TSG-RAN Working Group1 meeting #7 **TSGR1#7(99)b00**

Source: ~~———~~**Texas Instruments and InterDigital Communications Corporation**

Title: TDD Cell Search and Text Proposals for 25.221, 25.223 and 25.224

Document for: Approval

~~Introduction~~

~~Tdoc 578 by InterDigital presented a new cell search procedure, based on modulated secondary synchronization codes. The results of Tdoc 578 indicate that the proposed scheme exhibits better performance than the current scheme along with a lower complexity. Additional results are presented in Tdoc 976. An improved version of this scheme is presented in Tdoc B99.~~

~~An alternative scheme was presented by TI in Tdoc 815. It was agreed that the two schemes will be compared by using a common set of simulation parameters and assumptions. This common set is defined in Tdoc A52.~~

~~Tdoc B99 compares the performance of InterDigital scheme and TI scheme. Based on the results of Tdocs 578, 976 and B99, we propose to adopt the cell search scheme of Tdoc B99 for TDD mode. The required changes in the specification documents are given below.~~

Text proposal for 25.221

5.4 The physical synchronisation channel (PSCH)

[Editors Note : The detailed scheme of CCCH pointing by SCH is FFS.]

~~The PSCH is similar to the FDD SCH.~~ In order not to limit the UL/DL asymmetry the PSCH is mapped on one or two DL slots per frame only.

There are three cases of SCH and CCCH allocation as follows:

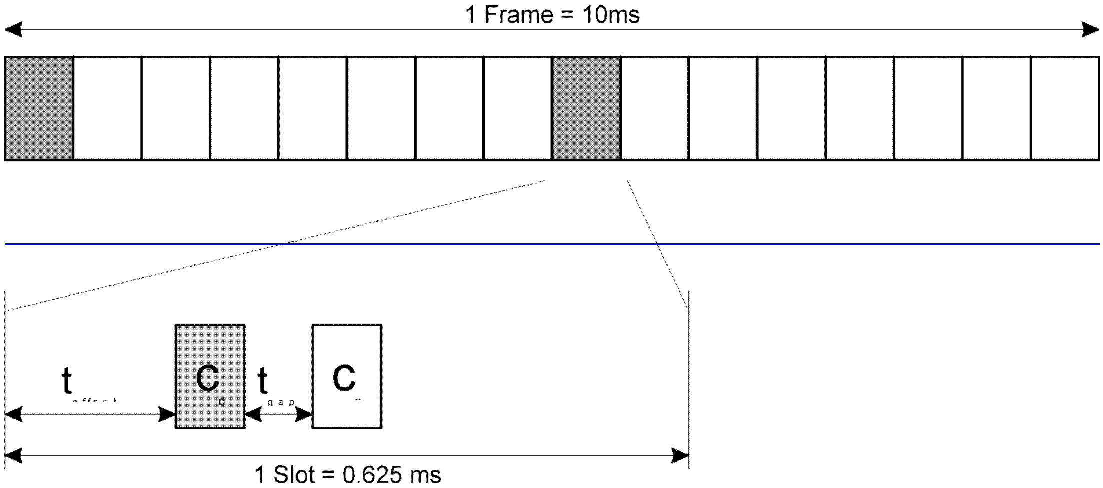
Case 1) SCH and CCCH allocated in TS#k, k=0….1~~5~~4

Case 2) SCH in two TS and CCCH in the same two TS: TS#k and TS#k+8, k=0…~~7~~6

Case 3) SCH in two TS, TS#k and TS#k+8, k=0…~~7~~6, and the primary CCCH TS#i, i=0…1~~5~~4, pointed by SCH

The position of SCH (value of k) in frame can change on a long term basis ~~in any case~~.

Figure 12 is one example, k=0, of Case 2 or Case 3. In this case, the PSCH uses system-wide always the same two DL slots, which are slot 0 and slot 8.



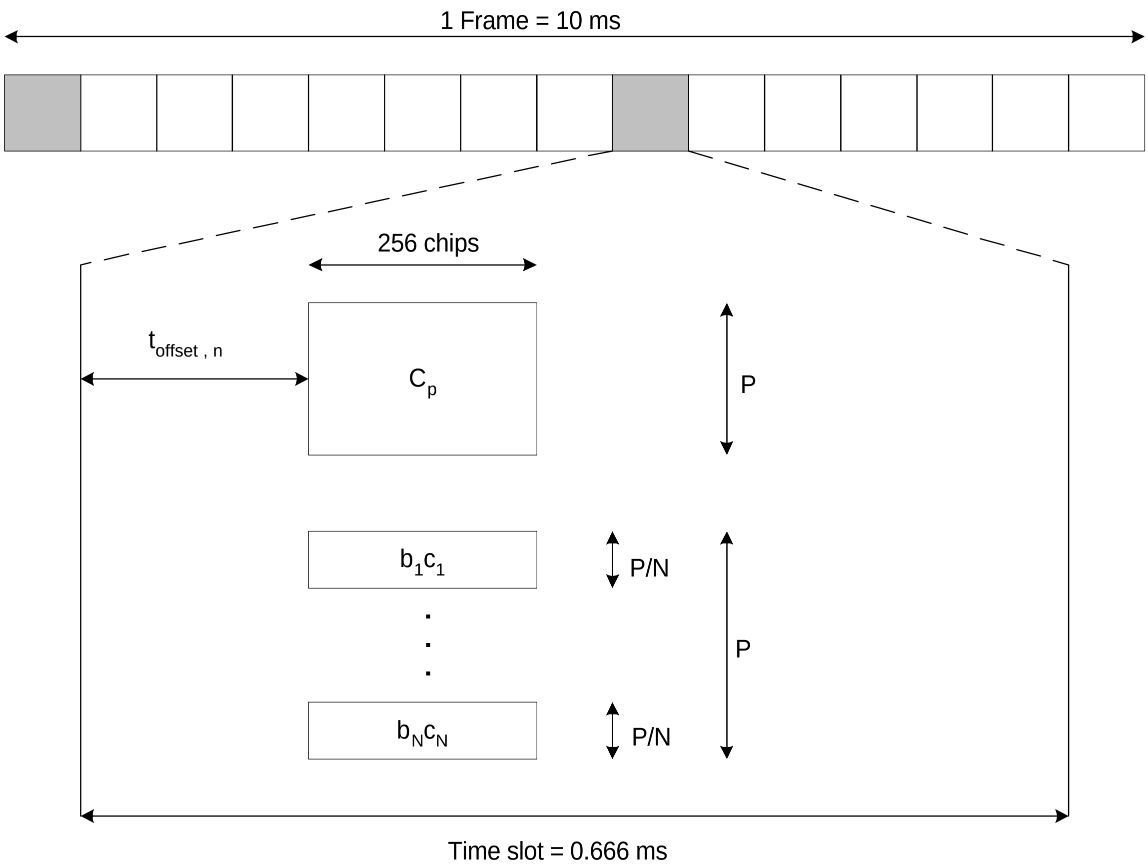


Figure 12 Scheme for Synchronisation channel SCH consisting of one primary sequence C_p and ~~N~~ 3 parallel secondary sequences ~~C_s~~ per slot. ~~(example of~~ for ~~Case 2 or Case 3, N=4 for case 2, N=7 for case 3.)~~

As depicted in Figure 12, the PSCH consists of a primary code sequence and ~~N~~3 parallel secondary code sequences ~~with~~ of length 256 chips ~~length~~each. ~~N=3 for case 1,N=4 for Case 2, and N=7 for case 3.The used sequences C_p and C_s are the same as in FDD-Mode, see [2].~~ The primary and secondary code sequences are defined in TS 25.223 Section 7.1. The secondary codes are transmitted either in the I channel or the Q channel, depending on the code group.

~~The time offset t_{gap} is the time between the primary synchronisation code and the secondary synchronisation code. It provides enough time for calculations and a better interference distribution, since the codes do not superimpose. The exact value is to be determined.~~

Due to mobile to mobile interference, it is mandatory for public TDD systems to keep synchronisation between base stations. As a consequence of this, a capture effect concerning PSCH can arise. The time offset t_{offset} enables the system to overcome the capture effect.

