TSG-RAN Working Group 1(Radio) meeting #5 ***TSGR1#5(99)587***
Cheju, Korea, 01 – 04 June 1999

Agenda Item:

Source: **Nokia**

Title: **Text proposal for uplink long scrambling codes**

Document for: **Discussion**

Introduction

AdHoc10 in WG1 #4, Shin-Yokohama, agreed use of lower degree polynomials than 41 in uplink. Polynomials of degree less than 32 fit well to 32 bit DSP architecture. Nokia suggests two 25 degree polynomials to replace the two 41 degree polynomials in uplink. There are sufficiently many long scrambling codes in the new set to use.

The total number of uplink long scrambling codes is proposed to be 2^{24} = 16 777 216 which means that 24 bits are needed for signalling between a UE and a BS.

The value 16 777 232 of the phase shift parameter makes it possible to use the architecture of Fig. 1 below for uplink long scrambling code generators if desired, see also R1-99393. In particular, the quadrature component can be computed by three taps per each shift register, which reduces the complexity of code generators considerably. This is a property of the proposed polynomials and does not necessarily hold for other polynomials in the same way. The same phase shift value is applied to all chips rates. The proposed value for the phase shift parameter also guarantees that the in-phase and the quadrature component have a separate segment of a Gold code.

Text proposal for 25.213

4.3.2.2 Long scrambling code

The long scrambling codes are formed as described in Section ~~6~~4.3.2, where c_1 and c_2 are constructed as the position wise modulo 2 sum of 40960 chip segments of two binary *m*-sequences generated by means of two generator polynomials of degree ~~41~~25. Let *x*, and *y* be the two *m*-sequences respectively. The *x* sequence is constructed using the primitive (over GF(2)) polynomial ~~$1+X^3+X^{41}$~~ $X^{25}+X^3+1$. The *y* sequence is constructed using the polynomial ~~$1+X^{20}+X^{41}$~~ $X^{25}+X^3+X^2+X+1$. The resulting sequences thus constitute segments of a set of Gold sequences.

The code, c_2, used in generating the quadrature component of the complex spreading code is a ~~875 µs~~ 16 777 232 chips shifted version of the code, c_1, used in generating the in phase component.

The uplink scrambling code word has a period of one radio frame of 10 ms.

Let $n_{~~40~~23} \ldots n_0$ be the 24 bit binary representation of the scrambling code number *n* (decimal), $n=0,1,2,\ldots,2^{24}-1$ with n_0 being the least significant bit. The *x* sequence depends on the chosen scrambling code number *n* and is denoted x_n, in the sequel. Furthermore, let $x_n(i)$ and $y(i)$ denote the *i*:th symbol of the sequence x_n and *y*, respectively

The *m*-sequences x_n and *y* are constructed as:

Initial conditions:

$x_n(0)=n_0$, $x_n(1)= n_1$, … $=x_n(~~39~~22)= n_{~~39~~22}$, $x_n(~~40~~23)= n_{~~40~~23}$, $x_n(24)=1$

$y(0)=y(1)= \ldots =y(~~39~~23)= y(~~40~~24)=1$

Recursive definition of subsequent symbols:

$x_n(i+~~41~~25) =x_n(i+3) + x_n(i)$ *modulo 2*, $i=0, 1, 2, \ldots, 2^{~~41~~25}-~~43~~27$,

$y(i+~~41~~25) = y(i+3)+y(i+2~~0~~)+ y(i+1)+y(i)$ *modulo 2*, $i=0, 1, 2, \ldots, 2^{~~41~~25}-~~43~~27$.

The definition of the *n*:th scrambling code word for the in phase and quadrature components follows as (the left most index correspond to the chip scrambled first in each radio frame):

$c_{1,n} = < x_n(0)+y(0), x_n(1)+y(1), \ldots, x_n(N-1)+y(N-1) >$,

$c_{2,n} = < x_n(M)+y(M), x_n(M+1)+y(M+1), \ldots, x_n(M+N-1) + y(M+N-1) >$,

again all sums being modulo 2 additions. (Both N and M are defined in Table 1.)

