequence):	1				0				0				1				1				0			
Received sequence:	1	0	1	0	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0	0	1	0	1
Walsh code:	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
XOR:	1	1	0	0	0	0	1	1	0	0	1	1	1	1	0	0	1	1	0	0	0	0	1	1
			?				?				?				?				?				?	

4-Bit Pseudo Noise (PN) Code Generator

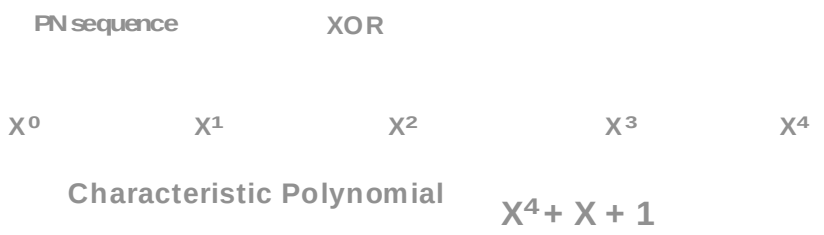
p_1 p_2 p_3 p_4

p_4 p_2 p_3 p_4

$p5 = p_1 + p_4$

p_4 p_5 p_2 p_3

PN Sequence Generation using a Linear Feedback Register



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

Randomness of the Generated String

2^n-1 1 0 0 1 1 0 1 0 1 1 1 1 0 0 0

if a zero is inserted:

2^n 1 0 0 1 1 0 1 0 1 1 1 1 0 0 0 0

Correlation, Auto-correlation, and Cross-correlation

	1	0	0	1	1	0	1	0	1	1	1	1	0	0	0	0
	1	0	0	1	1	0	1	0	1	1	1	1	0	0	0	0
XOR:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	1	1	0	1	0	1	1	1	1	0	0	0	0
	0	1	1	0	0	1	0	1	0	0	0	0	1	1	1	1
XOR:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

- In general, the **correlation** between two same-length bit strings is defined as the “degree of similarity” between them
- When the correlation is determined between two copies of the same string, it is called **auto-correlation**
- When the correlation is determined between any two same-length strings , it is called **cross-correlation**
- A PN string (and, as a matter of fact, any bit string), when correlated (XORed chip by chip) with an unshifted copy of itself, shows 100% correlation (the result is all 0’s); and when correlated (XORed chip by chip) with an unshifted copy of its logical negation , it shows -100% correlation (the result is all 1’s)