The time offset t_{offset} is one of 32 values, depending on the cell parameter, thus on the code group of the cell, cf. 'Table 9 Mapping scheme for Cell Parameters, Code Groups, Scrambling Codes, Midambles and t_{offset}' in 'TS25.223 Spreading and modulation (TDD)'. The exact value for t_{offset}, regarding column 'Associated t_{offset}' in Table 9 from TS25.223, is given by:

$$t_n = t_{offset,n} = n \cdot T_C \cdot \left\lfloor \frac{2560-96-512-\frac{t_{gap}}{T_C}}{31} \right\rfloor \quad ; n = 0 \ldots 31$$

$$t_{offset,n} = n \cdot T_c \left\lfloor \frac{2560-96-256}{31} \right\rfloor = n \cdot 71 T_c \;; \quad n = 0, \ldots, 31$$

~~We recommend to include this equation into TS25.221 specification document, as then t_{offset} can be derived immediately when t_{gap} is given.~~

Text Proposal for 25.223

7. Synchronisation codes

7.1 Code Generation

~~The code generation for synchronisation codes is handled in the same way as in FDD Mode. Thus we refer to TS 25.213, chapter '5.2.3 Synchronisation Codes'. From this procedure we obtain one primary synchronisation code $C_p = C_{SCH,0}$ and seventeen different secondary synchronisation codes $C_{S,i} = C_{SCH,i}$ with i=1…17.~~

~~To avoid misunderstandings when documents are reorganised in the future, we repeat the actual content of this chapter below using small font.~~

~~The Primary code sequence, C_p is constructed as a so-called generalised hierarchical Golay sequence. The Primary SCH is furthermore chosen to have good aperiodic auto correlation properties.~~

~~Letting a = < $x_1, x_2, x_3, \ldots, x_{16}$ > = < 0, 0, 0, 0, 0, 0, 1, 1, 0, 1,0, 1, 0, 1, 1, 0 > and~~

~~b = < $x_1, x_2, x_3, \ldots, x_8, x_1, x_2, x_3, \ldots, x_8$ >.~~

~~The PSC code is generated by repeating sequence 'a' modulated by a Golay complementary sequence.~~

~~Letting $y = < a, a, a, \bar{a}, \bar{a}, a, \bar{a}, \bar{a}, a, a, a, \bar{a}, a, \bar{a}, a, a >$~~

~~The definition of the PSC code word C_p follows (the left most index corresponds to the chip transmitted first in each time slot):~~

~~C_p=< y(0),y(1),y(2),...,y(255) >.~~

~~Let the sequence z = < *b, b, b, b, b, b, b, b, b, b, b, b, b, b, b, b*>. Then the Secondary Synchronization code words, <$C_1, \ldots, C_{17}$ > are constructed as the position wise addition modulo 2 of a Hadamard sequence and the sequence *z*.~~

~~The Hadamard sequences are obtained as the rows in a matrix H_8 constructed recursively by:~~

$$H_0 = (0)$$

$$H_k = \begin{pmatrix} H_{k-1} & H_{k-1} \\ H_{k-1} & \overline{H_{k-1}} \end{pmatrix}, \quad k \geq 1$$

~~The rows are numbered from the top starting with row *0* (the all zeros sequence).~~

~~The Hadamard sequence *h* depends on the chosen code number *n* and is denoted h_n in the sequel.~~

~~This code word is chosen from every 8th row of the matrix H_8. Therefore, there are 32 possible codewords out of which *n* = 1, 2,…,17 are used.~~

~~Furthermore, let $h_n(i)$ and $z(i)$ denote the *i*:th symbol of the sequence h_n and *z*, respectively.~~

~~Then h_n is equal to the row of H_8 numbered by the bit reverse of the 8 bit binary representation of *n*.~~

The definition of the n:th SCH code word follows (the left most index correspond to the chip transmitted first in each slot):

$C_{SCH,n}$ = < $h_n(0)$+ z(0), $h_n(1)$+ z(1), $h_n(2)$+ z(2), …,$h_n(255)$+ z(255) >,

All sums of symbols are taken modulo 2.

These PSC and SSC binary code words are converted to real valued sequences by the transformation '0' -> '+1', '1' -> '-1'.

The Secondary SCHcode words are defined in terms of $C_{SCH,n}$ and the definition of < C_1,…,C_{17} > now follows as:

C_i = $C_{SCH,i}$, i=1,…,17

7.1.1 Primary Code

The Primary code sequence, C_p is constructed as a so-called generalised hierarchical Golay sequence. This primary code is the same as the primary code used in FDD mode.The Primary SCH is furthermore chosen to have good aperiodic auto correlation properties.

Letting a = < x_1, x_2, x_3, …, x_{16} > = < 0, 0, 0, 0, 0, 0, 1, 1, 0, 1,0, 1, 0, 1, 1, 0 > ,

The PSC code is generated by repeating sequence 'a' modulated by a Golay complementary sequence.

Letting $y =< a, a, a, \bar{a}, \bar{a}, a, \bar{a}, \bar{a}, a, a, a, \bar{a}, a, \bar{a}, a, a >$

The definition of the PSC code word C_p follows (the left most index corresponds to the chip transmitted first in each time slot):

C_p=< y(0),y(1),y(2),...,y(255) >.

The PSC binary sequence is converted to a real valued sequence by the transformation '0'→1, and '1'→-1.

7.1.2 Secondary Codes

The Secondary Synchronization Codes (SSC's) are proposed to be common to all the base stations. However, before the transmission, they are QPSK modulated. The QPSK modulation carries the following information:

- The code group that the base station belongs to (5 bits; Cases 1,2,3)
- The position of the frame within an interleaving period of 20 mesec (1 bits, Cases 1,2,3)
- The position of the slot within the frame (1 bit, Cases 2,3)
- The location of the primary CCCH (3 bits, Case 3)

The secondary synchronization codes are partitioned into four (possibly overlapping) code sets – code_set_1, code_set_2, code_set_3, code_set_4. The set used provides the following information two code sets for Case 1, four code sets for Case 2 and thirty two code sets (possibly overlapping) for Case 3. The set used provide the following information:

Case 1:

Code Set	Code Group
1	0-157
2	8-1516-31

The ~~secondary codes may be transmitted either by the I channel or by the Q channel. The selection of I/Q provides an additional bit of informaiton. The remaining bits of information are provided~~ code group and frame position information is provided by modulating the secondary codes in the ~~-~~code set.

Case 2:

Code Set	Code Group
1	0-7
2	8-15
3	16-23
4	24-31

The slot timing and frame position information is provided by the comma free property of the code word and the Code group is provided by modulating some of the secondary codes in the code set.

Case 3:

Code set k, k=1:32 is associated with Code group k-1. The slot information, the frame position information is provided by the comma free property of the code and the primary CCCH position is provided by modulating some of the codes in the code set.

~~Case 1: Total number of information bits – 6. 2 bits are provided by the selection of code set. 1 bit is provided by the I/Q selection. Each code set contains 3 codes. The remaining 3 bits are provided by modulating the three codes in the code set.~~

~~Case 2: Total number of information bits – 7. 2 bits are provided by the selection of code set. 1 bit is provided by the I/Q selection. Each code set contains 4 codes. The remaining 4 bits are provided by modulating the four codes in the code set.~~

~~Case 3: total number of information bits –10. 2 bits are provided by the selection of code set. 1 bit is provided by the I/Q selection. Each code set contains 7 codes. The remaining 7 bits are provided by modulating the four codes in the code set.~~

~~Denote by N the number of codes in the code set (N=3 for Case1 , N=4 for case 2, N=7 for case 3).~~Denote by N, the number of secondary codes required for representing each of the above cases. N = 6 for case 1, N = 12 for case 2and N = 16 for case 3.

The SSC's {C_1, C_2~~, …,~~, C_{NN}} are constructed as the position wise modulu 2 addition of ~~the~~ a mask sequence ~~y~~z ~~defined in Section 7.1.1~~ and properly selected rows of the Hadamard matrix, H_8. H_8 is obtained recursively by,

$$H_0 = (0),$$

$$H_k = \begin{pmatrix} H_{k-1} & H_{k-1} \\ H_{k-1} & \overline{H}_{k-1} \end{pmatrix}, \quad k \geq 1$$

Let b = $< x_1,\ldots,x_8,\overline{x}_9,\ldots,\overline{x}_{16} >$ = <0,0,0,0,0,0,1,1,1,0,1,0,1,0,0,1> where $x_1,\ldots,x_{16}$ are defined in Section 7.1.1. Then, length 256 mask sequence z is given as,

z = $< b,b,b,\overline{b},b,b,\overline{b},\overline{b},b,\overline{b},b,\overline{b},\overline{b},\overline{b},\overline{b},\overline{b} >$.

The rows of Hadamarad matrix H_8 are numbered from the top starting with row 0, h_0, (the all ~~one~~zero sequence). The SSC's are constructed as

$C_k(i) = h_{r(k)}(i)$ + ~~y~~ $z(i)$, *modulo 2*, i = 0, …, 255 ;k=1,… N
where r(k) is given as below for Case 1, Case 2 and Case 3.