These binary code words are converted to real valued sequences by the transformation ‘0’ -> ‘+1’, ‘1’ ->

<Editor's note: $2^{24}-1$ is FFS>

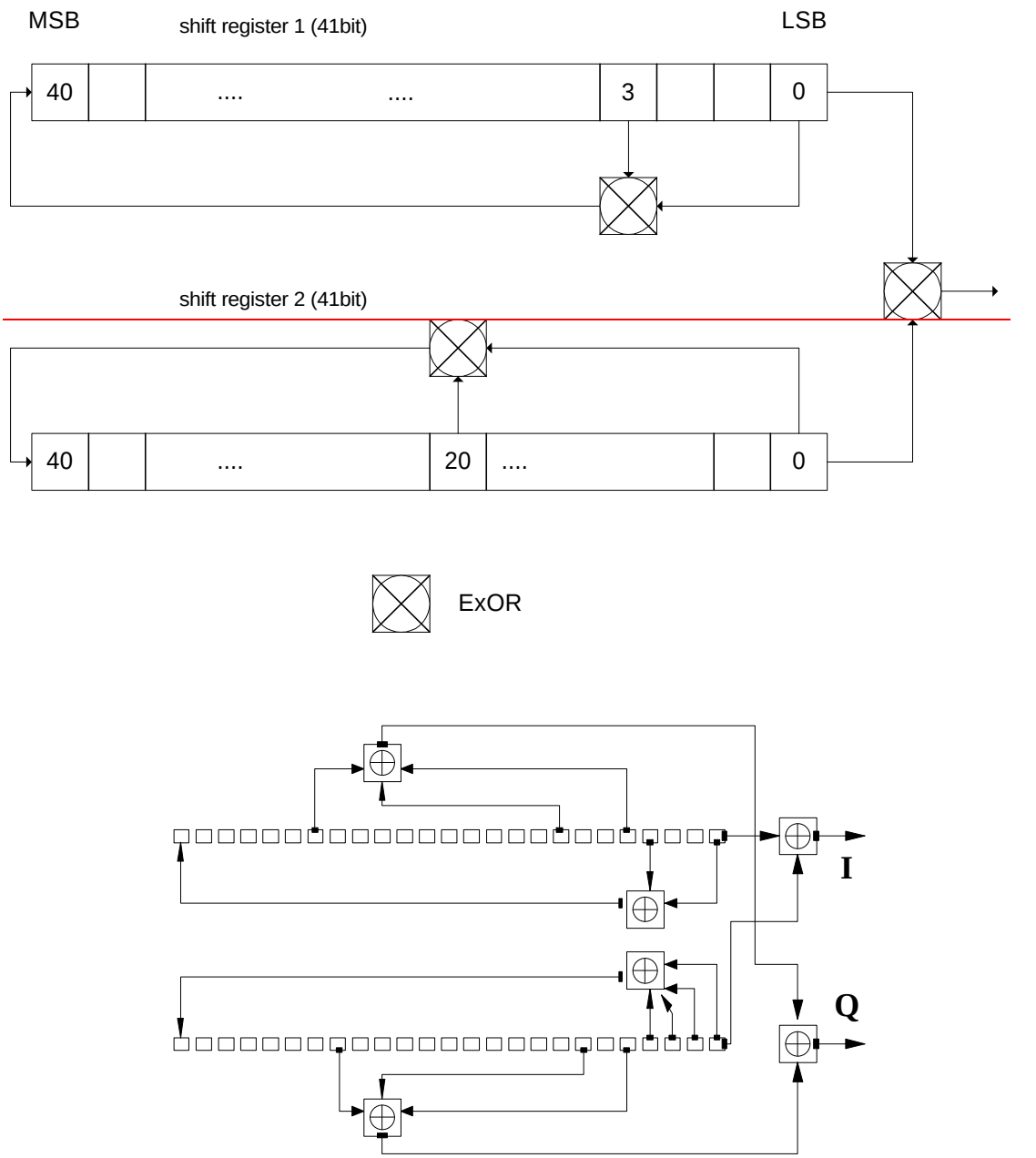

Figure 1. Configuration of uplink scrambling code generator (only an example).

Chip rate (Mcps)	Period N (chips)	I/Q Offset M (chips)	Range of phase (chip)	
			(c_1)	(c_2')
[1.024	10240	~~896~~16777232	0 – N-1	M –N+(M-1)
4.096	40960	~~3584~~16777232		
[8.192	81920	~~7168~~16777232		

[16.384	163840	~~1433361~~1677 7232			

Table 1. Correspondence between chip rate and uplink scrambling code phase range