No Correlation

	1	0	0	1	1	0	1	0	1	1	1	1	0	0	0	0
	0	1	0	0	1	1	0	1	0	1	1	1	1	0	0	0
XOR:	1	1	0	1	0	1	1	1	1	0	0	0	1	0	0	0

	1	0	0	1	1	0	1	0	1	1	1	1	0	0	0	0
	0	1	1	0	1	0	1	1	1	1	0	0	0	0	1	0
XOR:	1	1	1	1	0	0	0	1	0	0	1	1	0	0	1	0

	1	0	0	1	1	0	1	0	1	1	1	1	0	0	0	0
	1	0	0	1	0	1	0	0	0	0	1	1	1	1	0	1
XOR:	0	0	0	0	1	1	1	0	1	1	0	0	1	1	0	1

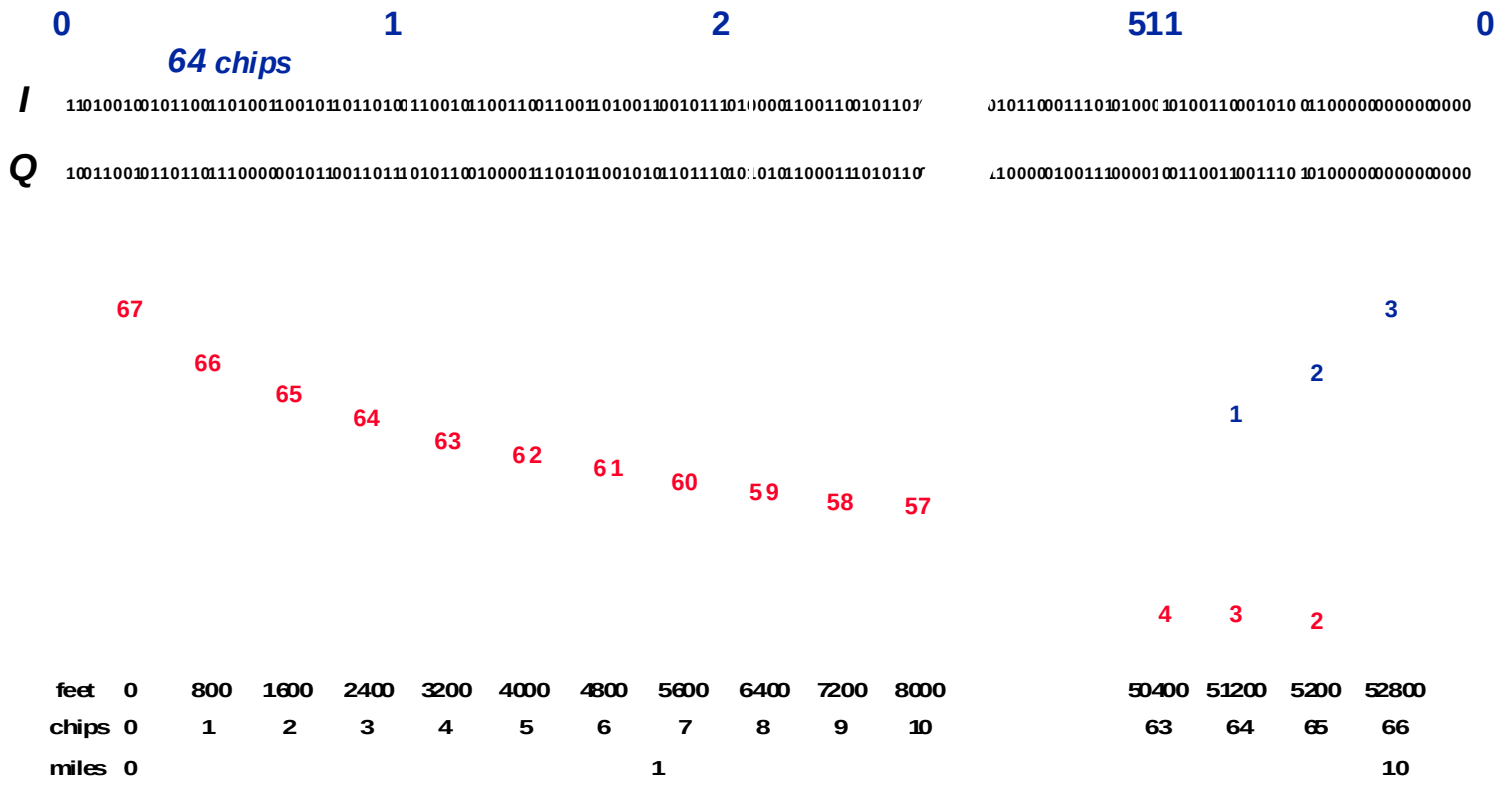
- A PN string, when compared to an identical *shifted* copy of itself, shows no or very little correlation. That is, when XORed with a shifted copy of itself (or with the logical negation of a shifted copy of itself) the result is about half 0's and half 1's

CDMA “Short” and “Long” PN Codes

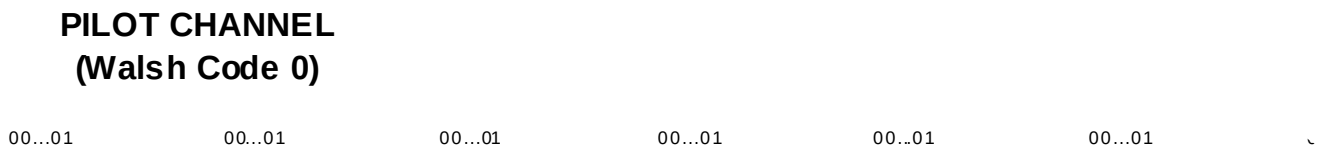
CDMA uses three PN code sequences: two “short” and one “long”