~~where the sequence y is the primary code defined in Section 7.1.1, and~~

Case 1

$$r(k) = \begin{cases} 16*[0,1,3] & \text{Code Set 1} \\ 16*[4,5,6] & \text{Code Set 2} \\ 16*[8,10,12] & \text{Code Set 3} \\ 16*[13,14,15] & \text{Code Set 4} \end{cases}$$

$$r(k) = \begin{cases} 16*[0,1,3] & \text{Code Set 1} \\ 16*[4,5,6] & \text{Code Set 2} \end{cases}$$

Case 2

$$r(k) = \begin{cases} 16*[0,1,2] & \text{Code Set 1} \\ 16*[3,4,5] & \text{Code Set 2} \\ 16*[6,7,8] & \text{Code Set 3} \\ 16*[9,10,11] & \text{Code Set 4} \end{cases}$$

Case 3

$$
r(k) = \begin{cases}
16*[0,1,2] & \text{Code Set 1} \\
16*[3,4,5] & \text{Code Set 2} \\
16*[6,7,8] & \text{Code Set 3} \\
16*[9,10,11] & \text{Code Set 4} \\
16*[12,13,14] & \text{Code Set 5} \\
16*[0,3,6] & \text{Code Set 6} \\
16*[0,4,7] & \text{Code Set 7} \\
16*[0,5,8] & \text{Code Set 8} \\
16*[0,9,12] & \text{Code Set 9} \\
16*[0,10,13] & \text{Code Set 10} \\
16*[0,11,14] & \text{Code Set 11} \\
16*[1,3,7] & \text{Code Set 12} \\
16*[1,4,6] & \text{Code Set 13} \\
16*[1,5,9] & \text{Code Set 14} \\
16*[1,8,10] & \text{Code Set 15} \\
16*[1,11,12] & \text{Code Set 16} \\
16*[1,13,15] & \text{Code Set 17} \\
16*[2,3,8] & \text{Code Set 18} \\
16*[2,4,9] & \text{Code Set 19} \\
16*[2,5,6] & \text{Code Set 20} \\
16*[2,7,10] & \text{Code Set 21} \\
16*[2,11,13] & \text{Code Set 22} \\
16*[2,12,15] & \text{Code Set 23} \\
16*[3,9,13] & \text{Code Set 24} \\
16*[3,10,12] & \text{Code Set 25} \\
16*[3,11,15] & \text{Code Set 26} \\
16*[4,8,11] & \text{Code Set 27} \\
16*[4,10,14] & \text{Code Set 28} \\
16*[5,7,11] & \text{Code Set 29} \\
16*[5,10,15] & \text{Code Set 30} \\
16*[6,9,14] & \text{Code Set 31} \\
16*[7,9,15] & \text{Code Set 32}
\end{cases}
$$

~~Note that the primary code can be written as the position wise modulu 2 addition of the sequence y and h_0.~~

The SSC binary sequences are converted to a real valued sequence by the transformation '0'→1, and '1'→-1.

7.1.2.1 Case 1: One slot per frame, SCH and CCCH are in the same slot.

~~The modulated SSC's for code set n are obtained by multiplication of $C_k(i)$ with b_k i.e b_k* $C_k(i)$ where b_k, are given as~~The three SSC's from each code set are appropriately modulated and transmitted in parallel in each SCH slot and are given as

~~Code Group~~	~~Code Set~~	~~b_3~~	~~b_2~~	~~b_1~~	~~Associated t_{offset}~~
~~0~~	~~1~~	~~1~~	~~1~~	~~1~~	~~t_0~~
~~1~~	~~1~~	~~1~~	~~1~~	~~-1~~	~~t_1~~
~~2~~	~~1~~	~~1~~	~~-1~~	~~1~~	~~t_2~~
~~…~~	~~…~~	~~…~~	~~…~~	~~…~~	~~…~~
~~7~~	~~1~~	~~-1~~	~~-1~~	~~-1~~	~~t_7~~
~~8~~	~~2~~	~~1~~	~~1~~	~~1~~	~~t_8~~
~~…~~	~~…~~	~~…~~	~~…~~	~~…~~	~~…~~
~~15~~	~~2~~	~~-1~~	~~-1~~	~~-1~~	~~t_{15}~~
~~16~~	~~3~~	~~1~~	~~1~~	~~1~~	~~t_{16}~~
~~…~~	~~…~~	~~…~~	~~…~~	~~…~~	~~…~~
~~23~~	~~3~~	~~-1~~	~~-1~~	~~-1~~	~~t_{23}~~
~~24~~	~~4~~	~~1~~	~~1~~	~~1~~	~~t_{24}~~
~~…~~	~~…~~	~~…~~	~~…~~	~~…~~	~~…~~
~~31~~	~~4~~	~~-1~~	~~-1~~	~~-1~~	~~t_{31}~~

~~I/Q selection represents the frame number within an interleaving period of 20 msec. It is 'I' for 'first frame' and 'Q' for 'second frame'~~.
Note that modulation by "j" indicates that the code is transmitted on the Q channel.

Code Group	Code Set	Frame 1			Frame 2			Associated t_{offset}
0	1	C_1	C_2	C_3	C_1	C_2	$-C_3$	t_0
1	1	C_1	$-C_2$	C_3	C_1	$-C_2$	$-C_3$	t_1
2	1	$-C_1$	C_2	C_3	$-C_1$	C_2	$-C_3$	t_2
3	1	$-C_1$	$-C_2$	C_3	$-C_1$	$-C_2$	$-C_3$	t_3
4	1	jC_1	jC_2	C_3	jC_1	jC_2	$-C_3$	t_4
5	1	jC_1	$-jC_2$	C_3	jC_1	$-jC_2$	$-C_3$	t_5
6	1	$-jC_1$	jC_2	C_3	$-jC_1$	jC_2	$-C_3$	t_6
7	1	$-jC_1$	$-jC_2$	C_3	$-jC_1$	$-jC_2$	$-C_3$	t_7
8	1	jC_1	jC_3	C_2	jC_1	jC_3	$-C_2$	t_8
9	1	jC_1	$-jC_3$	C_2	jC_1	$-jC_3$	$-C_2$	t_9
10	1	$-jC_1$	jC_3	C_2	$-jC_1$	jC_3	$-C_2$	t_{10}
11	1	$-jC_1$	$-jC_3$	C_2	$-jC_1$	$-jC_3$	$-C_2$	t_{11}
12	1	jC_2	jC_3	C_1	jC_2	jC_3	$-C_1$	t_{12}
13	1	jC_2	$-jC_3$	C_1	jC_2	$-jC_3$	$-C_1$	t_{13}
14	1	$-jC_2$	jC_3	C_1	$-jC_2$	jC_3	$-C_1$	t_{14}
15	1	$-jC_2$	$-jC_3$	C_1	$-jC_2$	$-jC_3$	$-C_1$	t_{15}
16	2	C_4	C_5	C_6	C_4	C_5	$-C_6$	t_{16}
17	2	C_4	$-C_5$	C_6	C_4	$-C_5$	$-C_6$	t_{17}
…	…	…	…	…	…	…	…	…
20	2	jC_4	jC_5	C_6	jC_4	jC_5	$-C_6$	t_{20}
…	…	…	…	…	…	…	…	…
24	2	jC_4	jC_6	C_5	jC_4	jC_6	$-C_5$	t_{24}
…	…	…	…	…	…	…	…	…

31	2	$-jC_5$	$-jC_6$	C_4	$-jC_5$	$-jC_6$	$-C_4$	t_{31}

7.1.2.1 Cases 2: SCH and CCCH are in the same slot

~~The modulated SSC's for code group n are obtained by multiplication of $C_k(i)$ with b_k, i.e b_k * $C_k(i)$ where b_k are given as~~

~~Code Group~~	~~Code Set~~	~~b_3~~	~~b_2~~	~~b_1~~	~~Associated t_{offset}~~
~~0~~	~~1~~	~~1~~	~~1~~	~~1~~	~~t_0~~
~~1~~	~~1~~	~~1~~	~~1~~	~~-1~~	~~t_1~~
~~2~~	~~1~~	~~1~~	~~-1~~	~~1~~	~~t_2~~
~~…~~	~~…~~	~~…~~	~~…~~	~~…~~	~~…~~
~~7~~	~~1~~	~~-1~~	~~-1~~	~~-1~~	~~t_7~~
~~8~~	~~2~~	~~1~~	~~1~~	~~1~~	~~t_8~~
~~…~~	~~…~~	~~…~~	~~…~~	~~…~~	~~…~~
~~15~~	~~2~~	~~-1~~	~~-1~~	~~-1~~	~~t_{15}~~
~~16~~	~~3~~	~~1~~	~~1~~	~~1~~	~~t_{16}~~
~~…~~	~~…~~	~~…~~	~~…~~	~~…~~	~~…~~
~~23~~	~~3~~	~~-1~~	~~-1~~	~~-1~~	~~t_{23}~~
~~24~~	~~4~~	~~1~~	~~1~~	~~1~~	~~t_{24}~~
~~…~~	~~…~~	~~…~~	~~…~~	~~…~~	~~…~~
~~31~~	~~4~~	~~-1~~	~~-1~~	~~-1~~	~~t_{31}~~

