CDMA TECHNOLOGY

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

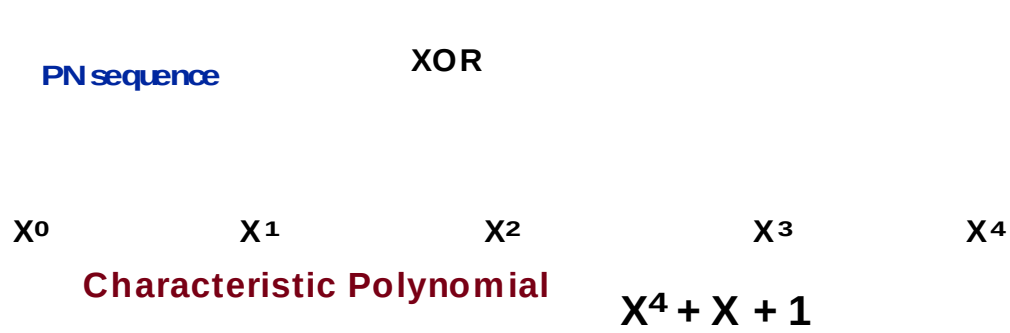
p_1 p_2 p_3 p_4

p_4 p_2 p_3 p_4

$p5 = p_1 + p_4$

p_4 p_5 p_2 p_3

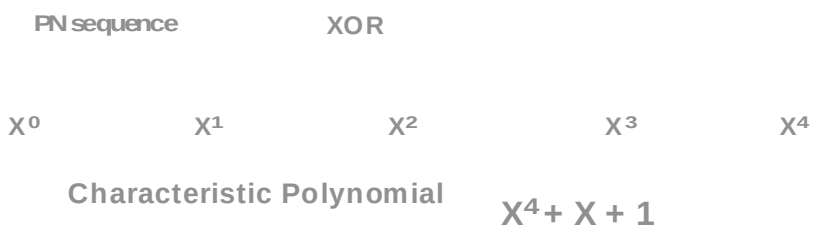
PN Sequence Generation



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

- The PN sequences are deterministic and periodic.
  - The length of the generated string is $2^n - 1$, where "n" is the number of elements in the register
  - The number of zeroes in the sequence is equal to the number of ones minus 1
  - The beginning of the sequence is the "1" that follows n-1 zeroes; the final 0 of the sequence corresponds to the state of the register when all its elements except the last one contain a zero

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

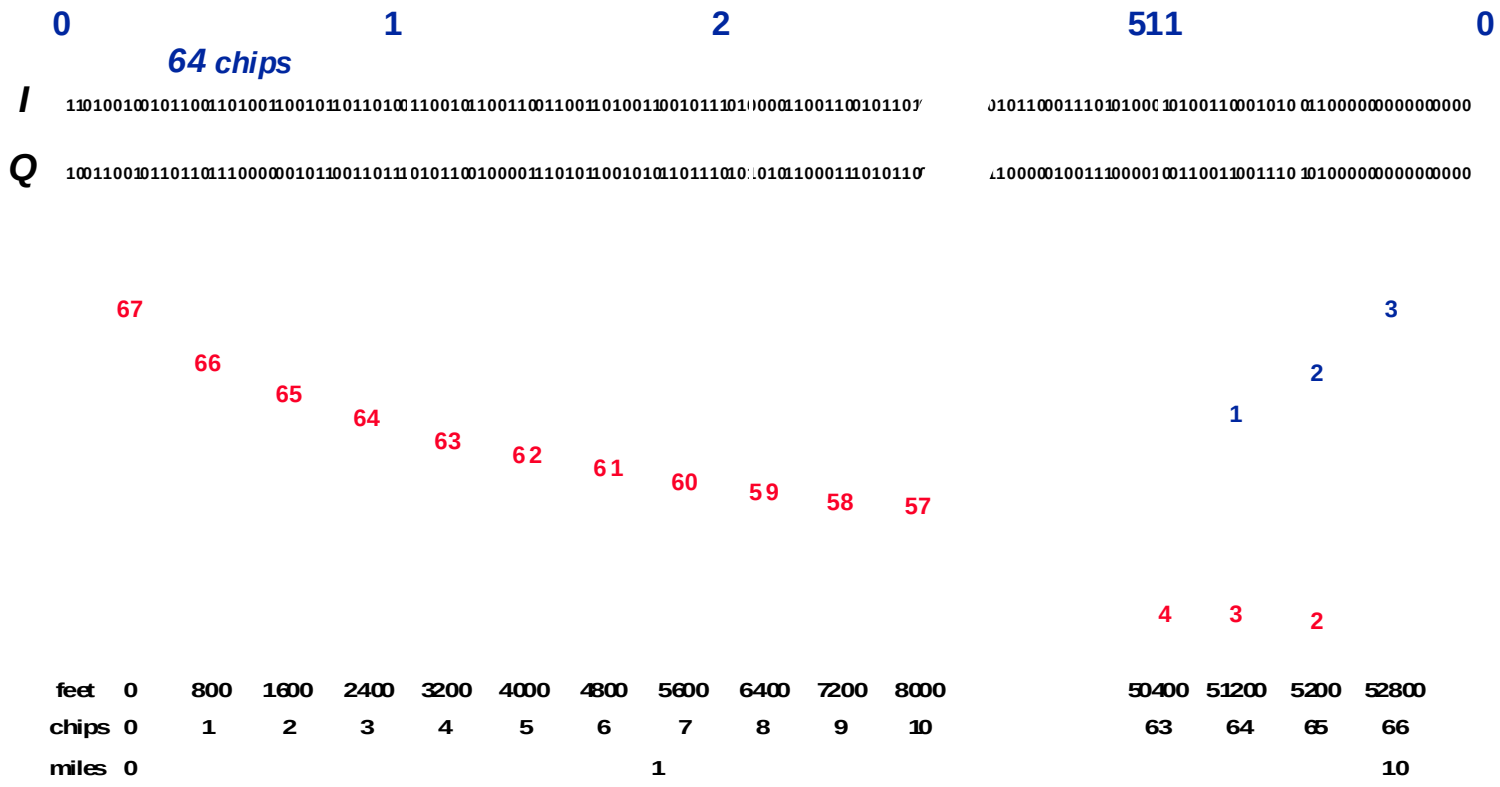
- A PN string, when compared to an identical *shifted* copy of itself, shows no or very little correlation. That is, when XORed with a shifted copy of itself (or with the logical negation of a shifted copy of itself) the result is about half 0's and half 1's

CDMA “Short” and “Long” PN Codes

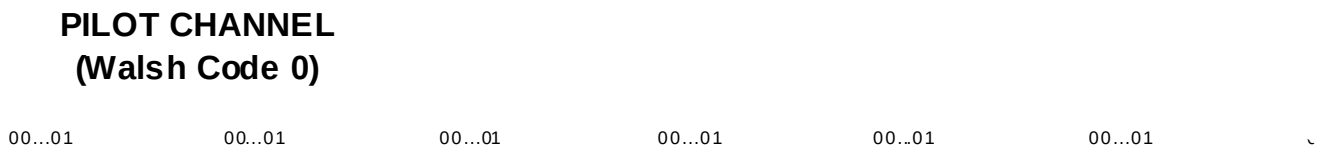
CDMA uses three PN code sequences: two “short” and one “long”

- **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

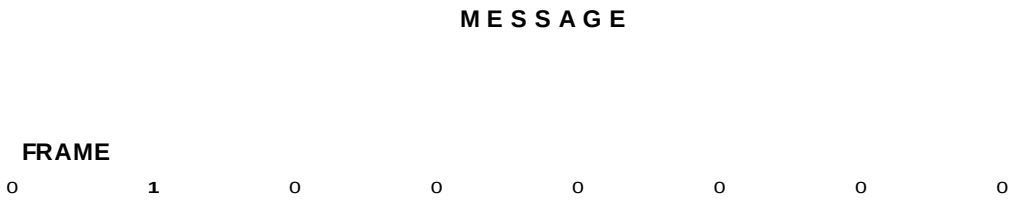


Pilot Channel Acquisition



- 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)

Frames and Messages



FRAME

- Logical unit of transmission
- Fixed length
  - no need for length info
- Each frame includes one or more overhead bits in addition to the “payload” of information bits
  - these overhead bits define the structure of the frame

MESSAGE

- Logical unit of information
- Variable length
  - must include length info
- A message is broken into small pieces that can fit in the payload portion of successive frames
  - one frame overhead bit could be used to identify the initial segment of a message

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

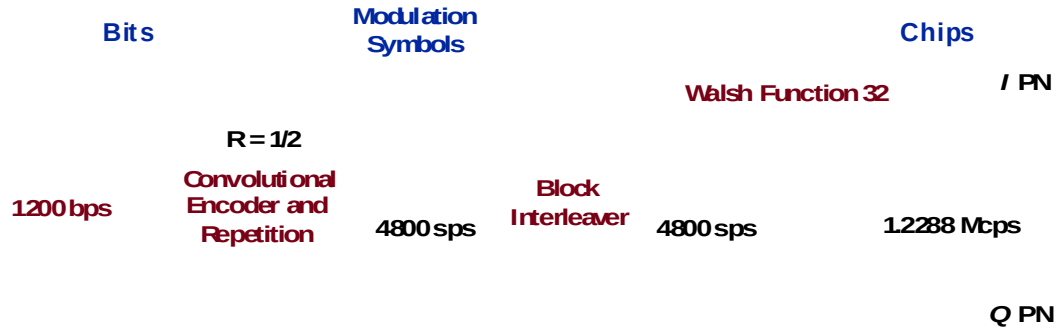


The **Pilot** channel carries no data, therefore it has **no frames**.

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

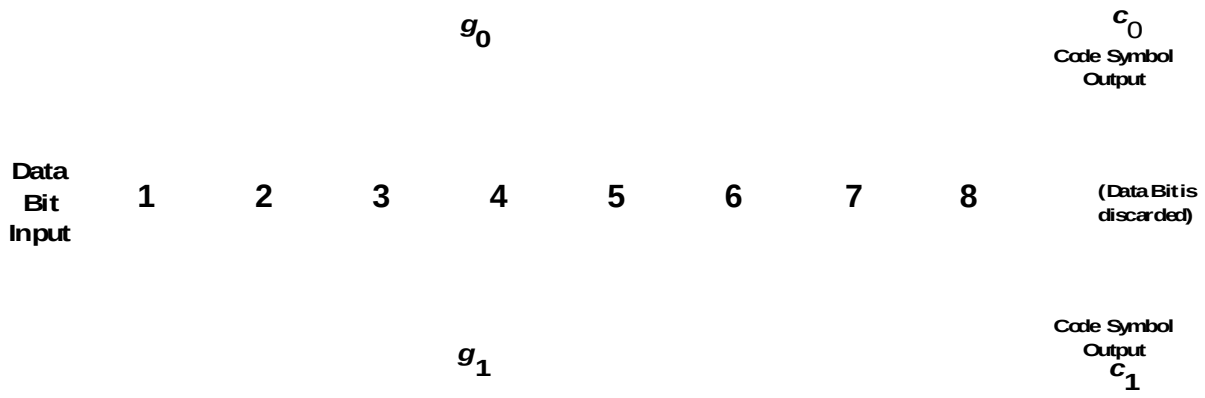
* In the CDMA Forward Code Channels, each repetition of a code symbol is called a modulation symbol

Convolutional Encoder

OUT

IN

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 ?