- **The two short PN codes (called “I” and “Q”) are used for quadratu spreading to differentiate between CDMA partitions (sectors/cells) the forward direction**
- **The two short codes are generated by 15-bit PN code generators. The generated strings are 2^{15} -1 bits long plus one zero inserted following the longest string of generated zeroes (32,768); and their cycle period is 26.666… milliseconds (or 75 times every 2 seconds)**
- **The long PN code is used for spreading and data scrambling/ randomization, and to differentiate among mobile stations in the reverse direction.**
- **The long code is generated by a 42-bit PN code generator. The generated string is 2^{42} -1 with no zero inserted (about 4.4 trillion) bits long; and its cycle period is approximately 41 days, 10 hours, 12 minutes and 19.4 seconds.**
- **The three CDMA PN codes are synchronized to the beginning of system time (January 6, 1980 at 00:00:00 hours)**

Sector / Cell Identification



Pilot Channel Acquisition



- The mobile station starts generating the I and Q PN short sequences by itself and correlating them with the received composite signal at every possible offset
  - In less that 15 seconds (typically 2 to 4 seconds) all possibilities (32,768) are checked
  - The mobile station remembers the offsets for which it gets the best correlation (where the E_b/N_0 is the best
- The mobile station locks on the best pilot (at the offset that results in the best E_b/N_0), and identifies the pattern defining the start of the short sequences (a '1' that follows fifteen consecutive '0's)
- Now the mobile station is ready to start de-correlating with Walsh code 32 to extract the Sync Channel (next section)

Sync Channel

- Used to provide essential system parameters
- Used during system acquisition stage
- The bit rate is 1200 bps
- The Sync channel has a frame duration of 26 $^{2}/_{3}$ ms
  - this frame duration matches the period of repetition of the PN Short Sequences
  - this simplifies the acquisition of the Sync Channel once the Pilot Channel has been acquired
- The Mobile Station re-synchronizes at the end of every call





(Acquired Pilot)



Sync Channel

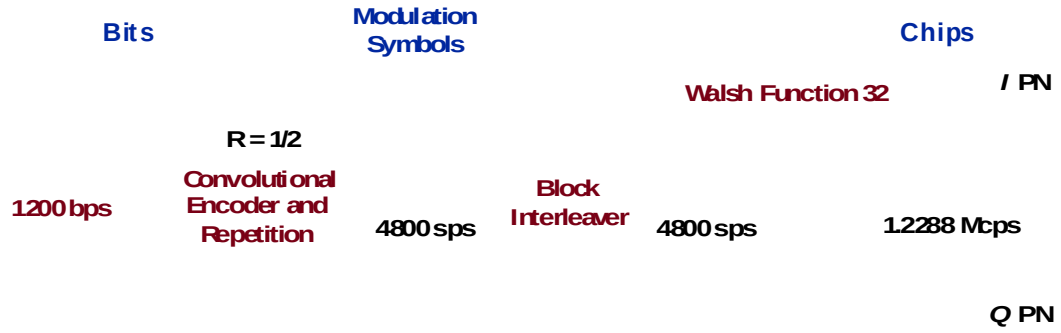


The **Pilot** channel carries no data, therefore it has **no frames**.

The **Sync** channel uses **26 $^{2}/_{3}$ ms** frames.

All other forward and reverse code channels use **20 ms** frames.

Sync Channel Generation



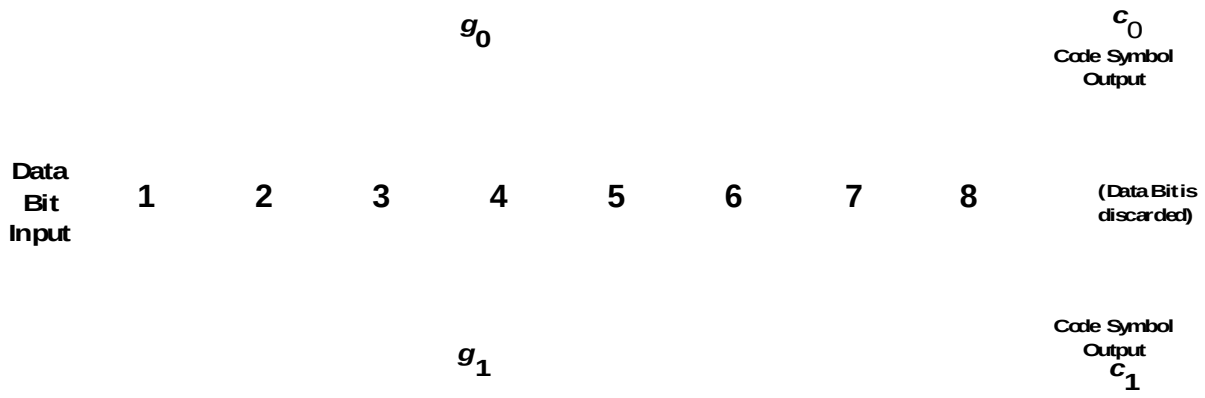
- There are 32 bits (1200 bps x 0.02666... second) in one Sync Channel frame
- The Rate 1/2 convolutional encoder doubles the bit rate, and the resulting 0s and 1s are now called “code symbols”
  - there are 64 code symbols in a Sync Channel frame
- The repetition process doubles the rate again, and each repetition of a code symbol is now called a “modulation symbol”
  - there are 128 modulation symbols in a Sync Channel frame
- Four copies of Walsh code #32 are used to spread each modulation symbol, resulting in a x256 rate increase; the resulting 0s and 1s are now called “ chips”
  - there are 32,768 chips in a Sync Channel frame (1024 chips per original bit)