~~I/Q selection represents the frame number within an interleaving period of 20 msec. It is 'I' for 'first frame' and 'Q' for 'second frame'.~~

~~Bit b_4 represents the slot position within the frame and it is '1' for 'slot k' and '-1' for 'slot k+8'.~~

Code Group	Code Set	Frame 1						Frame 2						Associated t_{offset}
		Slot k			Slot k+8			Slot k			Slot k+8			
0	1	C_1	C_2	C_3	C_1	C_2	$-C_3$	$-C_1$	$-C_2$	C_3	$-C_1$	$-C_2$	$-C_3$	t_0
1	1	C_1	$-C_2$	C_3	C_1	$-C_2$	$-C_3$	$-C_1$	C_2	C_3	$-C_1$	C_2	$-C_3$	t_1
2	1	jC_1	jC_2	C_3	jC_1	jC_2	$-C_3$	$-jC_1$	$-jC_2$	C_3	$-jC_1$	$-jC_2$	$-C_3$	t_2
3	1	jC_1	$-jC_2$	C_3	jC_1	$-jC_2$	$-C_3$	$-jC_1$	jC_2	C_3	$-jC_1$	jC_2	$-C_3$	t_3
4	1	jC_1	jC_3	C_2	jC_1	jC_3	$-C_2$	$-jC_1$	$-jC_3$	C_2	$-jC_1$	$-jC_3$	$-C_2$	t_4
5	1	jC_1	$-jC_3$	C_2	jC_1	$-jC_3$	$-C_2$	$-jC_1$	jC_3	C_2	$-jC_1$	jC_3	$-C_2$	t_5
6	1	jC_2	jC_3	C_1	jC_2	jC_3	$-C_1$	$-jC_2$	$-jC_3$	C_1	$-jC_2$	$-jC_3$	$-C_1$	t_6
7	1	jC_2	$-jC_3$	C_1	jC_2	$-jC_3$	$-C_1$	$-jC_2$	jC_3	C_1	$-jC_2$	jC_3	$-C_1$	t_7
8	2	C_4	C_5	C_6	C_4	C_5	$-C_6$	$-C_4$	$-C_5$	C_6	$-C_4$	$-C_5$	$-C_6$	t_8
9	2	C_4	$-C_5$	C_6	C_4	$-C_5$	$-C_6$	$-C_4$	C_5	C_6	$-C_4$	C_5	$-C_6$	t_9
10	2	jC_4	jC_5	C_6	jC_4	jC_5	$-C_6$	$-jC_4$	$-jC_5$	C_6	$-jC_4$	$-jC_5$	$-C_6$	t_{10}
11	2	jC_4	$-jC_5$	C_6	jC_4	$-jC_5$	$-C_6$	$-jC_4$	jC_5	C_6	$-jC_4$	jC_5	$-C_6$	t_{11}
12	2	jC_4	jC_6	C_5	jC_4	jC_6	$-C_5$	$-jC_4$	$-jC_6$	C_5	$-jC_4$	$-jC_6$	$-C_5$	t_{12}
13	2	jC_4	$-jC_6$	C_5	jC_4	$-jC_6$	$-C_5$	$-jC_4$	jC_6	C_5	$-jC_4$	jC_6	$-C_5$	t_{13}
14	2	jC_5	jC_6	C_4	jC_5	jC_6	$-C_4$	$-jC_5$	$-jC_6$	C_4	$-jC_5$	$-jC_6$	$-C_4$	t_{14}
15	2	jC_5	$-jC_6$	C_4	jC_5	$-jC_6$	$-C_4$	$-jC_5$	jC_6	C_4	$-jC_5$	jC_6	$-C_4$	t_{15}

16	3	C_7	C_8	C_9	C_7	C_8	$-C_9$	$-C_7$	$-C_8$	C_9	$-C_7$	$-C_8$	$-C_9$	t_{16}
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
23	3	jC_8	$-jC_9$	C_7	jC_8	$-jC_9$	$-C_7$	$-jC_8$	jC_9	C_7	$-jC_8$	jC_9	$-C_7$	t_{20}
24	4	C_{10}	C_{11}	C_{12}	C_{10}	C_{11}	$-C_{12}$	C_{10}	C_{11}	C_{12}	C_{10}	C_{11}	$-C_{12}$	t_{24}
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
31	4	jC_{11}	$-jC_{12}$	C_{10}	jC_{11}	jC_{12}	$-C_{10}$	$-jC_{11}$	jC_{12}	C_{10}	$-jC_{11}$	jC_{12}	$-C_{10}$	t_{31}

Note that, the C_p and $\{C_1, \ldots, C_N\}$ are pairwise mutually orthogonal and $\{C_1, \ldots, C_N\}$ were chosen to have good aperiodic cross correlation properties with C_p.

7.1.2.3 Case 3: SCH points the CCCH location in the frame

~~The modulated SSC's for code group n are obtained by multiplication of $C_k(i)$ with b_k, i.e b_k * $C_k(i)$ where b_k are given as~~

Code Group	Code Set	Frame 1						Frame 2						Associated t_{offset}
		Slot k			Slot k+8			Slot k			Slot k+8			
0	1	C_1	C_2	C_3	C_1	C_2	$-C_3$	$-C_1$	$-C_2$	C_3	$-C_1$	$-C_2$	$-C_3$	t_0
0	1	C_1	$-C_2$	C_3	C_1	$-C_2$	$-C_3$	$-C_1$	C_2	C_3	$-C_1$	C_2	$-C_3$	
0	1	jC_1	jC_2	C_3	jC_1	jC_2	$-C_3$	$-jC_1$	$-jC_2$	C_3	$-jC_1$	$-jC_2$	$-C_3$	
0	1	jC_1	$-jC_2$	C_3	jC_1	$-jC_2$	$-C_3$	$-jC_1$	jC_2	C_3	$-jC_1$	jC_2	$-C_3$	
0	1	jC_1	jC_3	C_2	jC_1	jC_3	$-C_2$	$-jC_1$	$-jC_3$	C_2	$-jC_1$	$-jC_3$	$-C_2$	
0	1	jC_1	$-jC_3$	C_2	jC_1	$-jC_3$	$-C_2$	$-jC_1$	jC_3	C_2	$-jC_1$	jC_3	$-C_2$	
0	1	jC_2	jC_3	C_1	jC_2	jC_3	$-C_1$	$-jC_2$	$-jC_3$	C_1	$-jC_2$	$-jC_3$	$-C_1$	
0	1	jC_2	$-jC_3$	C_1	jC_2	$-jC_3$	$-C_1$	$-jC_2$	jC_3	C_1	$-jC_2$	jC_3	$-C_1$	
1	2	C_4	C_5	C_6	C_4	C_5	$-C_6$	$-C_4$	$-C_5$	C_6	$-C_4$	$-C_5$	$-C_6$	t_2
1	2	C_4	$-C_5$	C_6	C_4	$-C_5$	$-C_6$	$-C_4$	C_5	C_6	$-C_4$	C_5	$-C_6$	
1	2	jC_4	jC_5	C_6	jC_4	jC_5	$-C_6$	$-jC_4$	$-jC_5$	C_6	$-jC_4$	$-jC_5$	$-C_6$	
1	2	jC_4	$-jC_5$	C_6	jC_4	$-jC_5$	$-C_6$	$-jC_4$	jC_5	C_6	$-jC_4$	jC_5	$-C_6$	
1	2	jC_4	jC_6	C_5	jC_4	jC_6	$-C_5$	$-jC_4$	$-jC_6$	C_5	$-jC_4$	$-jC_6$	$-C_5$	
1	2	jC_4	$-jC_6$	C_5	jC_4	$-jC_6$	$-C_5$	$-jC_4$	jC_6	C_5	$-jC_4$	jC_6	$-C_5$	
1	2	jC_5	jC_6	C_4	jC_5	jC_6	$-C_4$	$-jC_5$	$-jC_6$	C_4	$-jC_5$	$-jC_6$	$-C_4$	
1	2	jC_5	$-jC_6$	C_4	jC_5	$-jC_6$	$-C_4$	$-jC_5$	jC_6	C_4	$-jC_5$	jC_6	$-C_4$	
2	3	C_7	C_8	C_9	C_7	C_8	$-C_9$	$-C_7$	$-C_8$	C_9	$-C_7$	$-C_8$	$-C_9$	t_3
…	…	…	…	…	…	…	…	…	…	…	…	…	…	
2	3	jC_8	$-jC_9$	C_7	jC_8	$-jC_9$	$-C_7$	$-jC_8$	jC_9	C_7	$-jC_8$	jC_9	$-C_7$	
…	…	…	…	…	…	…	…	…	…	…	…	…	…	
31	32	C_7	C_9	C_{15}	C_7	C_9	$-C_{15}$	$-C_7$	$-C_9$	C_{15}	$-C_7$	$-C_9$	$-C_{15}$	t_{31}
31	32	C_7	$-C_9$	C_{15}	C_7	$-C_9$	$-C_{15}$	$-C_7$	C_9	C_{15}	$-C_7$	C_9	$-C_{15}$	
31	32	jC_7	jC_9	C_{15}	jC_7	jC_9	$-C_{15}$	$-jC_7$	$-jC_9$	C_{15}	$-jC_7$	$-jC_9$	$-C_{15}$	
31	32	jC_7	$-jC_9$	C_{15}	jC_7	$-jC_9$	$-C_{15}$	$-jC_7$	jC_9	C_{15}	$-jC_7$	jC_9	$-C_{15}$	
31	32	jC_7	jC_{15}	C_9	jC_7	jC_6	$-C_9$	$-jC_7$	$-jC_{15}$	C_9	$-jC_7$	$-jC_{15}$	$-C_9$	
31	32	jC_7	$-jC_{15}$	C_9	jC_7	$-jC_6$	$-C_9$	$-jC_7$	jC_{15}	C_9	$-jC_7$	jC_{15}	$-C_9$	
31	32	jC_9	jC_{15}	C_7	jC_9	jC_{15}	$-C_7$	$-jC_9$	$-jC_{15}$	C_7	$-jC_9$	$-jC_{15}$	$-C_7$	
31	32	jC_9	$-jC_{15}$	C_7	jC_9	$-jC_{15}$	$-C_7$	$-jC_9$	jC_{15}	C_7	$-jC_9$	jC_{15}	$-C_7$	