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

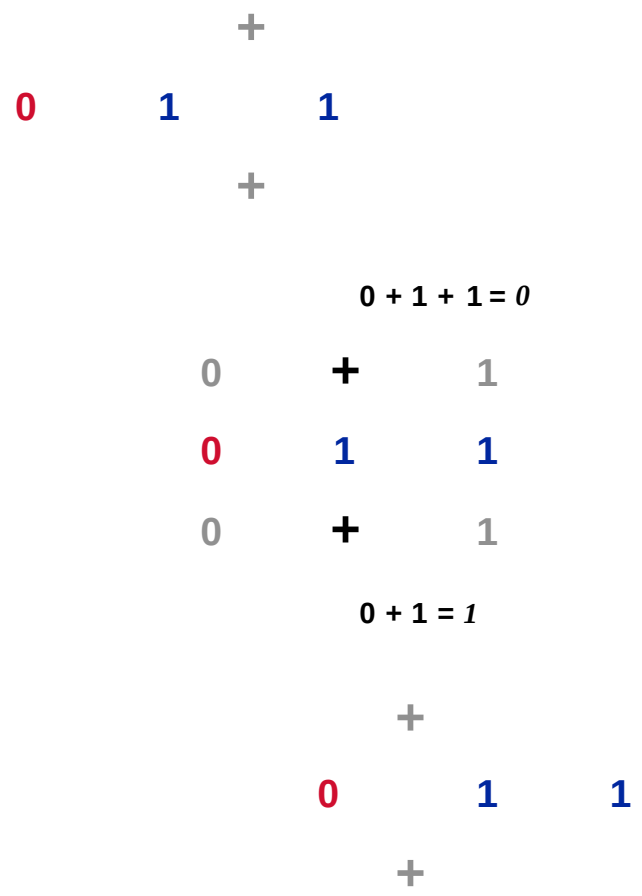
b_3 b_2 b_1 ?