Sync Channel Modulation Parameters

Data Rate	1200	Bits Per Second
PN Chip Rate	1.2288	Mega Chips Per Second
Code Rate	1 / 2	Bits Per Code Symbol
Code Repetition	2	Modulation Symbols* Per Code Symbol
Modulation Symbol Rate	4800	Symbols Per Second
PN Chips / Modulation Symbol	256	PN Chips / Modulation Symbol
PN Chips / Bit	1024	PN Chips / Bit

* In the CDMA Forward Code Channels, each repetition of a code symbol is called a modulation symbol

Rate 1/2, K=9 Convolutional Encoding



- Symbols generated as the information bits transit through the encoder, are related to all the bits currently in the register
- Each information bit contributes to multiple generated symbols
- This pattern of inter-relationships helps detect and correct errors
- The length of shift register plus 1 is called the “constraint length” of the convolutional encoder (K=9 in this case)
  - The longer the register, the better this scheme can correct bursty errors
  - Reduces power required to achieve same accuracy as without coding
- Here, two symbols are generated for every bit input (Rate 1/2)

Step-by-Step Example (Rate 1/2, k=4)

b_6 b_5 b_4 b_3 b_2 b_1

b_6 b_5 b_4 b_3 b_2 b_1

Step-by-Step Example (Rate 1/2, k=4) – cont.

b_6 b_5 b_4 b_3 b_2 b_1 b_6 b_5 b_4 b_3 b_2 b_1

$$C_{0,4} = b_4 + b_3 + b_1$$

b_6 b_5 b_4 b_3 b_2 b_1

$$C_{1,4} = b_4 + b_2 + b_1$$

Step-by-Step Example (step 1)

$b_3\ b_2\ b_1$? ? ?

$C_{0,1}$

$b_3\ b_2$ **b_1** **?** **?** **?**

$C_{1,1}$

$b_3\ b_2$ b_1 ? ? ?

Step-by-Step Example (step 2)

b_3 b_2 b_1 ? ?

$C_{0,2}$ $C_{0,1}$

b_3 **b_2** **b_1** **?** **?**

$C_{1,2}$ $C_{1,1}$

b_3 b_2 b_1 ? ?

Step-by-Step Example (step 3)

b_3 b_2 b_1 ?

$C_{0,3}$ $C_{0,2}$ $C_{0,1}$

b_3 b_2 b_1 ?

$C_{1,3}$ $C_{1,2}$ $C_{1,1}$

b_3 b_2 b_1 ?

An Even Simpler Convolutional Encode





Convolutional Encoder

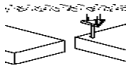
OUT

IN

Paging Channels

Paging Channel

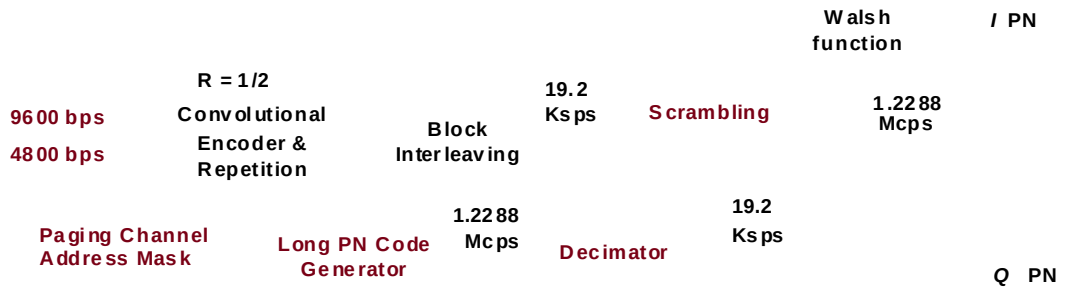




Used by the base station to transmit system overhead information and mobile station-specific messages.