~~Code Group~~	~~Code Set~~	~~b_6~~	~~b_5~~	~~b_4~~	~~b_3~~	~~b_2~~	~~b_1~~	~~Associated toffset~~

0	1	1	1	1	1	1	1	
0	1	1	1	-1	1	1	1	
0	1	1	-1	1	1	1	1	
0	1	1	-1	-1	1	1	1	t_0
0	1	-1	1	1	1	1	1	
0	1	-1	1	-1	1	1	1	
0	1	-1	-1	1	1	1	1	
0	1	-1	-1	-1	1	1	1	
1	1	1	1	1	1	1	-1	
1	1	1	1	-1	1	1	-1	
1	1	1	-1	1	1	1	-1	
1	1	1	-1	-1	1	1	-1	
1	1	-1	1	1	1	1	-1	t_1
1	1	-1	1	-1	1	1	-1	
1	1	-1	-1	1	1	1	-1	
1	1	-1	-1	-1	1	1	-1	
2	1	1	1	1	1	-1	1	
2	1	1	1	-1	1	-1	1	
2	1	1	-1	1	1	-1	1	
2	1	1	-1	-1	1	-1	1	
2	1	-1	1	1	1	-1	1	t_2
2	1	-1	1	-1	1	-1	1	
2	1	-1	-1	1	1	-1	1	
2	1	-1	-1	-1	1	-1	1	
…	…	…	…	…	…	…	…	…
31	4	1	1	1	-1	-1	-1	
31	4	1	1	-1	-1	-1	-1	
31	4	1	-1	1	-1	-1	-1	
31	4	1	-1	-1	-1	-1	-1	
31	4	-1	1	1	-1	-1	-1	t_{31}
31	4	-1	1	-1	-1	-1	-1	
31	4	-1	-1	1	-1	-1	-1	
31	4	-1	-1	-1	-1	-1	-1	

I/Q selection represents the frame number within an interleaving period of 20 msec. It is 'I' for 'first frame' and 'Q' for 'second frame'.

Bit b_7 represents the slot position within the frame and it is '1' for 'slot k' and '-1' for 'slot k+8'.

Note that, the C_p and $\{C_1, \ldots, C_N\}$ are pairwise mutually orthogonal and $\{C_1, \ldots, C_N\}$ were chosen to have good aperiodic cross correlation properties with C_p.

7.2 Code Allocation

~~Sequences of 8~~The secondary SCH codes~~, thus composed of $C_{S,i}$ from~~ defined in ~~chapter~~Section 7.1 above, are used to transmit information on the PSCH. In general the information on the code group of a cell and on the frame timing (see TS 25.224, Section '6.6.1 Cell Search') is transmitted in the PSCH. According to TS 25.221 section '7.4 The Physical Synchronisation Channel (PSCH)', there is case (3) where additional information from SCH transport channel is to be transmitted in the PSCH.
~~The sequences of secondary SCH codes are constructed such that their cyclic-shifts are unique, i.e. a non-zero cyclic shift less than 8 of any of the sequences is not equivalent to some cyclic shift of any other of the sequences. Also, a non-zero cyclic shift less than 8 of any of the sequences is not equivalent to itself with any other cyclic shift less than 8. This property is used to uniquely determine the transmitted sequence in the receiver.~~
The evaluation of transmitted information on code group and frame timing is shown in table 9, where the 32 code groups are listed. Each code group is containing 4 specific scrambling codes, each scrambling code associated with a specific short and long basic midamble code.
Each code group is additionally linked to a specific t_{Offset}, thus to a specific frame timing. By using this scheme, the UE can derive the position of the frame border due to the position of the SCH sequence and the knowledge of t_{Offset}. ~~Positioning of the secondary SCH codes is depicted in the last line of table 10 and 11.~~
The complete mapping of Code Group to Scrambling Code, Midamble Codes and t_{Offset} is depicted in table 9, cf. also TS 25.231.

CELL PARA-METER	Code Group	Associated Codes			Associated t_{Offset}
		Scrambling Code	Long Basic Midamble Code	Short Basic Midamble Code	
0	Group 1	Code 0	m_{PL0}	m_{SL0}	t_0
1		Code 1	m_{PL1}	m_{SL1}	
2		Code 2	m_{PL2}	m_{SL2}	
3		Code 3	m_{PL3}	m_{SL3}	
4	Group 2	Code 4	m_{PL4}	m_{SL4}	t_1
5		Code 5	m_{PL5}	m_{SL5}	
6		Code 6	m_{PL6}	m_{SL6}	
7		Code 7	m_{PL7}	m_{SL7}	
. . . .					
124	Group 32	Code 124	m_{PL124}	m_{SL124}	t_{31}
125		Code 125	m_{PL125}	m_{SL125}	
126		Code 126	m_{PL126}	m_{SL126}	
127		Code 127	m_{PL127}	m_{SL127}	

Table 9 Mapping scheme for Cell Parameters, Code Groups, Scrambling Codes, Midambles and t_{Offset}.
For basic midamble codes m_P cf.TS 25.221, section '7.2.3.1 & 7.2.3.2 Midamble Sequences'.
For CELL PARAMETERS also cf. TS 25.231.

~~The following subchapters 7.2.1 and 7.2.2 are referring to the three cases of PSCH/CCPCH usage as described in TS 25.221 section 7.4.~~

7.2.1 Code allocation for case 1 and 2

In table 10 the 32 sequences used in the cases 1 and 2 of PSCH/CCPCH scheme are listed. Again, these are used to encode the 32 different code groups.

It should be mentioned that the sequences used here can be derived from FDD sequences by puncturing every 2nd position, thus a UE can use same database for FDD and TDD.