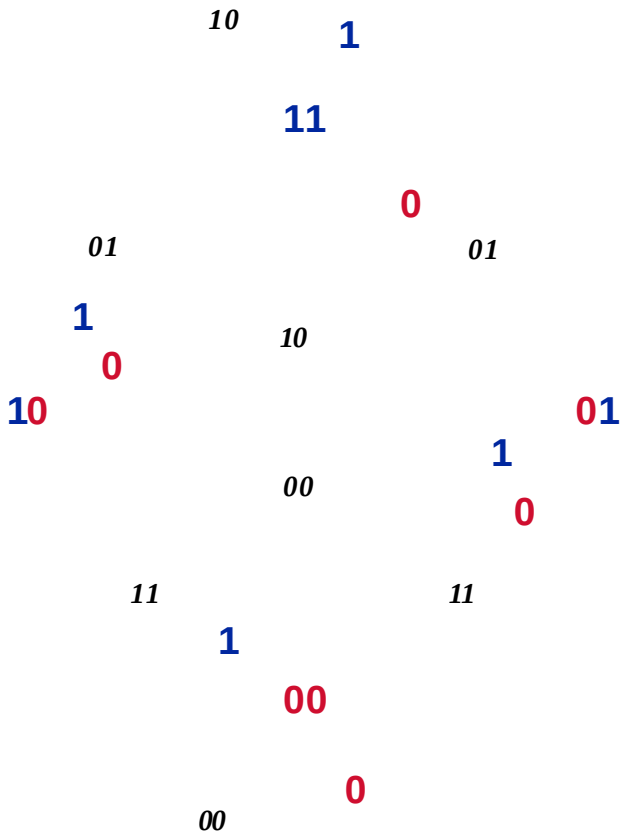
An Even Simpler Convolutional Encode



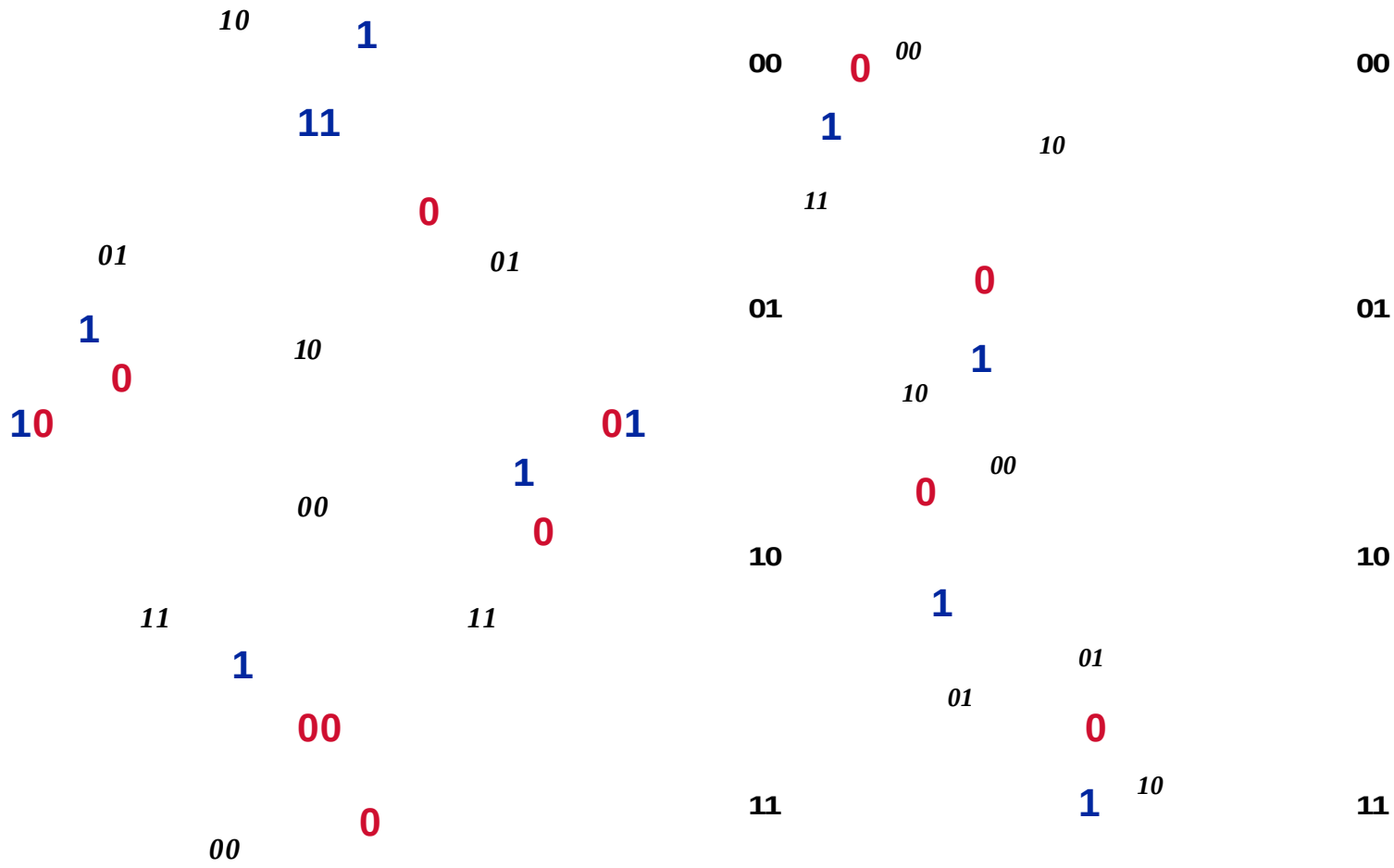


State Diagram

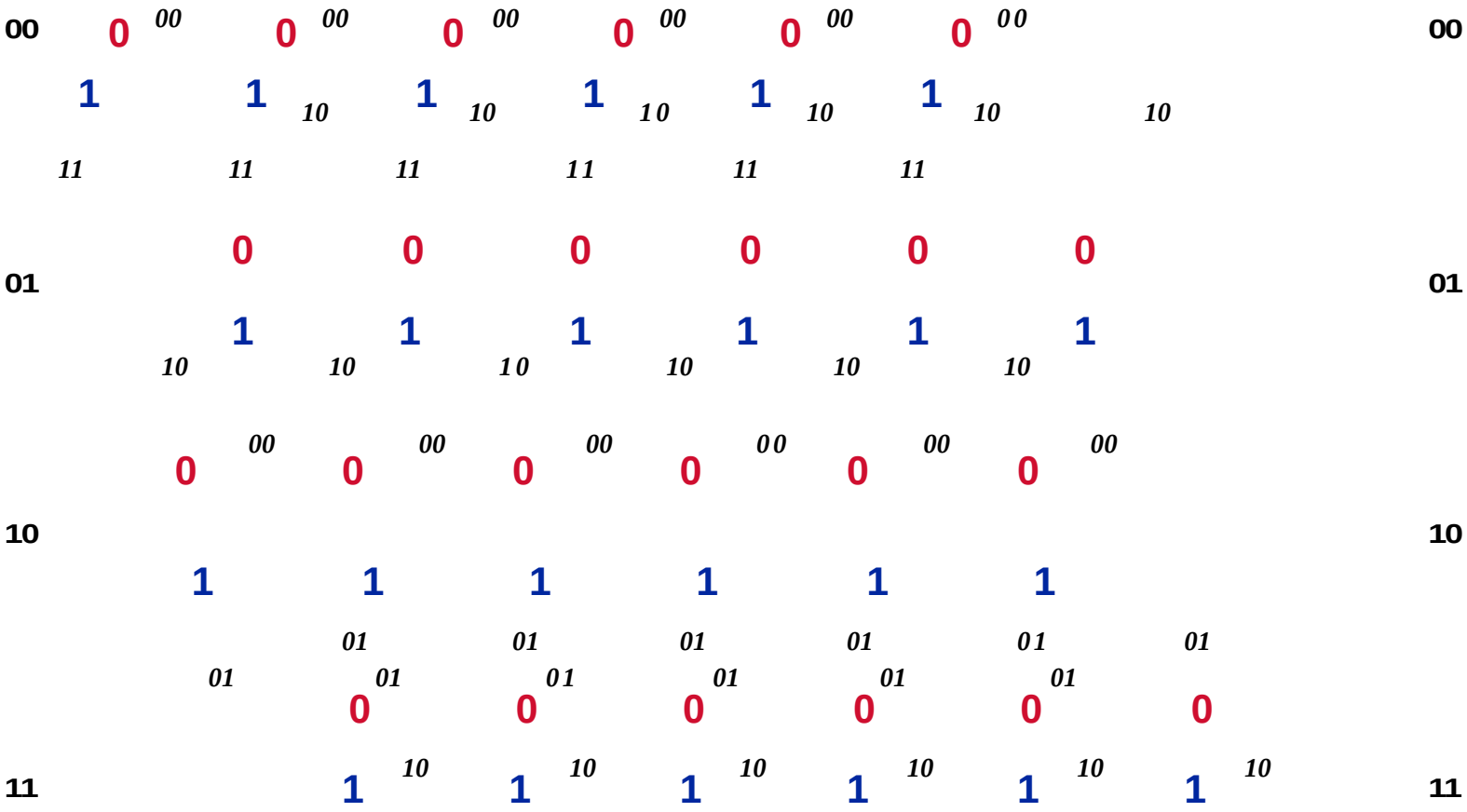




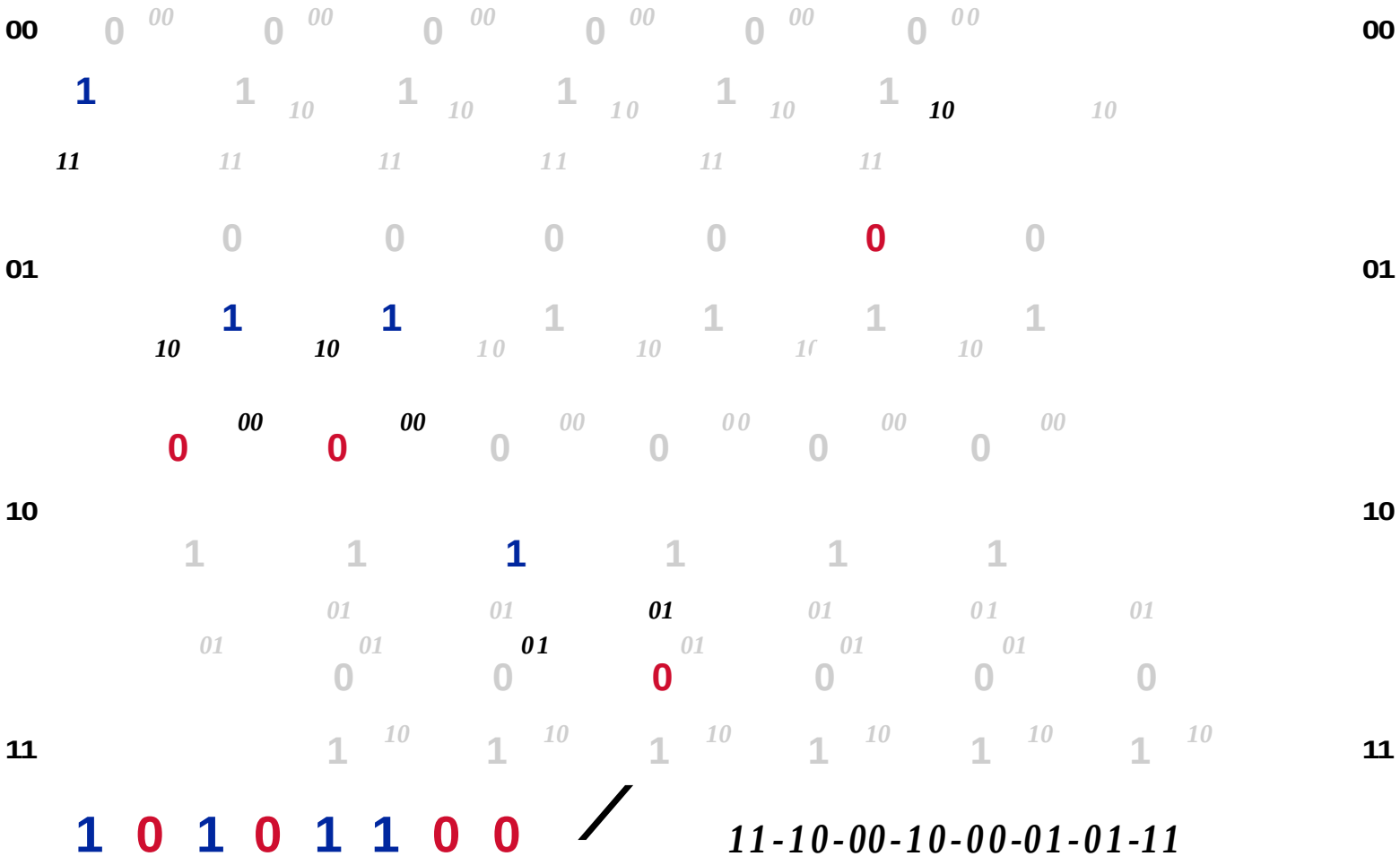
State Diagram as a Binary Tree



Trellis Diagram



Code Words



Hamming and Free Distance

0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0
0 0 0 0 1 0 0 0
0 0 0 0 1 1 0 0
• • •
1 0 1 0 1 1 0 0
• • •
1 1 1 1 0 1 0 0
1 1 1 1 1 0 0 0
1 1 1 1 1 1 0 0

$2^6 = 64$

one-to-one

valid code words: $2^6 = 64$

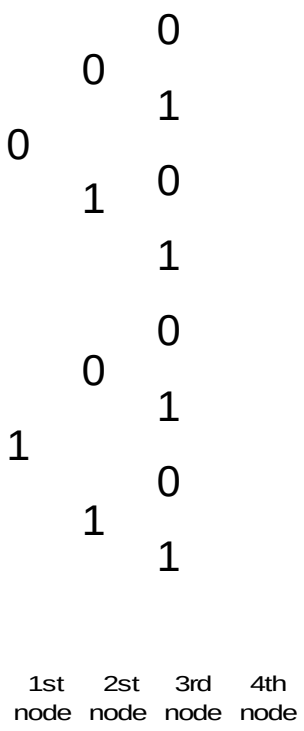
11-10-00-10-00-01-01-11

10-11-00-1 1-00-0 0-01-11

number of possible received 16-bit sequences:

$2^{16} = 65{,}536$

Viterbi Decoder



- Integrated circuit that implements the maximum likelihood algorithm devised by Andrew Viterbi in 1967
- Operates based on determining the correlation between the sections of the received signal and each one of the valid codes
- The Viterbi decoder uses a search tree to continuously calculate the Hamming distance between the received and valid codes. If an error path is detected, the decoder goes back to the previous node and tries the alternative path
- The code that generates the maximal amount of energy has the greater probability of being the one that was transmitted

Sync Channel Block Interleaver (Input Matrix)

1	9	17	25	33	41	49	57
1	9	17	25	33	41	49	57
2	10	18	26	34	42	50	58
2	10	18	26	34	42	50	58
3	11	19	27	35	43	51	59
3	11	19	27	35	43	51	59
4	12	20	28	36	44	52	60
4	12	20	28	36	44	52	60
5	13	21	29	37	45	53	61
5	13	21	29	37	45	53	61
6	14	22	30	38	46	54	62
6	14	22	30	38	46	54	62
7	15	23	31	39	47	55	63
7	15	23	31	39	47	55	63
8	16	24	32	40	48	56	64
8	16	24	32	40	48	56	64

Sync Channel Block Interleaver (Output Matrix)

assume that a burst of noise affects these symbols

1	3	2	4	1	3	2	4
33	35	34	36	33	35	34	36
17	19	18	20	17	19	18	20
49	51	50	52	49	51	50	52
9	11	10	12	9	11	10	12
41	43	42	44	41	43	42	44
25	27	26	28	25	27	26	28
57	59	58	60	57	59	58	60
5	7	6	8	5	7	6	8
37	39	38	40	37	39	38	40
21	23	22	24	21	23	22	24
53	55	54	56	53	55	54	56
13	15	14	16	13	15	14	16
45	47	46	48	45	47	46	48
29	31	30	32	29	31	30	32
61	63	62	64	61	63	62	64

Sync Channel Block Interleaver Restored

1	9	17	25	33	41	49	57
1	9	17	25	33	41	49	57
2	10	18	26	34	42	50	58
2	10	18	26	34	42	50	58
3	11	19	27	35	43	51	59
3	11	19	27	35	43	51	59
4	12	20	28	36	44	52	60
4	12	20	28	36	44	52	60
5	13	21	29	37	45	53	61
5	13	21	29	37	45	53	61
6	14	22	30	38	46	54	62
6	14	22	30	38	46	54	62
7	15	23	31	39	47	55	63
7	15	23	31	39	47	55	63
8	16	24	32	40	48	56	64
8	16	24	32	40	48	56	64

Sync Channel Structure

1200 bps

80 ms, 96 bits

Sync Channel Superframe

Sync Channel Superframe

Sync Channel Frame

26.67 ms
32 bits

SOM ***Sync Channel Frame Body***

31 bits

1 0 0 0 0 0 0

Sync Channel Message Capsule **(93 x N_s bits)**

Sync Channel Message

Padding

Sync Channel Message **(8 x MSG_LENGTH)**

as required

MSG_LENGTH

8 bits

Message Body

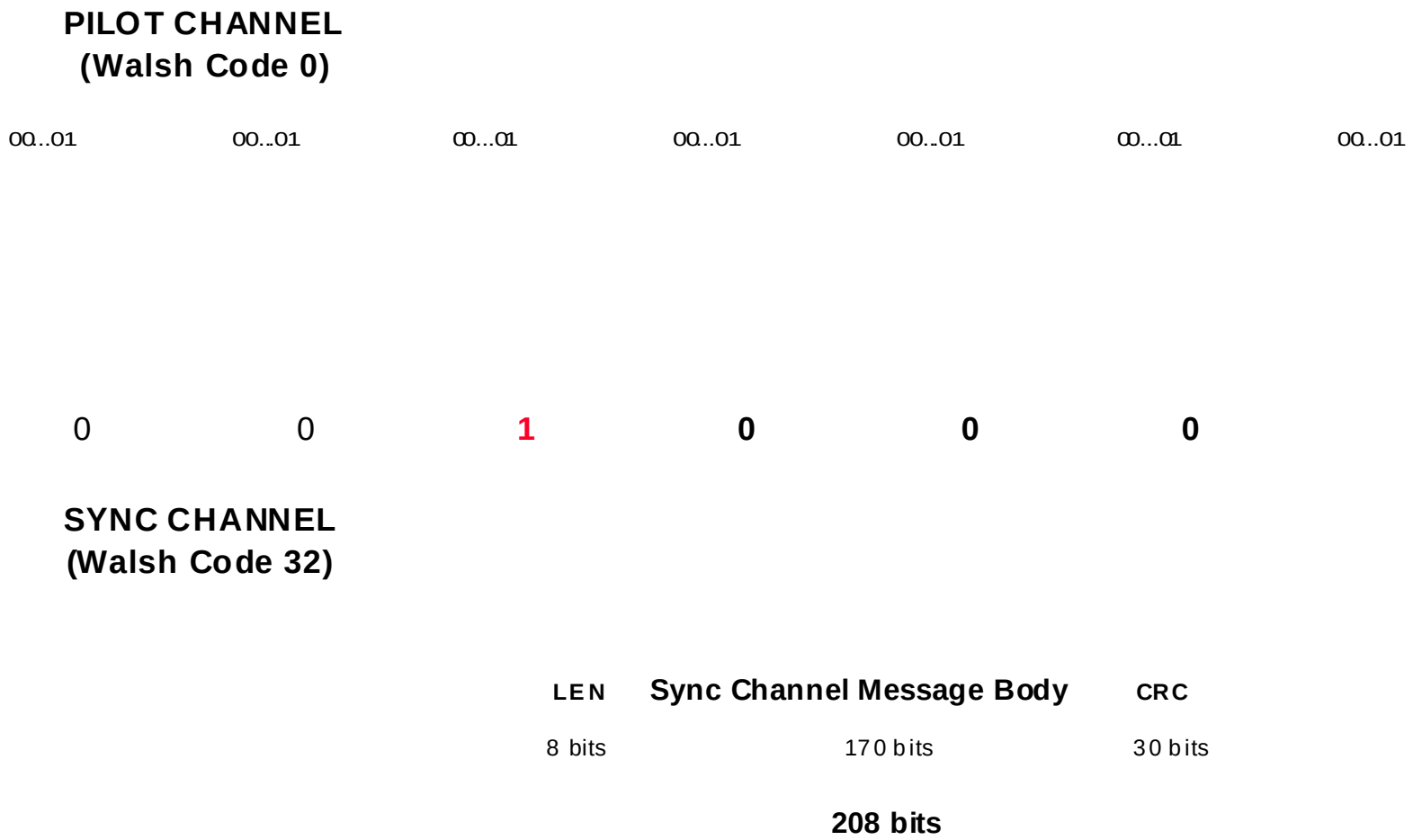
2-1146 bits

CRC

30 bits

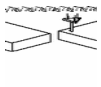
Ns = Number of Sync Channel Superframes needed for message transmission

Sync Channel Acquisition



Sync Channel Message Body Format

- When the Base Station sends a Sync Channel Message, it uses the fixed-length message format illustrated here