- Up to seven paging channels can be supported on a single CDMA frequency assignment
- Channel 1 (Walsh function 1) is the Primary Paging Channel
- Additional Paging Channels use Walsh functions 2 through 7
- Unused paging channels can be used as Forward Traffic Channels
- Two rates are supported: 9600 and 4800 bps (PRAT parameter in the Sync Channel Message)
- A single 9600 bps Paging Channel can support about 180 pages per second

Paging Channel Generation

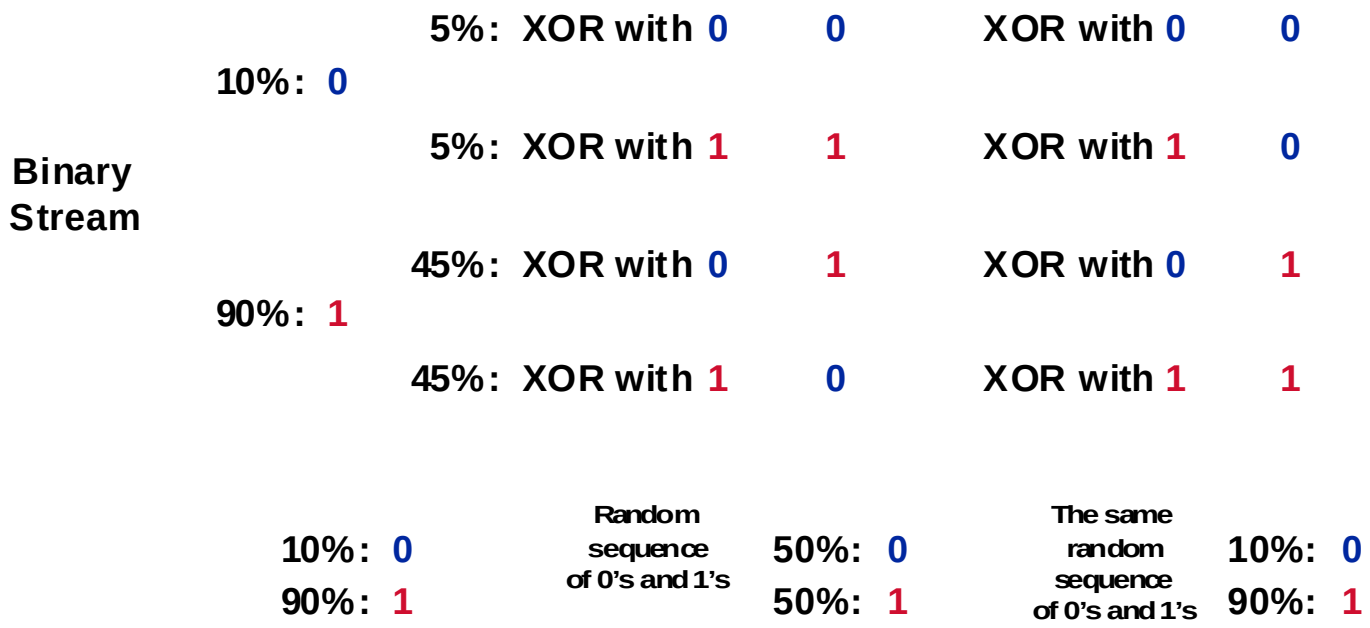


Paging Channel Modulation Parameters

Data Rate	9600	4800	Bits Per Second
PN Chip Rate	1.2288	1.2288	Mega Chips Per Second
Code Rate	1/2	1/2	Bits Per Code Symbol
Code Repetition	1	2	Modulation symbols * per code symbol
Modulation Symbol Rate	19200	19200	Code Symbols Per Second
PN Chips / Modulation Symbol	64	64	PN Chips Per Modulation Symbol
PN Chips / Bit	128	256	PN Chips Per Bit