Code Group	Secondary SCH Code Position								Associated t_{Offset}
	#1	#2	#3	#4	#5	#6	#7	#8	
Group1	C_1	C_2	C_6	C_{15}	C_8	C_7	C_3	C_{11}	t_0
Group2	C_1	C_9	C_{10}	C_{13}	C_{11}	C_3	C_2	C_{16}	t_1
Group 3	C_1	C_{16}	C_{14}	C_{11}	C_{14}	C_{16}	C_1	C_4	t_2
Group 4	C_1	C_6	C_1	C_9	C_{17}	C_{12}	C_{17}	C_9	t_3
Group 5	C_1	C_{13}	C_5	C_7	C_3	C_8	C_{16}	C_{14}	t_4
Group 6	C_1	C_3	C_9	C_5	C_6	C_4	C_{15}	C_2	t_5
Group 7	C_1	C_{10}	C_{13}	C_3	C_9	C_{17}	C_{14}	C_7	t_6
Group 8	C_1	C_{17}	C_{17}	C_1	C_{12}	C_1	C_{13}	C_{12}	t_7
Group 9	C_1	C_7	C_4	C_{16}	C_{15}	C_9	C_{12}	C_{17}	t_8
Group 10	C_1	C_{14}	C_8	C_{14}	C_1	C_5	C_{11}	C_5	t_9
Group 11	C_1	C_4	C_{12}	C_{12}	C_4	C_1	C_{10}	C_{10}	t_{10}
Group 12	C_1	C_{11}	C_{16}	C_{10}	C_7	C_{14}	C_9	C_{15}	t_{11}
Group 13	C_1	C_1	C_3	C_8	C_{10}	C_{10}	C_8	C_3	t_{12}
Group 14	C_1	C_8	C_7	C_6	C_{13}	C_6	C_7	C_8	t_{13}
Group 15	C_1	C_{15}	C_{11}	C_4	C_{16}	C_2	C_6	C_{13}	t_{14}
Group 16	C_1	C_5	C_{15}	C_2	C_2	C_{15}	C_5	C_1	t_{15}
Group 17	C_1	C_{12}	C_2	C_{17}	C_5	C_{11}	C_4	C_6	t_{16}
Group 18	C_2	C_{11}	C_{14}	C_4	C_{10}	C_1	C_{15}	C_8	t_{17}
Group 19	C_2	C_1	C_1	C_2	C_{13}	C_{14}	C_{14}	C_{13}	t_{18}
Group 20	C_2	C_8	C_5	C_{17}	C_{16}	C_{10}	C_{13}	C_1	t_{19}
Group 21	C_2	C_{15}	C_9	C_{15}	C_2	C_6	C_{12}	C_6	t_{20}
Group 22	C_2	C_5	C_{13}	C_{13}	C_5	C_2	C_{11}	C_{11}	t_{21}
Group 23	C_2	C_{12}	C_{17}	C_{11}	C_8	C_{15}	C_{10}	C_{16}	t_{22}
Group 24	C_2	C_2	C_4	C_9	C_{11}	C_{11}	C_9	C_4	t_{23}
Group 25	C_2	C_9	C_8	C_7	C_{14}	C_7	C_8	C_9	t_{24}
Group 26	C_2	C_{16}	C_{12}	C_5	C_{17}	C_3	C_7	C_{14}	t_{25}
Group 27	C_2	C_6	C_{16}	C_3	C_3	C_{16}	C_6	C_2	t_{26}
Group 28	C_2	C_{13}	C_3	C_1	C_6	C_{12}	C_5	C_7	t_{27}
Group 29	C_2	C_3	C_7	C_{16}	C_9	C_8	C_4	C_{12}	t_{28}
Group 30	C_2	C_{10}	C_{11}	C_{14}	C_{12}	C_4	C_3	C_{17}	t_{29}
Group 31	C_2	C_{17}	C_{15}	C_{12}	C_{15}	C_{17}	C_2	C_5	t_{30}
Group 32	C_2	C_7	C_2	C_{10}	C_1	C_{13}	C_1	C_{10}	t_{31}
Frame position	Frame #1		Frame #2		Frame #3		Frame #4		

Table 10 Spreading Code allocation for Secondary SCH Code, case 2) of PSCH/CCPCH scheme

7.2.2 Code allocation for case 3

In table 11 the 256 sequences used in case 3 of PSCH/CCPCH scheme are listed. In addition to the information on code group three bits from SCH transport

~~channel are transmitted to the UE with these codes.~~

<Editors note: The usage of CCPCH pointing is for further study (cf. TDoc R1#2(99) 74)>

Code Group	Secondary PSCH Code at Position								Additional Bits from SCH Transport Channel	Associated t_{Offset}
	#1	#2	#3	#4	#5	#6	#7	#8		
Group 1	C2	C14	C6	C8	C4	C9	C17	C15	000	t_0
	C2	C4	C10	C6	C7	C5	C16	C3	001	
	C3	C3	C5	C10	C12	C12	C10	C5	010	
	C3	C10	C9	C8	C15	C8	C9	C10	011	
	C3	C17	C13	C6	C1	C4	C8	C15	100	
	C3	C7	C17	C4	C4	C17	C7	C3	101	
	C3	C14	C4	C2	C7	C13	C6	C8	110	
	C3	C4	C8	C17	C10	C9	C5	C13	111	
Group 2	C3	C11	C12	C15	C13	C5	C4	C1	000	t_1
	C3	C1	C16	C13	C16	C1	C3	C6	001	
	C3	C8	C3	C11	C2	C14	C2	C11	010	
	C3	C15	C7	C9	C5	C10	C1	C16	011	
	C3	C5	C11	C7	C8	C6	C17	C4	100	
	C3	C12	C15	C5	C11	C2	C16	C9	101	
	C3	C2	C2	C3	C14	C15	C15	C14	110	
	C3	C9	C6	C1	C17	C11	C14	C2	111	
Group 3	C3	C16	C10	C16	C3	C7	C13	C7	000	t_2
	C3	C6	C14	C14	C6	C3	C12	C12	001	
	C3	C13	C1	C12	C9	C16	C11	C17	010	
	C4	C12	C13	C16	C14	C6	C5	C2	011	
	C4	C2	C17	C14	C17	C2	C4	C7	100	
	C4	C9	C4	C12	C3	C15	C3	C12	101	
	C4	C16	C8	C10	C6	C11	C2	C17	110	
	C4	C6	C12	C8	C9	C7	C1	C5	111	
Group 4	C4	C13	C16	C6	C12	C3	C17	C10	000	t_3
	C4	C3	C3	C4	C15	C16	C16	C15	001	
	C4	C10	C7	C2	C1	C12	C15	C3	010	
	C4	C17	C11	C17	C4	C8	C14	C8	011	
	C4	C7	C15	C15	C7	C4	C13	C13	100	
	C4	C14	C2	C13	C10	C17	C12	C1	101	
	C4	C4	C6	C11	C13	C13	C11	C6	110	
	C4	C11	C10	C9	C16	C9	C10	C11	111	
Group 5	C4	C1	C14	C7	C2	C5	C9	C16	000	t_4
	C4	C8	C1	C5	C5	C1	C8	C4	001	
	C4	C15	C5	C3	C8	C14	C7	C9	010	
	C4	C5	C9	C1	C11	C10	C6	C14	011	
	C5	C4	C4	C5	C16	C17	C17	C16	100	
	C5	C11	C8	C3	C2	C13	C16	C4	101	
	C5	C1	C12	C1	C5	C9	C15	C9	110	
	C5	C8	C16	C16	C8	C5	C14	C14	111	
Group 6	C5	C15	C3	C14	C11	C1	C13	C2	000	t_5
	C5	C5	C7	C12	C14	C14	C12	C7	001	
	C5	C12	C11	C10	C17	C10	C11	C12	010	
	C5	C2	C15	C8	C3	C6	C10	C17	011	
	C5	C9	C2	C6	C6	C2	C9	C5	100	
	C5	C16	C6	C4	C9	C15	C8	C10	101	
	C5	C6	C10	C2	C12	C11	C7	C15	110	
	C5	C13	C14	C17	C15	C7	C6	C3	111	