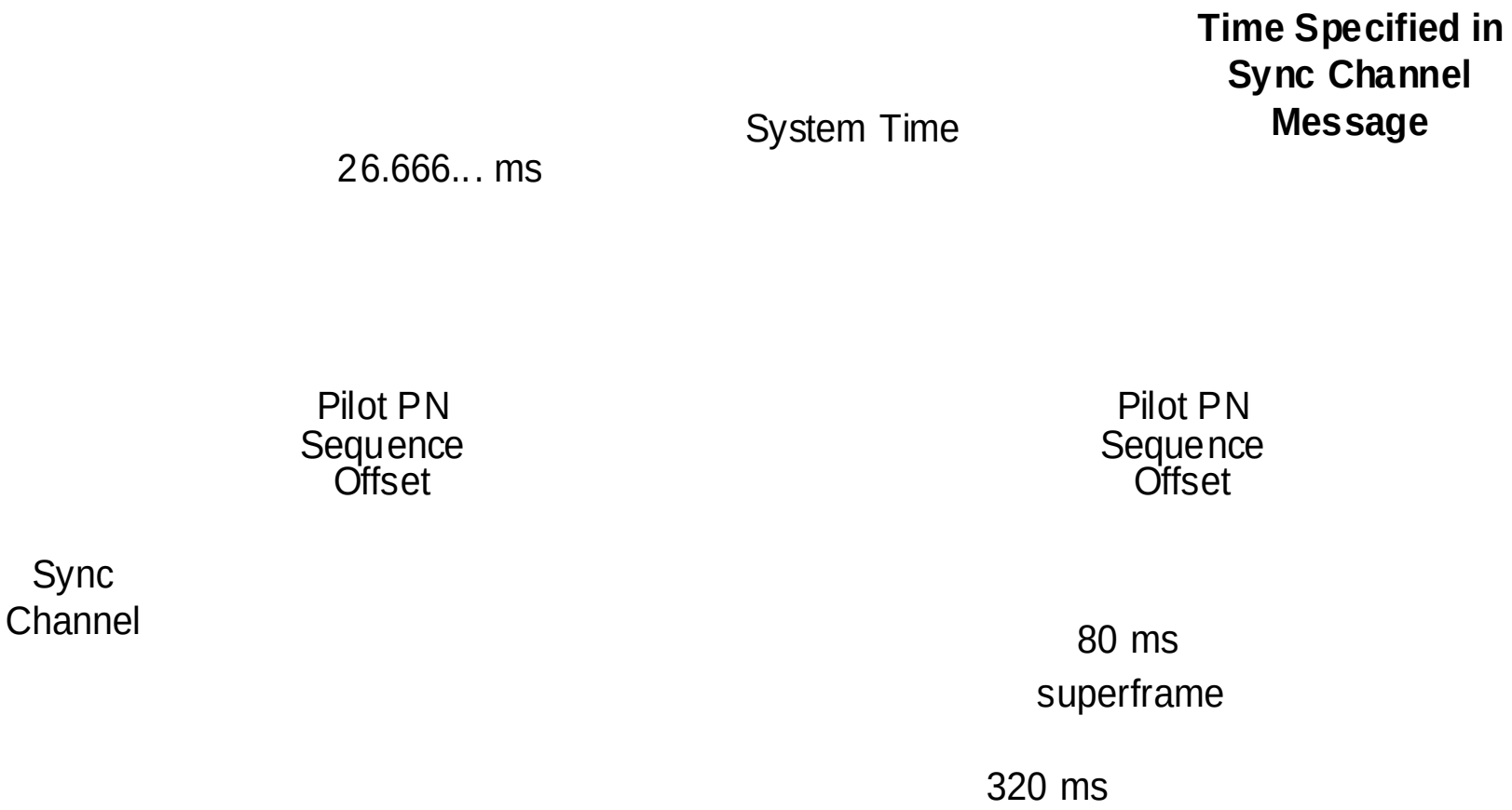


SYNC



Field	Length (bits)
MSG_TYPE (‘00000001’)	**8**
P_REV	**8**
MIN_PREV	**8**
SID	**15**
NID	**16**
PILOT_PN	**9**
LC_STATE	**42**
SYS_TIME	**36**
LP_SEC	**8**
LTM_OFF	**6**
DAYLT	**1**
PRAT	**2**
CDMA_FREQ	**11**
	Total : 170

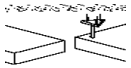
Mobile Station Timing Synchronization



Paging Channels

Paging Channel

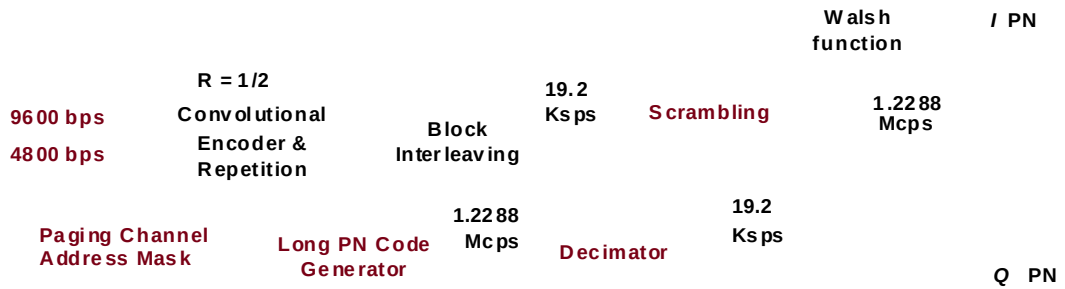




Used by the base station to transmit system overhead information and mobile station-specific messages.

- 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



- There are 192 [96] bits (9600 [4800] bps x 0.020 second) in one Paging Channel frame
- The Rate 1/2 convolutional encoder doubles the bit rate, resulting" 384 [192] code symbols in a Paging Channel frame
- If the 4800 bps rate is used, the repetition process doubles the rate again, so that, at either rate, 384 modulation symbols per Paging Channel frame result
- 384 modulation symbols per frame times 50 frames per second = 19.2 Ksps
- One copy of Walsh code #1 (or #2, ... or #7) is used to spread each modulation symbol. This results in a x64 rate increase to 1.2288 Mcps
  - that is, 24,576 chips per Paging Channel frame, or 128 [256] chips per original bit at 9600 [4800] bps

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

Paging Channel - 9600 bps Block Interleaver (Input Array)

16 Columns

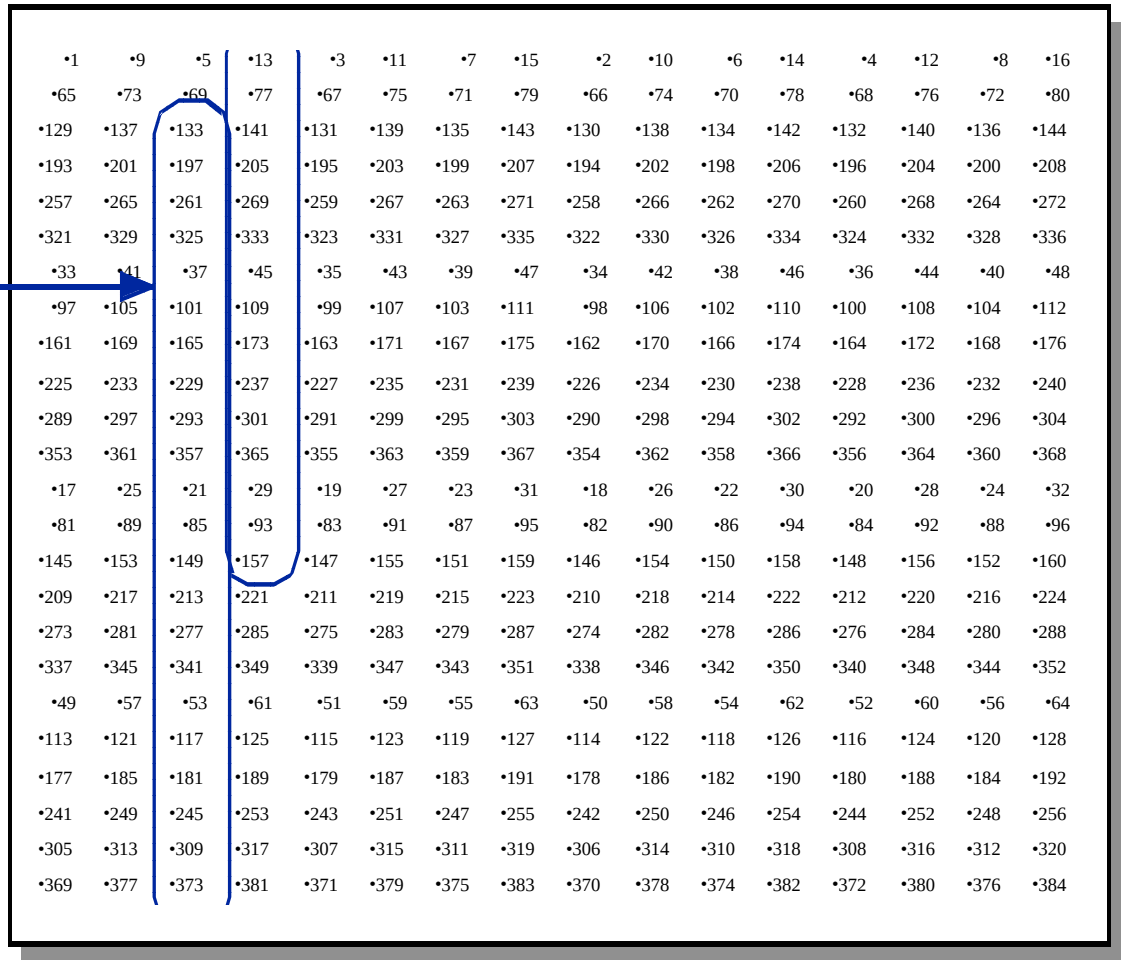
24 rows

1	25	49	73	97	121	145	169	193	217	241	265	289	313	337	361
2	26	50	74	98	122	146	170	194	218	242	266	290	314	338	362
3	27	51	75	99	123	147	171	195	219	243	267	291	315	339	363
4	28	52	76	100	124	148	172	196	220	244	268	292	316	340	364
5	29	53	77	101	125	149	173	197	221	245	269	293	317	341	365
6	30	54	78	102	126	150	174	198	222	246	270	294	318	342	366
7	31	55	79	103	127	151	175	199	223	247	271	295	319	343	367
8	32	56	80	104	128	152	176	200	224	248	272	296	320	344	368
9	33	57	81	105	129	153	177	201	225	249	273	297	321	345	369
10	34	58	82	106	130	154	178	202	226	250	274	298	322	346	370
11	35	59	83	107	131	155	179	203	227	251	275	299	323	347	371
12	36	60	84	108	132	156	180	204	228	252	276	300	324	348	372
13	37	61	85	109	133	157	181	205	229	253	277	301	325	349	373
14	38	62	86	110	134	158	182	206	230	254	278	302	326	350	374
15	39	63	87	111	135	159	183	207	231	255	279	303	327	351	375
16	40	64	88	112	136	160	184	208	232	256	280	304	328	352	376
17	41	65	89	113	137	161	185	209	233	257	281	305	329	353	377
18	42	66	90	114	138	162	186	210	234	258	282	306	330	354	378
19	43	67	91	115	139	163	187	211	235	259	283	307	331	355	379
20	44	68	92	116	140	164	188	212	236	260	284	308	332	356	380
21	45	69	93	117	141	165	189	213	237	261	285	309	333	357	381
22	46	70	94	118	142	166	190	214	238	262	286	310	334	358	382
23	47	71	95	119	143	167	191	215	239	263	287	311	335	359	383
24	48	72	96	120	144	168	192	216	240	264	288	312	336	360	384

- The 384 modulation symbols in a frame are input into a 24 by 16 block interleaver array (read down by columns, from left to right)
- The array represents a 20 ms interval worth of information

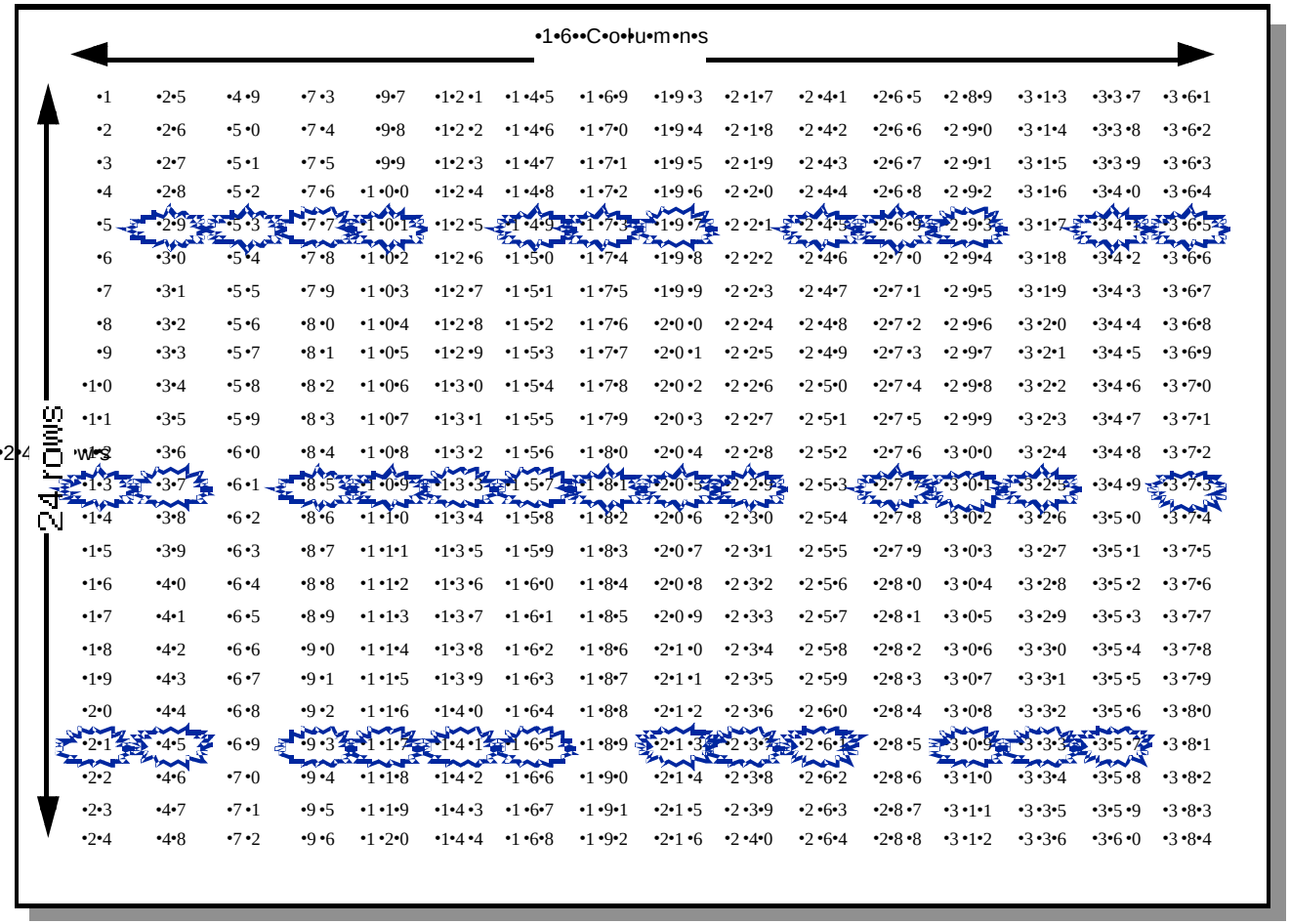
9600 bps Block Interleaver (Output Array)