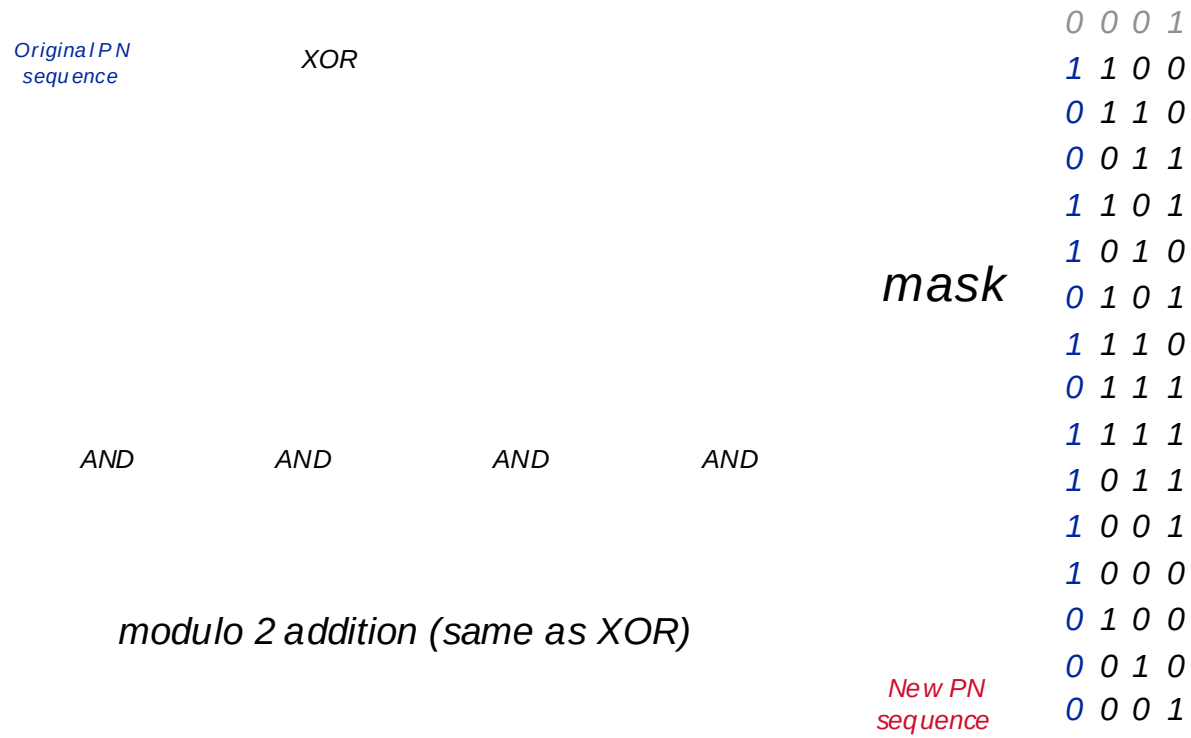
* Each repetition of a code symbol is a modulation symbol

Data Scrambling (Data Randomization



Exactly the same random sequence of 0's and 1's must be used at both ends. Perfect synchronization is required.

Masks


Original PN sequence
XOR
mask
AND
AND
AND
AND
modulo 2 addition (same as XOR)
New PN sequence
0 0 0 1
1 1 0 0
0 1 1 0
0 0 1 1
1 1 0 1
1 0 1 0
0 1 0 1
1 1 1 0
0 1 1 1
1 1 1 1
1 0 1 1
1 0 0 1
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1

Masks (Example 1)

Notice that not every bit in the mask needs to be ANDed with the corresponding register bit, but not ANDing a particular mask bit with the corresponding register bit has the same effect as making this bit ‘0’ and ANDing it anyway!

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

without mask:	1	0	0	1	1	0	1	0	1	1	1	1	0	0	0
with mask 1101:	0	1	1	1	1	0	0	0	1	0	0	1	1	0	1

Masks (Example 2)

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

without mask:	1	0	0	1	1	0	1	0	1	1	1	1	0	0	0
with mask 1100:	0	1	0	0	1	1	0	1	0	1	1	1	1	0	0

Paging Channel Scrambling

A Paging Channel is scrambled by the long code, offset by a mask constructed as follows:

41	29	28	24	23	21	20	9	8	0
1100011001101		00000		PCN		000000000000			PILOT_PN

Where:

PCN is the Paging Channel Number, and

PILOT_PN is the Pilot short PN code offset index