TSG-RAN Working Group 1 meeting #7bis
Kyongju, South Korea
October 4 – October 5, 1999

TSGR1#7bis(99)e59

Agenda item:

Source: Ericsson

Title: Clarification of RACH preamble generation

Document for: Decision

Text proposal for TS 25.213 V1.3.1

4.3.3 Random access codes

4.3.3.1 Preamble codes

4.3.3.1.1 Preamble code construction

The random access preamble code $C_{pre,n,s}$ is a complex valued sequence. It is built from a real valued preamble scrambling code $C_{RACH,n}$ and a real valued preamble signature code $C_{sig,s}$ as follows:

$$C_{pre,n,s}(k) = C_{RACH,n}(k) \times C_{sig,s}(k) \times e^{j(\frac{\pi}{4}+\frac{\pi}{2}k)},\ k = 0, 1, 2, 3, \ldots, 4095,$$

where $k = 0$ corresponds to the chip transmitted first in time and $C_{RACH,n}$ and $C_{sig,s}$ are defined in 4.3.3.1.2 and 4.3.3.1.3 below respectively.

4.3.3.1.2 Preamble scrambling code

The preamble scrambling code ~~for the preamble part is as follows~~is a real code that is generated in a similar way as the long scrambling codes for dedicated channels.

~~The code generating method is the same as for the real part of the long codes on dedicated channels. Only the first 4096 chips of the code are used for preamble spreading with the chip rate of 3.84 Mchip/s. The long code c1 for the in-phase component is used directly on both in phase and quadrature branches without offset between branches.~~ The preamble scrambling code is defined as the position wise modulo 2 sum of 4096 chips segments of two binary m-sequences generated by means of two generator polynomials of degree 25. Let x and y be the two m-sequences respectively. The x sequence is constructed using the primitive (over GF(2)) polynomial $X^{25}+X^3+1$. The y sequence is constructed using the polynomial $X^{25}+X^3+X^2+X+1$. The resulting sequences thus constitute segments of a set of Gold sequences.

Let $n_{23} \ldots n_0$ be the binary representation of the code number n (decimal) with n_0 being the least significant bit. Code numbers between 0 and 255 are used for the random access channel. The m-sequences x_n and y are constructed as:

Initial conditions:

$x_n(0)=n_0$, $x_n(1)= n_1$, … $=x_n(22)= n_{22}$, $x_n(23)= n_{23}$, $x_n(24)=1$

$y(0)=y(1)= \ldots =y(23)= y(24)=1$

Recursive definition of subsequent symbols:

$x_n(i+25) =x_n(i+3) + x_n(i)$ modulo 2, $i=0,\ldots, 4070$,

$y(i+25) = y(i+3)+y(i+2)+y(i+1)+y(i)$ *modulo 2*, $i=0,\ldots, 4070$.

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

$z_n(i) = x_n(i)+y(i)$, $i = 0, 1, 2, \ldots, 4095$,

~~$C_{RACH,n} = \langle x_n(0)+y(0), x_n(1)+y(1), \ldots, x_n(4095)+y(4095) \rangle$,~~

where a~~A~~ll sums of symbols are taken modulo 2.

The preamble spreading code is described in Figure 1.