- *Assume that a*
- *burst of noise*
- *damages all*
- *these bits*

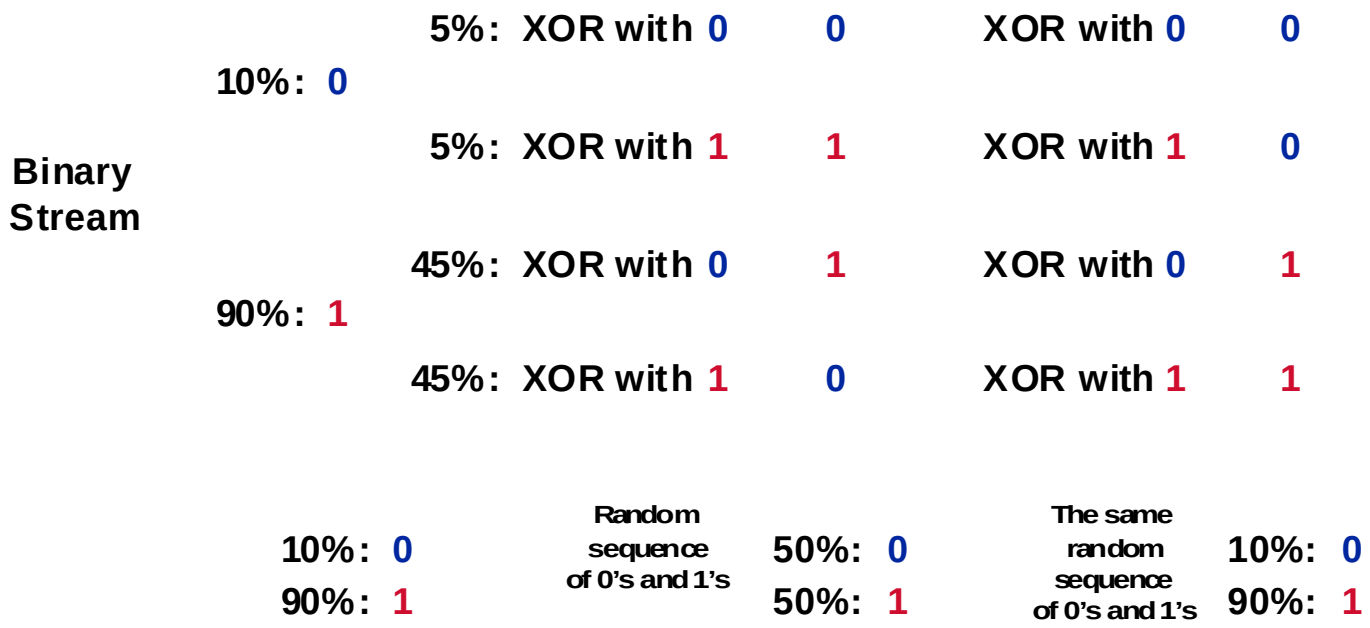


•1	•9	•5	•13	•3	•11	•7	•15	•2	•10	•6	•14	•4	•12	•8	•16
•65	•73	•69	•77	•67	•75	•71	•79	•66	•74	•70	•78	•68	•76	•72	•80
•129	•137	•133	•141	•131	•139	•135	•143	•130	•138	•134	•142	•132	•140	•136	•144
•193	•201	•197	•205	•195	•203	•199	•207	•194	•202	•198	•206	•196	•204	•200	•208
•257	•265	•261	•269	•259	•267	•263	•271	•258	•266	•262	•270	•260	•268	•264	•272
•321	•329	•325	•333	•323	•331	•327	•335	•322	•330	•326	•334	•324	•332	•328	•336
•33	•41	•37	•45	•35	•43	•39	•47	•34	•42	•38	•46	•36	•44	•40	•48
•97	•105	•101	•109	•99	•107	•103	•111	•98	•106	•102	•110	•100	•108	•104	•112
•161	•169	•165	•173	•163	•171	•167	•175	•162	•170	•166	•174	•164	•172	•168	•176
•225	•233	•229	•237	•227	•235	•231	•239	•226	•234	•230	•238	•228	•236	•232	•240
•289	•297	•293	•301	•291	•299	•295	•303	•290	•298	•294	•302	•292	•300	•296	•304
•353	•361	•357	•365	•355	•363	•359	•367	•354	•362	•358	•366	•356	•364	•360	•368
•17	•25	•21	•29	•19	•27	•23	•31	•18	•26	•22	•30	•20	•28	•24	•32
•81	•89	•85	•93	•83	•91	•87	•95	•82	•90	•86	•94	•84	•92	•88	•96
•145	•153	•149	•157	•147	•155	•151	•159	•146	•154	•150	•158	•148	•156	•152	•160
•209	•217	•213	•221	•211	•219	•215	•223	•210	•218	•214	•222	•212	•220	•216	•224
•273	•281	•277	•285	•275	•283	•279	•287	•274	•282	•278	•286	•276	•284	•280	•288
•337	•345	•341	•349	•339	•347	•343	•351	•338	•346	•342	•350	•340	•348	•344	•352
•49	•57	•53	•61	•51	•59	•55	•63	•50	•58	•54	•62	•52	•60	•56	•64
•113	•121	•117	•125	•115	•123	•119	•127	•114	•122	•118	•126	•116	•124	•120	•128
•177	•185	•181	•189	•179	•187	•183	•191	•178	•186	•182	•190	•180	•188	•184	•192
•241	•249	•245	•253	•243	•251	•247	•255	•242	•250	•246	•254	•244	•252	•248	•256
•305	•313	•309	•317	•307	•315	•311	•319	•306	•314	•310	•318	•308	•316	•312	•320
•369	•377	•373	•381	•371	•379	•375	•383	•370	•378	•374	•382	•372	•380	•376	•384

9600 bps De-interleaving

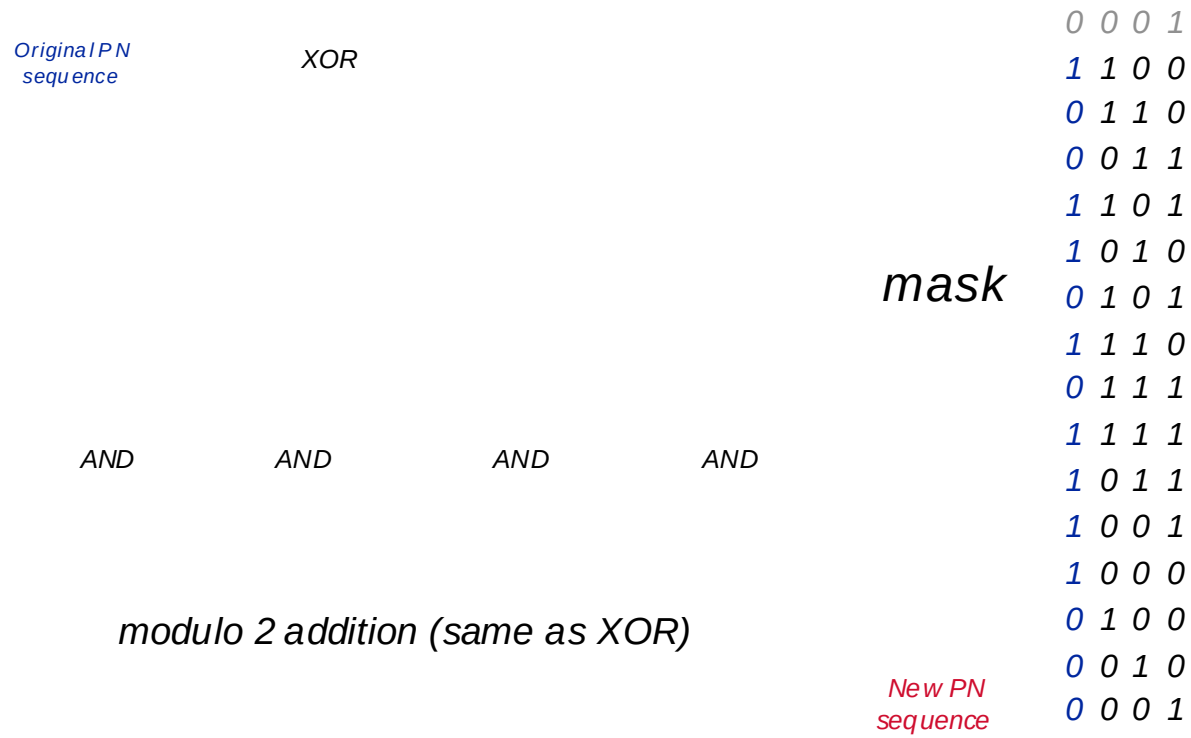


Data Scrambling (Data Randomization



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:

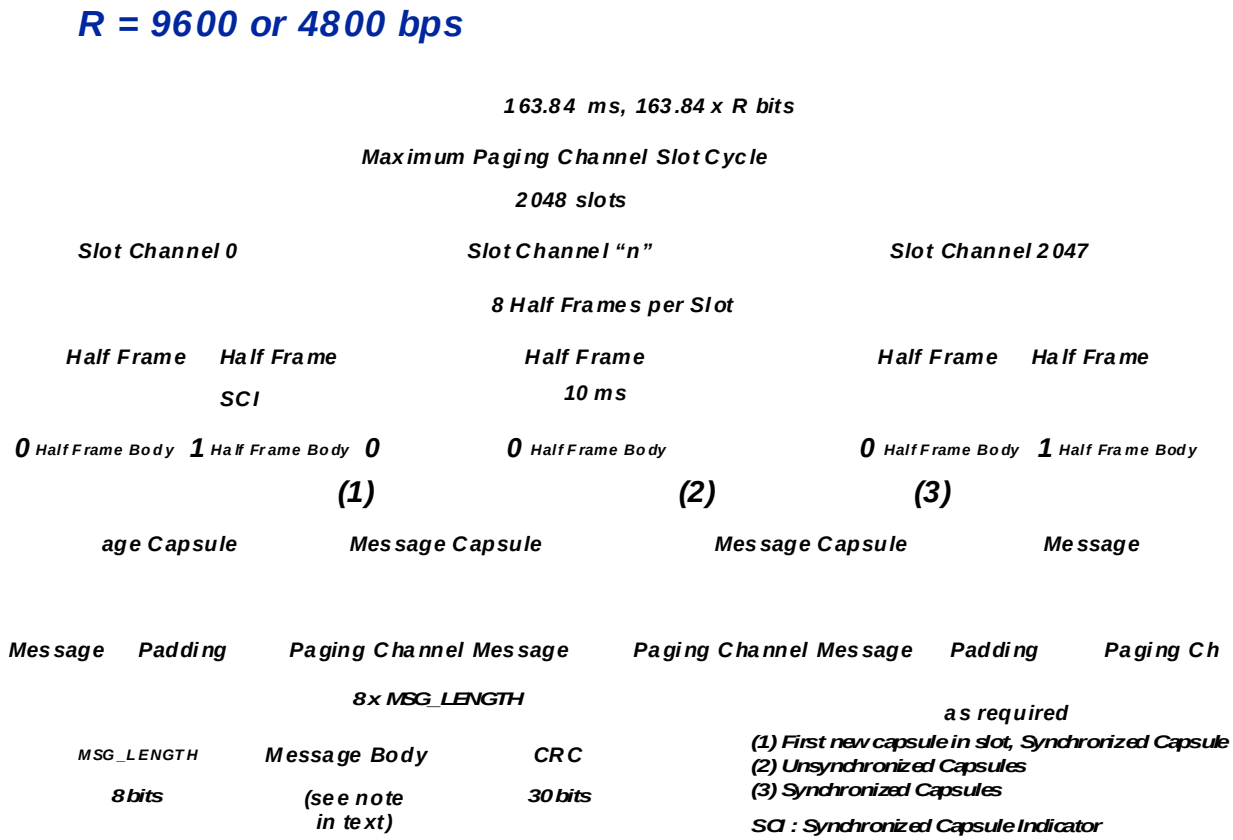
41	29	28	24	23	21	20	9	8	0
1100011001101		00000		PCN		000000000000			PILOT_PN