Group7	C5	C3	C1	C15	C1	C3	C5	C8	000	t_6
	C5	C10	C5	C13	C4	C16	C4	C13	001	
	C5	C17	C9	C11	C7	C12	C3	C1	010	
	C5	C7	C13	C9	C10	C8	C2	C6	011	
	C5	C14	C17	C7	C13	C4	C1	C11	100	
	C6	C13	C12	C11	C1	C11	C12	C13	101	
	C6	C3	C16	C9	C4	C7	C11	C1	110	
	C6	C10	C3	C7	C7	C3	C10	C6	111	
Group 8	C6	C17	C7	C5	C10	C16	C9	C11	000	t_7
	C6	C7	C11	C3	C13	C12	C8	C16	001	
	C6	C14	C15	C1	C16	C8	C7	C4	010	
	C6	C4	C2	C16	C2	C4	C6	C9	011	
	C6	C11	C6	C14	C5	C17	C5	C14	100	
	C6	C1	C10	C12	C8	C13	C4	C2	101	
	C6	C8	C14	C10	C11	C9	C3	C7	110	
	C6	C15	C1	C8	C14	C5	C2	C12	111	
Group 9	C6	C5	C5	C6	C17	C1	C1	C17	000	t_8
	C6	C12	C9	C4	C3	C14	C17	C5	001	
	C6	C2	C13	C2	C6	C10	C16	C10	010	
	C6	C9	C17	C17	C9	C6	C15	C15	011	
	C6	C16	C4	C15	C12	C2	C14	C3	100	
	C6	C6	C8	C13	C15	C15	C13	C8	101	
	C7	C5	C3	C17	C3	C5	C7	C10	110	
	C7	C12	C7	C15	C6	C1	C6	C15	111	
Group 10	C7	C2	C11	C13	C9	C14	C5	C3	000	t_9
	C7	C9	C15	C11	C12	C10	C4	C8	001	
	C7	C16	C2	C9	C15	C6	C3	C13	010	
	C7	C6	C6	C7	C1	C2	C2	C1	011	
	C7	C13	C10	C5	C4	C15	C1	C6	100	
	C7	C3	C14	C3	C7	C11	C17	C11	101	
	C7	C10	C1	C1	C10	C7	C16	C16	110	
	C7	C17	C5	C16	C13	C3	C15	C4	111	
Group 11	C7	C7	C9	C14	C16	C16	C14	C9	000	t_{10}
	C7	C14	C13	C12	C2	C12	C13	C14	001	
	C7	C4	C17	C10	C5	C8	C12	C2	010	
	C7	C11	C4	C8	C8	C4	C11	C7	011	
	C7	C1	C8	C6	C11	C17	C10	C12	100	
	C7	C8	C12	C4	C14	C13	C9	C17	101	
	C7	C15	C16	C2	C17	C9	C8	C5	110	
	C8	C14	C11	C6	C5	C16	C2	C7	111	
Group 12	C8	C4	C15	C4	C8	C12	C1	C12	000	t_{11}
	C8	C11	C2	C2	C11	C8	C17	C17	001	
	C8	C1	C6	C17	C14	C4	C16	C5	010	
	C8	C8	C10	C15	C17	C17	C15	C10	011	
	C8	C15	C14	C13	C3	C13	C14	C15	100	
	C8	C5	C1	C11	C6	C9	C13	C3	101	
	C8	C12	C5	C9	C9	C5	C12	C8	110	
	C8	C2	C9	C7	C12	C1	C11	C13	111	
Group 13	C8	C9	C13	C5	C15	C14	C10	C1	000	t_{12}
	C8	C16	C17	C3	C1	C10	C9	C6	001	
	C8	C6	C4	C1	C4	C6	C8	C11	010	
	C8	C13	C8	C16	C7	C2	C7	C16	011	
	C8	C3	C12	C14	C10	C15	C6	C4	100	

	C8	C10	C16	C12	C13	C11	C5	C9	101	
	C8	C17	C3	C10	C16	C7	C4	C14	110	
	C8	C7	C7	C8	C2	C3	C3	C2	111	
Group 14	C9	C6	C2	C12	C7	C10	C14	C4	000	t_{13}
	C9	C13	C6	C10	C10	C6	C13	C9	001	
	C9	C3	C10	C8	C13	C2	C12	C14	010	
	C9	C10	C14	C6	C16	C15	C11	C2	011	
	C9	C17	C1	C4	C2	C11	C10	C7	100	
	C9	C7	C5	C2	C5	C7	C9	C12	101	
	C9	C14	C9	C17	C8	C3	C8	C17	110	
	C9	C4	C13	C15	C11	C16	C7	C5	111	
Group 15	C9	C11	C17	C13	C14	C12	C6	C10	000	t_{14}
	C9	C1	C4	C11	C17	C8	C5	C15	001	
	C9	C8	C8	C9	C3	C4	C4	C3	010	
	C9	C15	C12	C7	C6	C17	C3	C8	011	
	C9	C5	C16	C5	C9	C13	C2	C13	100	
	C9	C12	C3	C3	C12	C9	C1	C1	101	
	C9	C2	C7	C1	C15	C5	C17	C6	110	
	C9	C9	C11	C16	C1	C1	C16	C11	111	
Group 16	C9	C16	C15	C14	C4	C14	C15	C16	000	t_{15}
	C10	C15	C10	C1	C9	C4	C9	C1	001	
	C10	C5	C14	C16	C12	C17	C8	C6	010	
	C10	C12	C1	C14	C15	C13	C7	C11	011	
	C10	C2	C5	C12	C1	C9	C6	C16	100	
	C10	C9	C9	C10	C4	C5	C5	C4	101	
	C10	C16	C13	C8	C7	C1	C4	C9	110	
	C10	C6	C17	C6	C10	C14	C3	C14	111	
Group 17	C10	C13	C4	C4	C13	C10	C2	C2	000	t_{16}
	C10	C3	C8	C2	C16	C6	C1	C7	001	
	C10	C10	C12	C17	C2	C2	C17	C12	010	
	C10	C17	C16	C15	C5	C15	C16	C17	011	
	C10	C7	C3	C13	C8	C11	C15	C5	100	
	C10	C14	C7	C11	C11	C7	C14	C10	101	
	C10	C4	C11	C9	C14	C3	C13	C15	110	
	C10	C11	C15	C7	C17	C16	C12	C3	111	
Group 18	C10	C1	C2	C5	C3	C12	C11	C8	000	t_{17}
	C10	C8	C6	C3	C6	C8	C10	C13	001	
	C11	C7	C1	C7	C11	C15	C4	C15	010	
	C11	C14	C5	C5	C14	C11	C3	C3	011	
	C11	C4	C9	C3	C17	C7	C2	C8	100	
	C11	C11	C13	C1	C3	C3	C1	C13	101	
	C11	C1	C17	C16	C6	C16	C17	C1	110	
	C11	C8	C4	C14	C9	C12	C16	C6	111	
Group 19	C11	C15	C8	C12	C12	C8	C15	C11	000	t_{18}
	C11	C5	C12	C10	C15	C4	C14	C16	001	
	C11	C12	C16	C8	C1	C17	C13	C4	010	
	C11	C2	C3	C6	C4	C13	C12	C9	011	
	C11	C9	C7	C4	C7	C9	C11	C14	100	
	C11	C16	C11	C2	C10	C5	C10	C2	101	
	C11	C6	C15	C17	C13	C1	C9	C7	110	
	C11	C13	C2	C15	C16	C14	C8	C12	111	
Group 20	C11	C3	C6	C13	C2	C10	C7	C17	000	t_{19}
	C11	C10	C10	C11	C5	C6	C6	C5	001	

	C11	C17	C14	C9	C8	C2	C5	C10	010	
	C12	C16	C9	C13	C13	C9	C16	C12	011	
	C12	C6	C13	C11	C16	C5	C15	C17	100	
	C12	C13	C17	C9	C2	C1	C14	C5	101	
	C12	C3	C4	C7	C5	C14	C13	C10	110	
	C12	C10	C8	C5	C8	C10	C12	C15	111	
Group 21	C12	C17	C12	C3	C11	C6	C11	C3	000	t_{20}
	C12	C7	C16	C1	C14	C2	C10	C8	001	
	C12	C14	C3	C16	C17	C15	C9	C13	010	
	C12	C4	C7	C14	C3	C11	C8	C1	011	
	C12	C11	C11	C12	C6	C7	C7	C6	100	
	C12	C1	C15	C10	C9	C3	C6	C11	101	
	C12	C8	C2	C8	C12	C16	C5	C16	110	
	C12	C15	C6	C6	C15	C12	C4	C4	111	
Group 22	C12	C5	C10	C4	C1	C8	C3	C9	000	t_{21}
	C12	C12	C14	C2	C4	C4	C2	C14	001	
	C12	C2	C1	C17	C7	C17	C1	C2	010	
	C12	C9	C5	C15	C10	C13	C17	C7	011	
	C13	C8	C17	C2	C15	C3	C11	C9	100	
	C13	C15	C4	C17	C1	C16	C10	C14	101	
	C13	C5	C8	C15	C4	C12	C9	C2	110	
	C13	C12	C12	C13	C7	C8	C8	C7	111	
Group 23	C13	C2	C16	C11	C10	C4	C7	C12	000	t_{22}
	C13	C9	C3	C9	C13	C17	C6	C17	001	
	C13	C16	C7	C7	C16	C13	C5	C5	010	
	C13	C6	C11	C5	C2	C9	C4	C10	011	
	C13	C13	C15	C3	C5	C5	C3	C15	100	
	C13	C3	C2	C1	C8	C1	C2	C3	101	
	C13	C10	C6	C16	C11	C14	C1	C8	110	
	C13	C17	C10	C14	C14	C10	C17	C13	111	
Group 24	C13	C7	C14	C12	C17	C6	C16	C1	000	t_{23}
	C13	C14	C1	C10	C3	C2	C15	C6	001	
	C13	C4	C5	C8	C6	C15	C14	C11	010	
	C13	C11	C9	C6	C9	C11	C13	C16	011	
	C13	C1	C13	C4	C12	C7	C12	C4	100	
	C14	C17	C8	C8	C17	C14	C6	C6	101	
	C14	C7	C12	C6	C3	C10	C5	C11	110	
	C14	C14	C16	C4	C6	C6	C4	C16	111	
Group 25	C14	C4	C3	C2	C9	C2	C3	C4	000	t_{24}
	C14	C11	C7	C17	C12	C15	C2	C9	001	
	C14	C1	C11	C15	C15	C11	C1	C14	010	
	C14	C8	C15	C13	C1	C7	C17	C2	011	
	C14	C15	C2	C11	C4	C3	C16	C7	100	
	C14	C5	C6	C9	C7	C16	C15	C12	101	
	C14	C12	C10	C7	C10	C12	C14	C17	110	
	C14	C2	C14	C5	C13	C8	C13	C5	111	
Group 26	C14	C9	C1	C3	C16	C4	C12	C10	000	t_{25}
	C14	C16	C5	C1	C2	C17	C11	C15	001	
	C14	C6	C9	C16	C5	C13	C10	C3	010	
	C14	C13	C13	C14	C8	C9	C9	C8	011	
	C14	C3	C17	C12	C11	C5	C8	C13	100	
	C14	C10	C4	C10	C14	C1	C7	C1	101	
	C15	C9	C16	C14	C2	C8	C1	C3	110	