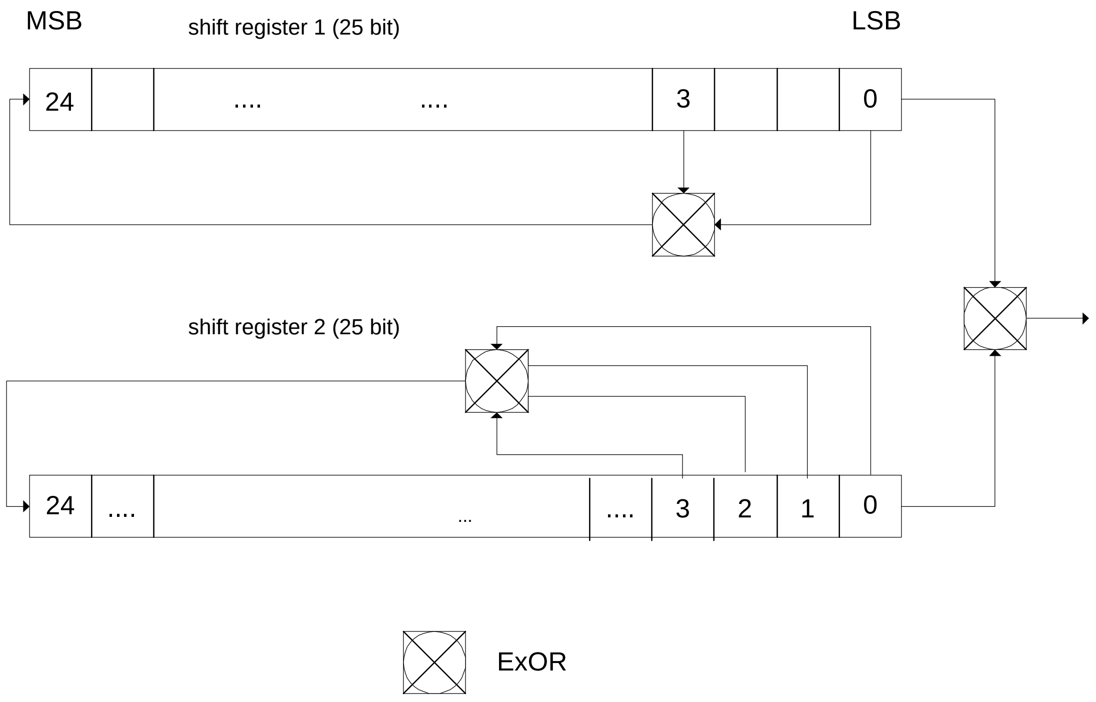


Figure 1. Preamble scrambling code generator

~~Before transmission these binary code words are converted to real valued sequences by the transformation '0' -> '+1', '1' -> '-1'.~~

Now, the real valued code $C_{RACH,n}$ is defined as follows:

$$C_{RACH,n}(i) = \begin{cases} 1 & if \ \ z_n(i) = 0 \\ -1 & if \ \ z_n(i) = 1 \end{cases} \quad i = 0,1,\ldots,4095.$$

4.3.3.2 Preamble signature code

The preamble signature code~~part~~ $C_{sig,s}$ corresponding to signature *s* consists of 256 repetitions of a length 16 signature $P_s(n)$, $n = 0, 1, \ldots, 15$~~, <P0,P1,…,P15>~~. Therefore,

$C_{sig,s}(i) = P_s(i \text{ modulo } 16)$, $i = 0, 1, \ldots, 4095$.

~~Before scrambling the preamble is therefore~~

$$\cancel{P_0, P_1, \cdots, P_{15}, P_0, P_1, \cdots, P_{15}, \cdots\cdots, P_0, P_1, \cdots, P_{15}}$$

The signature $P_s(n)$ is from the set of 16 Hadamard codes of length 16. These are listed in Table XXX ~~Table 3~~.

	~~Preamble symbols~~															
~~Signature~~	~~P_0~~	~~P_1~~	~~P_2~~	~~P_3~~	~~P_4~~	~~P_5~~	~~P_6~~	~~P_7~~	~~P_8~~	~~P_9~~	~~P_{10}~~	~~P_{11}~~	~~P_{12}~~	~~P_{13}~~	~~P_{14}~~	~~P_{15}~~
~~1~~	~~A~~	~~A~~	~~A~~	~~A~~	~~A~~	~~A~~	~~A~~	~~A~~	~~A~~	~~A~~	~~A~~	~~A~~	~~A~~	~~A~~	~~A~~	~~A~~
~~2~~	~~A~~	~~-A~~	~~A~~	~~-A~~	~~A~~	~~-A~~	~~A~~	~~-A~~	~~A~~	~~-A~~	~~A~~	~~-A~~	~~A~~	~~-A~~	~~A~~	~~-A~~
~~3~~	~~A~~	~~A~~	~~-A~~	~~-A~~	~~A~~	~~A~~	~~-A~~	~~-A~~	~~A~~	~~A~~	~~-A~~	~~-A~~	~~A~~	~~A~~	~~-A~~	~~-A~~
~~4~~	~~A~~	~~-A~~	~~-A~~	~~A~~	~~A~~	~~-A~~	~~-A~~	~~A~~	~~A~~	~~-A~~	~~-A~~	~~A~~	~~A~~	~~-A~~	~~-A~~	~~A~~
~~5~~	~~A~~	~~A~~	~~A~~	~~A~~	~~-A~~	~~-A~~	~~-A~~	~~-A~~	~~A~~	~~A~~	~~A~~	~~A~~	~~-A~~	~~-A~~	~~-A~~	~~-A~~
~~6~~	~~A~~	~~-A~~	~~A~~	~~-A~~	~~-A~~	~~A~~	~~-A~~	~~A~~	~~A~~	~~-A~~	~~A~~	~~-A~~	~~-A~~	~~A~~	~~-A~~	~~A~~
~~7~~	~~A~~	~~A~~	~~-A~~	~~-A~~	~~-A~~	~~-A~~	~~A~~	~~A~~	~~A~~	~~A~~	~~-A~~	~~-A~~	~~-A~~	~~-A~~	~~A~~	~~A~~
~~8~~	~~A~~	~~-A~~	~~-A~~	~~A~~	~~-A~~	~~A~~	~~A~~	~~-A~~	~~A~~	~~-A~~	~~-A~~	~~A~~	~~-A~~	~~A~~	~~A~~	~~-A~~
~~9~~	~~A~~	~~A~~	~~A~~	~~A~~	~~A~~	~~A~~	~~A~~	~~A~~	~~-A~~	~~-A~~	~~-A~~	~~-A~~	~~-A~~	~~-A~~	~~-A~~	~~-A~~
~~10~~	~~A~~	~~-A~~	~~A~~	~~-A~~	~~A~~	~~-A~~	~~A~~	~~-A~~	~~-A~~	~~A~~	~~-A~~	~~A~~	~~-A~~	~~A~~	~~-A~~	~~A~~
~~11~~	~~A~~	~~A~~	~~-A~~	~~-A~~	~~A~~	~~A~~	~~-A~~	~~-A~~	~~-A~~	~~-A~~	~~A~~	~~A~~	~~-A~~	~~-A~~	~~A~~	~~A~~
~~12~~	~~A~~	~~-A~~	~~-A~~	~~A~~	~~A~~	~~-A~~	~~-A~~	~~A~~	~~-A~~	~~A~~	~~A~~	~~-A~~	~~-A~~	~~A~~	~~A~~	~~-A~~
~~13~~	~~A~~	~~A~~	~~A~~	~~A~~	~~-A~~	~~-A~~	~~-A~~	~~-A~~	~~-A~~	~~-A~~	~~-A~~	~~-A~~	~~A~~	~~A~~	~~A~~	~~A~~
~~14~~	~~A~~	~~-A~~	~~A~~	~~-A~~	~~-A~~	~~A~~	~~-A~~	~~A~~	~~-A~~	~~A~~	~~-A~~	~~A~~	~~A~~	~~-A~~	~~A~~	~~-A~~
~~15~~	~~A~~	~~A~~	~~-A~~	~~-A~~	~~-A~~	~~-A~~	~~A~~	~~A~~	~~-A~~	~~-A~~	~~A~~	~~A~~	~~A~~	~~A~~	~~-A~~	~~-A~~
~~16~~	~~A~~	~~-A~~	~~-A~~	~~A~~	~~-A~~	~~A~~	~~A~~	~~-A~~	~~-A~~	~~A~~	~~A~~	~~-A~~	~~A~~	~~-A~~	~~-A~~	~~A~~