Where:

PCN is the Paging Channel Number, and

PILOT_PN is the Pilot short PN code offset index

Paging Channel Structure



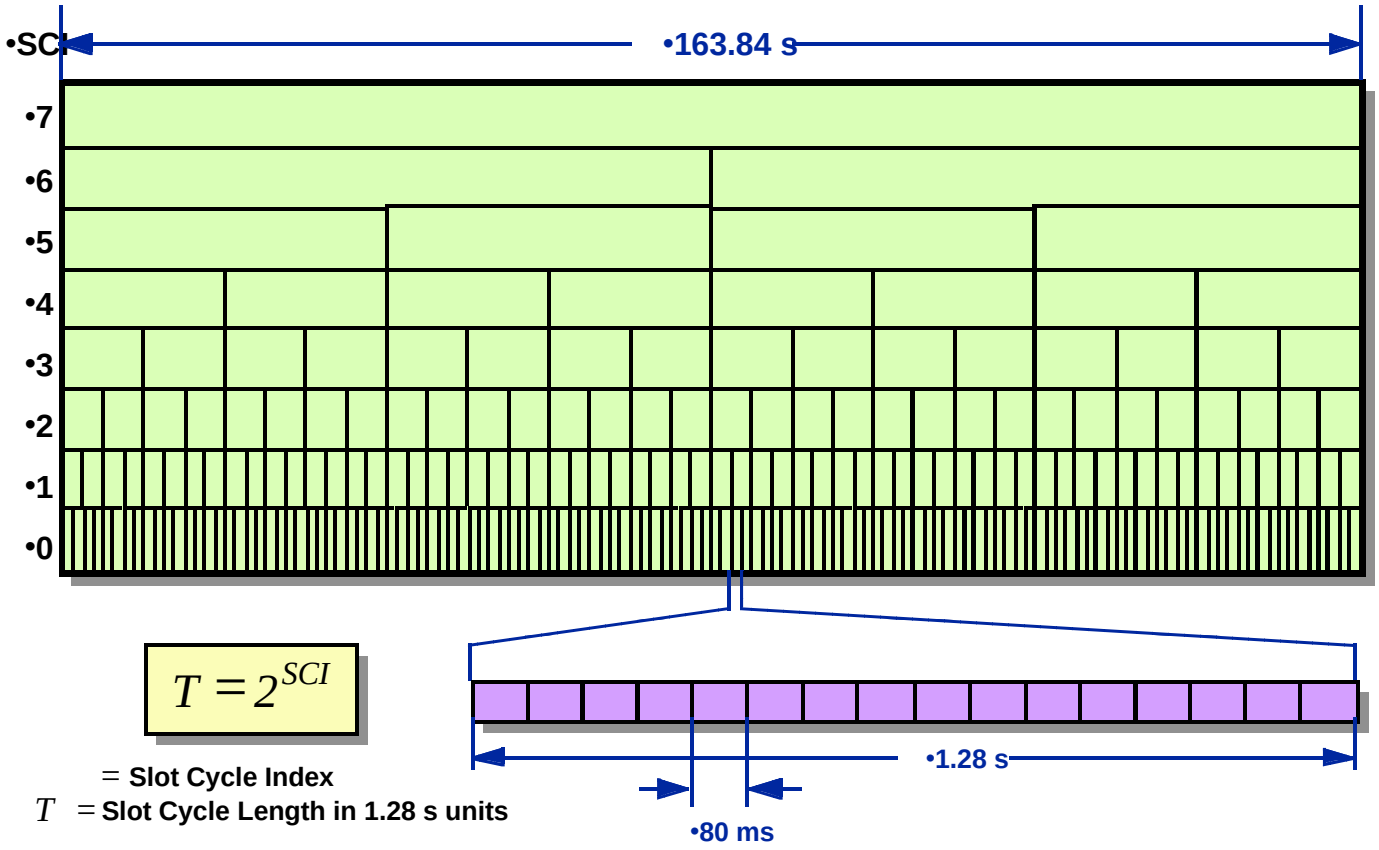
Paging Channel Determination

- CDMA Channel (1.25 MHz band) Determination
  - In a CDMA system with multiple CDMA channels, the mobile station must first determine the CDMA channel to be used
  - For this, it will use a hash function with the mobile station's IMSI and the number of CDMA Channels on which the base station transmits Paging Channels as input
- Paging Channel Determination
  - Then the mobile station must select a Paging Channel among those transmitted in that CDMA channel
  - For this, it will use a hash function with the mobile station's IMSI and the number of Paging Channels on the selected CDMA frequency

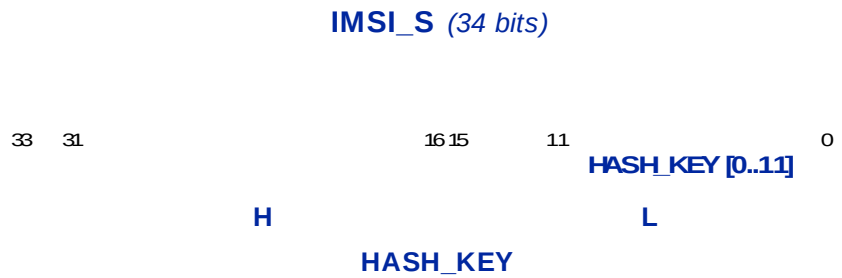
Paging Channel Modes

- Non-Slotted Mode Operation
  - The mobile station continuously monitors the Paging Channel
  - Paging Channel messages can occupy any one of the 2048 slots in the maximum slot cycle
- Slotted Mode Operation
  - Provides scheduled transmission of messages for a specific mobile station
  - Monitoring frequency can range from every 1.28 seconds to once every 163.84 seconds
  - Support is optional and enabled by the mobile station
  - The mobile station can “sleep” or reduce power consumption during non-active states
- The mobile station must indicate the desired operating mode by means of the SLOTTED_MODE field in one certain messages.
- The mobile station is operating in slotted mode, it can also specify its preferred slot cycle index by means of the SLOT_CYCLE_INDEX field in the same message.

Slot Cycles


•SCI
•163.84 s
•7
•6
•5
•4
•3
•2
•1
•0
$T = 2^{SCI}$
= Slot Cycle Index
T = Slot Cycle Length in 1.28 s units
•1.28 s
•80 ms

Paging Slot Determination

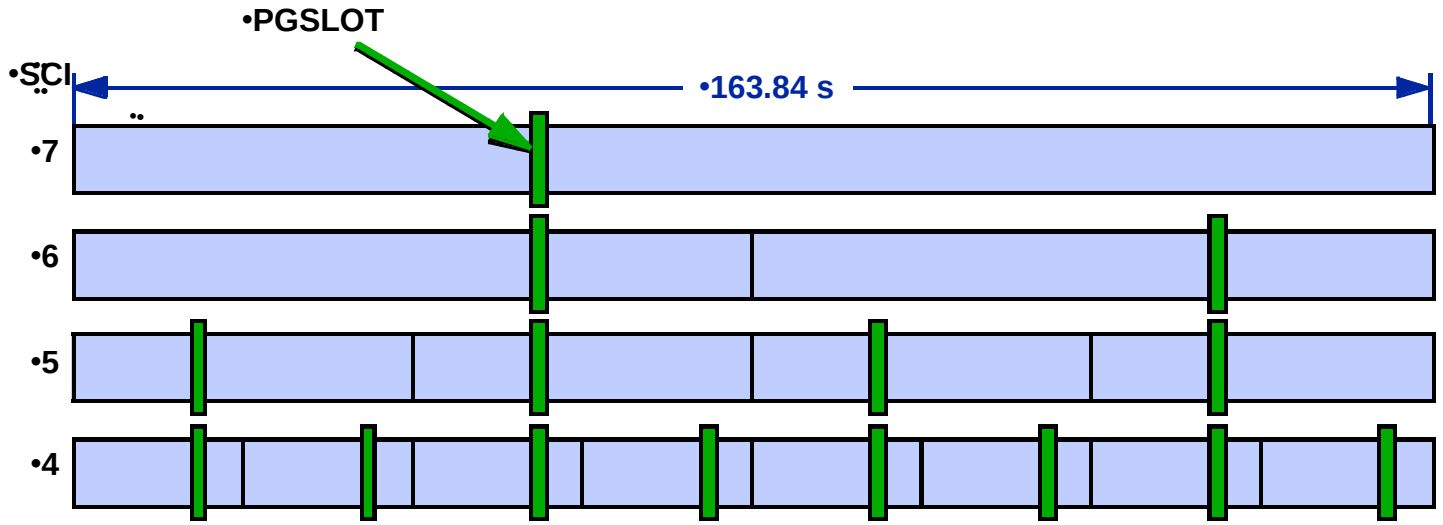


$$DECORR = 6 \times HASH_KEY\ [0..11]$$

$$PGSLOT = \text{int}\left(\frac{2048 \left(\left(40503 \left(H \text{ XOR } L \text{ XOR } DECORR \right) \right) \bmod 2^{16} \right)}{2^{16}} \right)$$

- To determine its assigned paging slots, the mobile station uses the hash function to select a number, PGSLOT, in the range 0 to 2047 (spanning the maximum slot cycle which is 163.84 seconds)

Paging Slot Determination

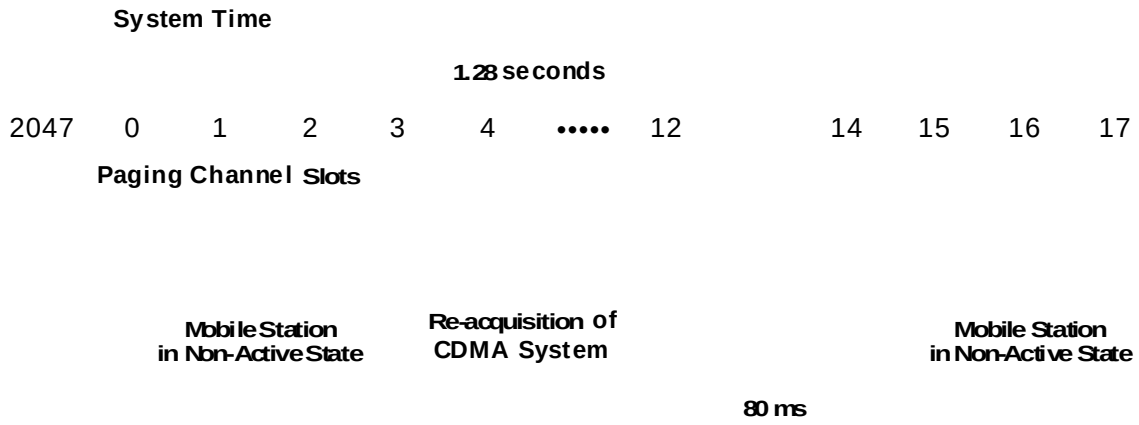


•n •The mobile station's assigned slots are given by:

$$(\text{int}\ (t/4)\text{-PGSLOT})\text{mod}(16 \times T) = 0$$

•where "t" is the system time in frames and SCI is the slot cycle index

Slotted Mode Operation

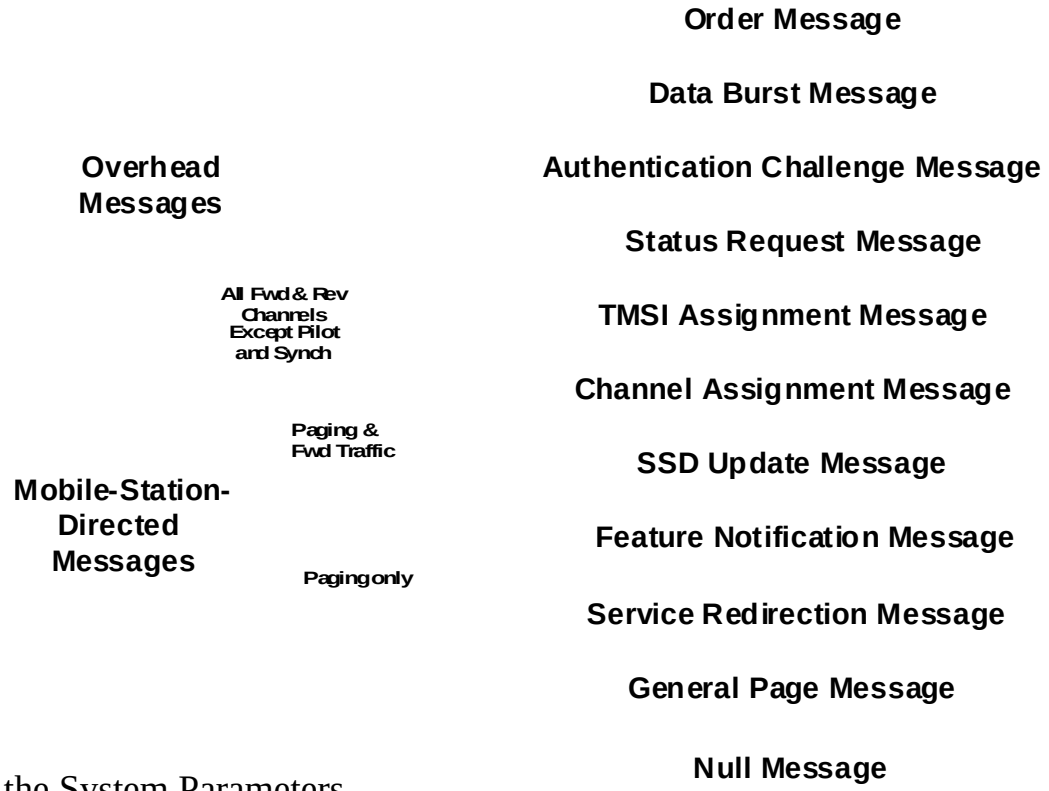


- The figure shows an example for a slot cycle length of 1.28 seconds (16 slots) in which, based on PGSLOT, the slot to be monitored was determined to be the 13th.
- The mobile station “wakes up” at slot #12 and re-acquires the system
- It monitors the slot for messages during slot #13
- The next slot the mobile station must monitor is 16 slots later (slot #29)
- If a page message started in one slot does not fit, it continues in the following slot (each paging message must be completely contained in one or two slots)
- Eventually the mobile realizes that there are no more pages for its class (see General Page Message structure discussed later) and “goes back to sleep”

Paging Channel Overhead Messages

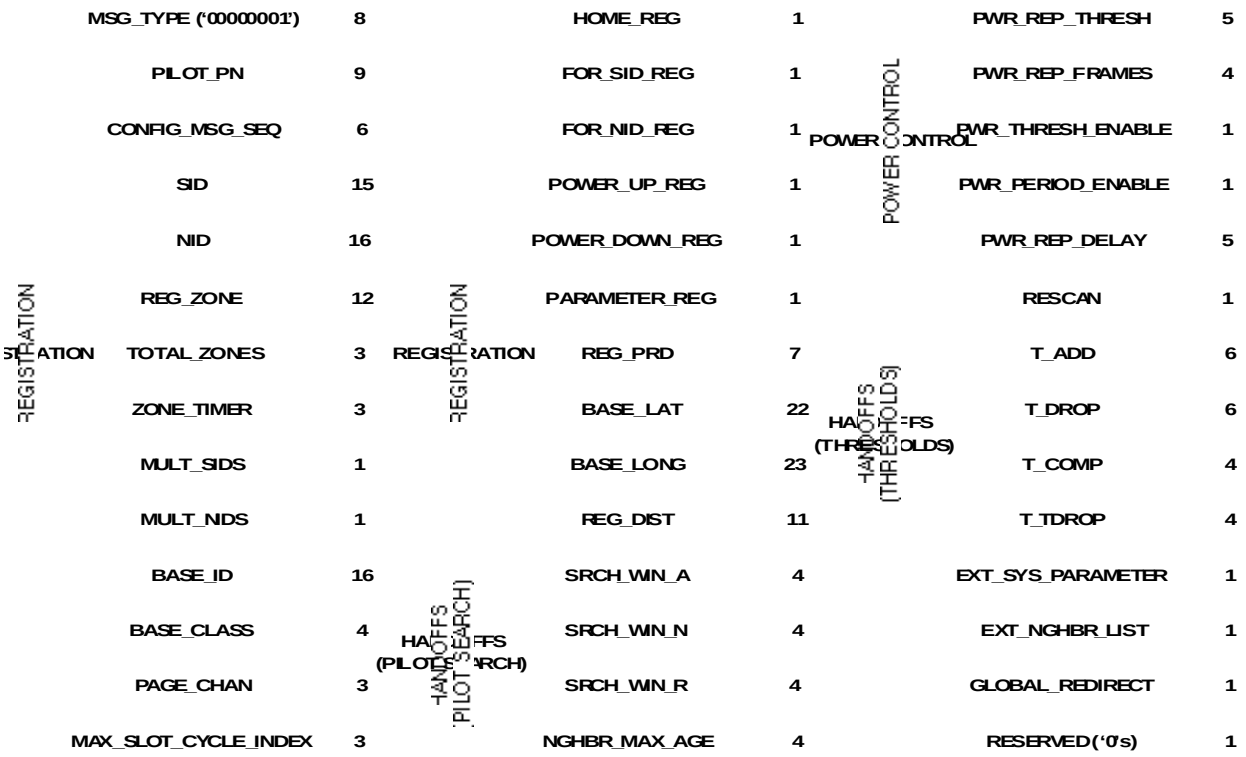
Overhead Messages		ACC_CONFIG_SEQ
		Access Parameters Message
		CONFIG_SEQ
		System Parameters Message
	Configuration Parameter Messages	CDMA Channel List Message
		Extended System Parameters Message
		Extended Neighbor List Message
Mobile-Station-Directed Messages		
		Global Service Redirection Message

Paging Channel Mobile-Station-Directed Messages



Only **selected fields** of the System Parameters Message are discussed next. All the other fields in this message as well as some of the remaining overhead messages are discussed later where it is more appropriate.

System Parameters Message (Paging Channels)



System Parameters Message (Paging Channels) – cont.

- **MSG_TYPE** – Message type: ‘00000001’
- **PILOT_PN** – Pilot PN offset index for this base station.
- **CONFIG_MSG_SEQ** – Current value of CONFIG_SEQ
- **SID** – System identification
- **NID** – Network identification
- **MULT_SIDS** – ‘1’ if the mobile station may store multiple entries of SID_NID_LIST containing different SIDs; ‘0’ otherwise.
- **MULT_NIDS** – ‘1’ if the mobile station may store multiple entries of SID_NID_LIST having the same SID (with different NIDs); ‘0’ otherwise.
- **BASE_ID** – This base station identification number
- **BASE_CLASS** – ‘0001’ indicating that this is a Public PCS System

System Parameters Message (Paging Channels) – cont.

- **PAGE_CHAN** – Number of Paging Channels on this CDMA Channel in this base station (cannot be ‘000’)
- **MAX_SLOT_CYCLE_INDEX** – Maximum value permitted for the Slot Cycle Index in this base station
- **RESCAN** – ‘1’ if the mobile stations are to re-initialize and re-acquire the system upon receiving this message; ‘0’ otherwise
- **EXT_SYS_PARAMETER** – ‘1’ if the base station sends the “Extended System Parameters Message” on the Paging Channel; ‘0’ otherwise
- **EXT_NGHBR_LIST** – ‘1’ if the base station sends the “Extended Neighbor List Message” on the Paging Channel; “0” otherwise
- **GLOBAL_REDIRECT** – ‘1’ if the base station is sending the “Global Service Redirection Message” on the Paging Channel; ‘0’ otherwise

CDMA Channel List Message (Paging Channels)

Field	Length (bits)
MSG_TYPE (‘00000100’)	8
PILOT_PN	9
CONFIG_MSG_SEQ	6
One or more occurrences of the following:	
CDMA_FREQ	**11**
One occurrence of the following:	
RESERVED (‘0’s)	**0 - 7 (as needed)**

Block	Preferred Set Channel Numbers
A	25, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275
D	325, 350, 375
B	425, 450, 475, 500, 525, 550, 575, 600, 625, 650, 675
E	725, 750, 775
F	825, 850, 875
C	925, 950, 975, 1000, 1025, 1050, 1075, 1100, 1025, 1150, 1175

CDMA Channel List Message (Paging Channels) – cont.

- **CDMA_FREQ** – CDMA Channel frequency assignment. The base station includes one occurrence of this field for each CDMA Channel that is supported by this base station. Each occurrence of this field is set to the CDMA channel number for the corresponding CDMA Channel assignment.
- The order in which occurrences of this field are included gives the designations of the supported CDMA Channels as “CDMA Channel 1” through “CDMA Channel N”.
- CDMA Channels in the preferred set of CDMA frequency assignments should occur first.
- **RESERVED** – This field contains as many zeroes as needed in order to make the length of the entire message an integer number of octets.

Extended System Parameters Message (Paging Channels)

Field	Length (bits)
MSG_TYPE (‘00001101’)	8
PILOT_PN	9
CONFIG_MSG_SEQ	6
RESERVED (‘0’)	1
PREF_MSID_TYPE	**3**
MCC	**10**
IMSI_11_12	**7**
TMSI_ZONE_LEN	**4**
TMSI_ZONE	**8 x TMSI_ZONE_LEN**
BCAST_INDEX	**3**
RESERVED (‘0000000’)	7

Extended System Parameters Message (Paging Channels) – cont.

- **PREF_MSID_TYPE –** Preferred Access Channel Mobile Station Identifier Type. Defines the type of MSID that the mobile station is to use on the Access Channel. Values are as follows:

  ‘000’ – IMSI
  ‘001’ – IMSI and ESN
  ‘010’ – TMSI (valid TMSI is assigned); IMSI (TMSI not assigned)
  ‘011’ – TMSI (valid TMSI is assigned); IMSI and ESN (TMSI not assigned)
  ALL OTHER VALUES ARE RESERVED.

  - **MSID**: <insert definition>

- **MCC** – Mobile Country Code.

  - **MOBILE COUNTRY CODE**: 1 to three digits identifying the country, as defined in “CCITT Blue Book, Volume II, Fascicle II.2, Recommendation E.212, November 1988.

- **IMSI_11_12** – Digits 11 and 12 of the IMSI.

  - **DIGITS 11 AND 12 OF THE IMSI**: Same as the “Mobile Network Code” (MNC).

- **TMSI_ZONE_LEN** – Number of octets in the TMSI_ZONE field.

- **TMSI_ZONE** – Number of this TMSI zone. (TMSI zone is an administrative zone that allows the TMSI to be reused).

- **BCAST_INDEX** – Broadcast Slot Cycle Index. To enable periodic broadcast paging, the base station sets this field to an unsigned 3-bit value (1-7) equal to the broadcast slot cycle index. To disable periodic broadcast paging, the base station sets this field to ‘000’.

Extended Neighbor List Message (Paging Channels)

Field	Length (bits)
MSG_TYPE (‘00001110’)	8
PILOT_PN	9
CONFIG_MSG_SEQ	6
PILOT_INC	4
Zero or more occurrences of the following record:	
NGHBR_CONFIG	**3**
NGHBR_PN	**9**
SEARCH_PRIORITY	**2**
FREQ_INCL	**1**
NGHBR_BAND	**0 or 5**
NGHBR_FREQ	**0 or 11**
One occurrence of the following field:	
RESERVED (‘0’s)	0 - 7 (as needed)

Extended Neighbor List Message (Paging Channels) – cont.