	C15	C16	C3	C12	C5	C4	C17	C8	111	
Group 27	C15	C6	C7	C10	C8	C17	C16	C13	000	t_{26}
	C15	C13	C11	C8	C11	C13	C15	C1	001	
	C15	C3	C15	C6	C14	C9	C14	C6	010	
	C15	C10	C2	C4	C17	C5	C13	C11	011	
	C15	C17	C6	C2	C3	C1	C12	C16	100	
	C15	C7	C10	C17	C6	C14	C11	C4	101	
	C15	C14	C14	C15	C9	C10	C10	C9	110	
	C15	C4	C1	C13	C12	C6	C9	C14	111	
Group 28	C15	C11	C5	C11	C15	C2	C8	C2	000	t_{27}
	C15	C1	C9	C9	C1	C15	C7	C7	001	
	C15	C8	C13	C7	C4	C11	C6	C12	010	
	C15	C15	C17	C5	C7	C7	C5	C17	011	
	C15	C5	C4	C3	C10	C3	C4	C5	100	
	C15	C12	C8	C1	C13	C16	C3	C10	101	
	C15	C2	C12	C16	C16	C12	C2	C15	110	
	C16	C1	C7	C3	C4	C2	C13	C17	111	
Group 29	C16	C8	C11	C1	C7	C15	C12	C5	000	t_{28}
	C16	C15	C15	C16	C10	C11	C11	C10	001	
	C16	C5	C2	C14	C13	C7	C10	C15	010	
	C16	C12	C6	C12	C16	C3	C9	C3	011	
	C16	C2	C10	C10	C2	C16	C8	C8	100	
	C16	C9	C14	C8	C5	C12	C7	C13	101	
	C16	C16	C1	C6	C8	C8	C6	C1	110	
	C16	C6	C5	C4	C11	C4	C5	C6	111	
Group 30	C16	C13	C9	C2	C14	C17	C4	C11	000	t_{29}
	C16	C3	C13	C17	C17	C13	C3	C16	001	
	C16	C10	C17	C15	C3	C9	C2	C4	010	
	C16	C17	C4	C13	C6	C5	C1	C9	011	
	C16	C7	C8	C11	C9	C1	C17	C14	100	
	C16	C14	C12	C9	C12	C14	C16	C2	101	
	C16	C4	C16	C7	C15	C10	C15	C7	110	
	C16	C11	C3	C5	C1	C6	C14	C12	111	
Group 31	C17	C10	C15	C9	C6	C13	C8	C14	000	t_{30}
	C17	C17	C2	C7	C9	C9	C7	C2	001	
	C17	C7	C6	C5	C12	C5	C6	C7	010	
	C17	C14	C10	C3	C15	C1	C5	C12	011	
	C17	C4	C14	C1	C1	C14	C4	C17	100	
	C17	C11	C1	C16	C4	C10	C3	C5	101	
	C17	C1	C5	C14	C7	C6	C2	C10	110	
	C17	C8	C9	C12	C10	C2	C1	C15	111	
Group 32	C17	C15	C13	C10	C1	C15	C17	C3	000	t_{31}
	C17	C5	C17	C8	C16	C11	C16	C8	001	
	C17	C12	C4	C6	C2	C7	C15	C13	010	
	C17	C2	C8	C4	C5	C3	C14	C1	011	
	C17	C9	C12	C2	C8	C16	C13	C6	100	
	C17	C16	C16	C17	C11	C12	C12	C11	101	
	C17	C6	C3	C15	C14	C8	C11	C16	110	
	C17	C13	C7	C13	C17	C4	C10	C4	111	
Frame position	Frame #1		Frame #2		Frame #3		Frame #4			

Table 11 Spreading Code allocation for Secondary SCH Code, case 3) of

Text Proposal for 25.224

4.5 Synchronisation and Cell Search Procedures

4.5.1 Cell Search

During the initial cell search, the UE searches for a cell. It then determines the midamble, the downlink scrambling code and frame synchronisation of that cell. The initial cell search uses the Physical Synchronisation Channel (PSCH) described in S1.21TS 25.221. The generation of synchronisation codes is described in S1.23TS 25.223.

This initial cell search is carried out in three steps:

Step 1: Slot synchronisation

During the first step of the initial cell search procedure the UE uses the primary synchronisation code c_p to acquire slot synchronisation to the strongest cell. Furthermore, frame synchronisation with the uncertainty of 1 out of 2 is obtained in this step. A single matched filter (or any similar device) is used for this purpose, that is matched to the primary synchronisation code which is common to all cells. The procedure is according to the description for the FDD mode in S1.14.

Step 2: Frame synchronisation and code-group identification

The Step 2 is described for the case where PSCH and CCPCH are in timeslot k and k+8 with k=0…76.

During the second step of the initial cell search procedure, the UE uses the sequence of modulated Secondary Synchronisation Codes to find frame synchronisation and identify one of 32 code groups. Each code group is linked to a specific t_{Offset}, thus to a specific frame timing, and is containing 4 specific scrambling codes. Each scrambling code is associated with a specific short and long basic midamble code.

In Cases 2 and 3 it is required to detect the position of the next synchronization slots. To detect the position of the next synchronization slots, the primary synchronization code is correlated with the received signal at offsets of 7 and 8 time slots from the position of the primary code that was detected in Step 1.

Then, the received signal at the positions of the synchronization codes is correlated with the primary synchronization Code C_p and the secondary synchronization codes $\{C_1,\ldots,C_N\}$. Note that the correlations are performed coherently forover M time slots, where at each slot a phase correction is provided by the correlation with the primary code. The minimal number of time slots is M=21, and the performance improves with increasing M.

The detection of secondary synchronisation sequence is done by correlating the received signal at the positions of the Secondary Synchronisation Code with all possible sequences of Secondary Synchronisation Codes, similar to FDD Mode. After four frames a sequence of eight codes is available providing all necessary information described above. Nevertheless, it should be noted that due to the special coding already three codes show the sequence unambiguously, i.e. a UE can determine the whole sequence when three codes have been received.

Step 3: Scrambling code identification

During the third and last step of the initial cell-search procedure, the UE determines the exact basic midamble code and the accompanying scrambling code used by the found cell. They are

identified through correlation over the CCPCH with all four midambles of the code group identified in the second step . Thus the third step is a one out of four decision.
This step is taking into account that the CCPCH containing the BCH is transmitted using the first spreading code ($a_{Q=16}^{(h=1)}$ in figure 2 of ~~S1.23~~TS 25.223 section '6.2 Spreading Codes') and using the first midamble $\mathbf{m}^{(1)}$ (derived from basic midamble code $\mathbf{m}_P$, cf. ~~S1.21~~TS 25.221 section '7.2.3 Training sequences for spread bursts'). Thus CCPCH code and midamble can be immediately derived when knowing scrambling code and basic midamble code.