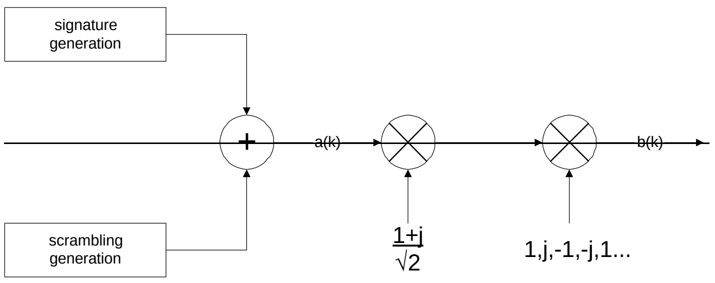
Preamble signature	Value of *n*															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$P_0(n)$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$P_1(n)$	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1
$P_2(n)$	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1
$P_3(n)$	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1
$P_4(n)$	1	1	1	1	-1	-1	-1	-1	1	1	1	1	-1	-1	-1	-1
$P_5(n)$	1	-1	1	-1	-1	1	-1	1	1	-1	1	-1	-1	1	-1	1
$P_6(n)$	1	1	-1	-1	-1	-1	1	1	1	1	-1	-1	-1	-1	1	1
$P_7(n)$	1	-1	-1	1	-1	1	1	-1	1	-1	-1	1	-1	1	1	-1
$P_8(n)$	1	1	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1
$P_9(n)$	1	-1	1	-1	1	-1	1	-1	-1	1	-1	1	-1	1	-1	1
$P_{10}(n)$	1	1	-1	-1	1	1	-1	-1	-1	-1	1	1	-1	-1	1	1
$P_{11}(n)$	1	-1	-1	1	1	-1	-1	1	-1	1	1	-1	-1	1	1	-1
$P_{12}(n)$	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	1	1	1	1
$P_{13}(n)$	1	-1	1	-1	-1	1	-1	1	-1	1	-1	1	1	-1	1	-1
$P_{14}(n)$	1	1	-1	-1	-1	-1	1	1	-1	-1	1	1	1	1	-1	-1
$P_{15}(n)$	1	-1	-1	1	-1	1	1	-1	-1	1	1	-1	1	-1	-1	1

Table 1. ~~Preamble~~ Length 16 signatures $P_s(n)$ corresponding to signature s.

~~The value of A = +1 in bipolar representation which is equivalent to 0 in boolean representation.~~

~~4.3.3.3 Preamble PAPR reduction~~

~~In order to reduce the PAPR during RACH preamble transmission the following technique is used.~~



~~**Figure 8 - Baseband modulator for RACH preamble.**~~

~~The binary preamble *a*(*k*) is modulated to get the complex valued preamble *b*(*k*),~~

~~$b(k) = a(k)\, e^{j(\frac{\pi}{4}+\frac{\pi}{2}k)}$, $k = 0, 1, 2, 3, \ldots, 4095$.~~

---- SNIP ----

Numbering of following paragraphs needs updating!