- **PILOT_INC** – The mobile station will search for the Remaining Set pilots as pilot PN sequence index values that are multiples of the value (1-15) in this field.

- **NGHBR_CONFIG** – Describes the configuration of the neighbor corresponding to this record relative to the configuration of this base station as follows:

  '000' – Has the **same** number of frequencies with Paging Channels, and with the **same** number of Paging Channels
  • **FREQ_INCL = 0**: frequency assignment is the same as in the current current base station
  • **FREQ_INCL =1**: frequency assignment is given by NGHBR_BAND and NGHBR_FREQ

  '001' – Has the **same** number of frequencies with Paging Channels, but with a **different** number of Paging Channels
  • **FREQ_INCL = 0**: frequency assignment is the same as in the current current base station
  • **FREQ_INCL =1**: frequency assignment is given by NGHBR_BAND

- and NGHBR_FREQ
  '010' – May have a **different** number of frequencies with Paging Channels
  • **FREQ_INCL = 0**: neighbor has a Primary Paging Channel on the first CDMA channel listed in the "CDMA Channel List Message" transmitted by the current base station
  • **FREQ_INCL =1**: neighbor has a Primary Paging Channel in the frequency assignment is given by NGHBR_BAND and NGHBR_FREQ

Extended Neighbor List Message (Paging Channels) – cont.

- ‘011’ – This neighbor’s configuration is **unknown**
  - **FREQ_INCL = 0**: this CDMA frequency assignment has a Pilot Channel
  - **FREQ_INCL = 1**: the CDMA frequency assignment given by NGHBR_BAND and NGHBR_FREQ has a Pilot Channel

  ALL OTHER VALUES ARE RESERVED.

- **NGHBR_PN** – Pilot PN sequence offset index for the neighbor (in units of 64 PN chips).

- **SEARCH_PRIORITY** – Defines the search priority for the Pilot Channel corresponding to NGHBR_PN as follows:
  - ‘00’ – Low
  - ‘01’ – Medium
  - ‘10’ – High
  - ‘11’ – Very High

- **FREQ_INCL** – Frequency included indicator. This field is set to ‘1’ if the NGHBR_FREQ field for this neighbor base station is included in this record; and it is set to ‘0’ if not.

- **NGHBR_BAND** – CDMA band class as follows:
  - ‘0’ – 800 MHz cellular
  - ‘1’ – 1900 MHz PCS

- **NGHBR_FREQ** – CDMA Channel number of the CDMA Channel containing the Paging Channel the mobile station is to search for.

- **RESERVED** – Enough zero bits to make the length of the message an integer number of octets.

Global Service Redirection Message, (Paging Channels)

Field	Length (bits)
MSG_TYPE ('00010010')	8
PILOT_PN	9
CONFIG_MSG_SEQ	6
REDIRECT_ACCOLC	16
RETURN_IF_FAIL	1
DELETE_TMSI	1
RESERVED	1
One occurrence of the following record:	
RECORD_TYPE	8
RECORD_LEN	8
Type-specific fields	8 x RECORD_LEN

Subfield	Length (bits)
BAND_CLASS	5
EXPECTED_SID	15
EXPECTED_NID	16
RESERVED	4
NUM_CHANS	4
NUM_CHANS occurrences of the following field:	
CDMA_CHAN	11
RESERVED ('0's)	0-7 (as needed)

Global Service Redirection Message, (Paging Channels) – cont.

- **ACK_SEQ –** Acknowledge sequence number. The base station sets this field to the value of the MSG_SEQ from the most recently received Access Channel message requiring acknowledgment from the mobile station addressed by this order
- **MSG_SEQ –** Message sequence number. The base station sets this field to the message sequence number for this order.
- **ACK_REQ –** Acknowledgment required indicator. The base station sets this field to ‘1’ to request from the mobile station acknowledge to this order. The base station sets this field to ‘0’ to indicate that the mobile station is not required to acknowledge this order.
- **VALID_ACK –** Valid acknowledgment indicator. To acknowledge the most recently received Access Channel message from the mobile station the base station sets this field to ‘1’. If this order does not acknowledge the most recently received Access Channel message from the mobile station, the base station sets this field to ‘0’.
- **ADDR_TYPE –** Address type. The base station sets this field to the value shown in the following table corresponding to the type of address contained in the address field:

Global Service Redirection Message, (Paging Channels) – cont.

Description	ADDR-TYPE (binary)	ADDR_LEN (octets)
Reserved	000	-
ESN	001	4 to 7
IMSI	010	5 to 7
TMSI	011	2 to 12
Reserved	100	-
BROADCAST	101	Variable
All other ADDR_TYPE values are reserved		

•**ADDR_LEN –** Address field length. The base station sets this field to the number of octets included in the ADDRESS field.

•**ADDRESS –** Mobile station or broadcast address. The base station sets this field to the mobile station or broadcast address, according to the address type specified in the ADDR_TYPE field

•If the ADDR_TYPE is equal to ‘001’, the ADDRESS field has the following structure:

Global Service Redirection Message, (Paging Channels) – cont.

Sub field	Length (bits)
ESN8	8 x ADDR_LEN

If the ADDR_TYPE is equal to ‘010’, the ADDRESS field has the following structure

Sub field	Length (bits)
IMI_CLASS	1
IMI Class-specific subfields	7 + 8 x (ADDR_LEN – 1)

If the ADDR_TYPE is equal to ‘011’, the ADDRESS field has the following structure

Sub field	Length (bits)
TMSI_ZONES	If ADDR_LEN > 4 then ADDR_LEN –4; otherwise 0
TMSI_CODE_ADDR	If ADDR_LEN > 4 then ADDR_LEN –4; otherwise ADDR_LEN

General Page Message (Paging Channels)

Field	Length (bits)
MSG_TYPE (‘00010001’)	8
CONFIG_MSG_SEQ	6
ACC_MSG_SEQ	6
CLASS_0_DONE	1
CLASS_1_DONE	1
TMSI_DONE	1
ORDERED_TMSIS	1
BROADCAST_DONE	1
RESERVED_	4
ADD_LENGTH	3
ADD_PFIELD	8 x ADD_LENGTH
Zero or more occurrences of the following page record:	
PAGE_CLASS	2
PAGE_SUBCLASS	2
Page-type-specific fields	Page-type-specific
REESERVED (‘0’s)	0–7 (as needed)

Page Record Formats	Page Class (binary)	Page Subclass (binary)	Page Record Format No (decimal)
Class 0, IMSI_S included	00	00	*0*
Class 0, IMSI_S and IMSI_11_12 included	00	01	*1*
Class 0, IMSI and MCC included	00	10	*2*
Class 0, IMSI_S, IMSI_11_12, and MCC included	00	11	*3*
Class 1, IMSI and MCC included	01	00	*4*
Class 1, IMSI_S, IMSI_11_12, and MCC included	01	01	*5*
Class 2 with 32 bit TMSI_CODE_ADDR (SID, NID & TMSI_ZONE not included)	10	00	*8*
Class 2 with 24 bit TMSI_CODE_ADDR (SID, NID & TMSI_ZONE not included)	10	01	*9*
Class 2 with 16 bit TMSI_CODE_ADDR (SID, NID & TMSI_ZONE not included)	10	10	*10*
Class 2 with 32 bit TMSI_CODE_ADDR (SID, NID & TMSI_ZONE included)	10	11	*11*
Class 3, Broadcast	11	00	*12*

General Page Message (Paging Channels) – cont.

- Page Channel messages other than the General Page Message can be addressed, by means of the ADDRESS field, to either a specific mobile station identified by its ESN (ADDR_TYPE = ‘001’), a specific IMSI (ADDR_TYPE = ‘010’), or a specific TMSI (ADDR_TYPE = ‘011’).
- General Page Messages can only be addressed to specific IMSIs or TMSIs.
- The base station assigns the mobile station a page class and subclass which determines how it will be addressing that mobile station in General Page Messages. If the PAGE_CLASS of a page record is 00 or 01, then this paging record refers to a mobile addressed by IMSI. If the PAGE_CLASS of a page record is 10, then this paging record refers to a mobile addressed by TMSI. If the PAGE_CLASS of a page record is 11, then this paging record refers to a mobile addressed as part of a broadcast group.
- General Page Messages contain four fields, “CLASS_0_DONE”, “CLASS_1_DONE”, “TMSI_DONE”, and “ORDERED_TMSIS”, which indicate when a personal station operating in slotted mode may stop monitoring the Paging Channel.
    - The concepts of “CLASS 0” and “CLASS 1” have to do with the number of digits in the IMSI, and are defined in the next slide.
- **CLASS_0_DONE** – Set to ‘1’ by the base station if all the “class 0” page records or other directed messages for mobile stations operating in slotted mode, active in this slot, and having an assigned class 0 IMSI, have been sent by the end of this “General Page Message”. Set to ‘0’ otherwise.

General Page Message (Paging Channels) – cont.

- **CLASS_1_DONE** – Set to ‘1’ by the base station if all the “class 1” page records or other directed messages for mobile stations operating in slotted mode, active in this slot, and having an assigned class 1 IMSI, have been sent by the end of this “General Page Message”. Set to ‘0’ otherwise.
- **TMSI_DONE** – Set to ‘1’ by the base station if all the “class 10” page records or other directed messages for mobile stations operating in slotted mode, active in this slot, and having an assigned TMSI have been sent by the end of this “General Page Message”. Set to ‘0’ otherwise.
- **ORDERED_TMSIS** – Set to ‘1’ by the base station if all the “class 10” page records in all the “General Page Messages” sent for mobile stations operating in slotted mode, active in this slot, have the TMSI code values of the TMSI_CODE and TMSI_CODE_ADDR fields in ascending numerical order. Set to ‘0’ otherwise.
- **BROADCAST_DONE** – Set to ‘1’ by the base station if all the broadcast (“class 11”) page records, have been sent by the end of this “General Page Message”. Set to ‘0’ otherwise.
- **ADD_LENGTH** – Number of octets in the page-message-specific fields.
- **ADD_FIELD** – Additional page-message-specific fields.
- **RESERVED** – Up to 7 bits (as needed) to make the total length of the message an integer number of octets.

IMSI

IMSI (Up to 15 digits)

MCC NMSI (Up to 12 digits)

MNC MSIN (Up to 10 digits)

(3 digits) (2 digits)

IMSI – International Mobile Station Identity
MCC – Mobile Country Code
NMSI – National Mobile Station Identity
MNC – Mobile Network Code
MSIN – Mobile Station Identification Number

Class 0 IMSI: 15 digits — Class 1 IMSI: less than 15 digits

IMSI_S

IMSI with less than 10 digits

0 0

IMSI with 10 digits

IMSI with more than 10 digits

(10 digits)

(34 bits)

IMSI_S2 **IMSI_S1**

IMSI_S

IMSI_11_12

IMSI with less than 12 digits

0 0

1 2 3 4 5 6 7 8 9 10 11 12

IMSI with 12 digits

IMSI_11_12

1 2 3 4 5 6 7 8 9 10 11 12

IMSI with more than 12 digits

1 2 3 4 5 6 7 8 9 10 11 12 13 14

When the IMSI has 12 or more digits, IMSI_11_12 is equal to the 11th and 12th digits of the IMSI. When the IMSI has fewer than 12 digits, IMSI_11_12 is equal to the last two digits.

Page Records (Class 0)

Class 0, Subclass 0

Field	Length (bits)
MSG_SEQ	3
IMSI_S	34
SPECIAL_SERVICE	1
SERVICE_OPTION	0 or 16

Class 0, Subclass 1

Field	Length (bits)
MSG_SEQ	3
IMSI_11_12	7
IMSI_S	34
SPECIAL_SERVICE	1
SERVICE_OPTION	0 or 16

Class 0, Subclass 2

Field	Length (bits)
MSG_SEQ	3
MCC	10
IMSI_S	34
SPECIAL_SERVICE	1
SERVICE_OPTION	0 or 16

Class 0, Subclass 3

Field	Length (bits)
MSG_SEQ	3
MCC	10
IMSI_11_12	7
IMSI_S	34
SPECIAL_SERVICE	1
SERVICE_OPTION	0 or 16

Page Records (Class 0) – cont.

See next series of slides