

# Mueller Exhibit 14

**View:** [Next in topic](#) | [Previous in topic](#)  
[Next by same author](#) | [Previous by same author](#)  
[Previous page \(August 1999\)](#) | [Back to main 3GPP\\_TSG\\_RAN\\_WG1 page](#)  
[Join or leave 3GPP\\_TSG\\_RAN\\_WG1](#)  
[Reply](#) | [Post a new message](#)  
[Search](#)  
[Log in](#)

**Options:** [Chronologically](#) | [Most recent first](#)  
[Proportional font](#) | [Non-proportional font](#)

**Date:** Thu, 26 Aug 1999 01:08:58 KST  
**Reply-To:** [\[log in to unmask\]](#)  
**Sender:** "3GPP\_TSG\_RAN\_WG1: TSG RAN Working Group 1"  
<[\[log in to unmask\]](#)>  
**From:** "(Changsoo PARK)" <[\[log in to unmask\]](#)>  
**Subject:** AH10, R1-99b59 multiple scrambling code  
**Comments:** cc: [\[log in to unmask\]](#)  
**Content-Type:** multipart/mixed;

Dear all,

Please find the attached "Text proposal regarding multiple scrambling codes" from SAMSUNG.

R1-99b59.zip(R1-99b59.pdf) :Text proposal regarding Multiple Scrambling Codes  
Source : SAMSUNG

Best regards,  
Jaeyoel KIM.

---

[R1-99b59.PDF](#) [application/octet-stream]

[R1-99b59.zip](#) [application/zip]

---

Back to: [Top of message](#) | [Previous page](#) | [Main 3GPP\\_TSG\\_RAN\\_WG1 page](#)

[LIST.ETSI.ORG](#)





Hanover, Germany

August 30 - September 3, 1999

**Agenda item:**

**Source:** Samsung

**Title:** .....

**Document for:** Decision

**Abstract**

Samsung proposed about the multiple scrambling code generation in [1]. This text proposal describe the text change of the multiple scrambling code section to the contribution.

**Proposed Text**

**5.2.2 Scrambling code**

There are a total  $512 \times 512 = 262,144$  scrambling codes, numbered 0..262,143. The scrambling codes are divided into 512 sets each of a primary scrambling code and 511 secondary scrambling codes.

The primary scrambling codes consist of scrambling codes  $i = 0..511$ . The  $i$ :th set of secondary scrambling codes consists of scrambling codes  $i+k \times 512$ , where  $k=1..511$ .

There is a one-to-one mapping between each primary scrambling code and 511 secondary scrambling codes in a set such that  $i$ :th primary scrambling code corresponds to  $i$ :th set of scrambling codes.

The set of primary scrambling codes is further divided into 32 scrambling code groups, each consisting of 16 primary scrambling codes. The  $j$ :th scrambling code group consists of scrambling codes  $j+16k$ , where  $j=0..31$  and  $k=0..15$ . Each cell is allocated one and only one primary scrambling code. The primary CCPCH is always transmitted using the primary scrambling code. The other downlink physical channels can be

transmitted with either the primary scrambling code or a secondary scrambling code from the set associated with the primary scrambling code of the cell.

~~<Editor's note: There may be a need to limit the actual number of codes used in each set of secondary scrambling codes, in order to limit the signalling requirements.>~~

<Editor's note: it is not standardised how many scrambling codes a UE must decode in parallel.>

The scrambling code sequences are constructed by combining two real sequences into a complex sequence. Each of the two real sequences 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 18. The resulting sequences thus constitute segments of a set of Gold sequences. The scrambling codes are repeated for every 10 ms radio frame. Let  $x$  and  $y$  be the two sequences respectively. The  $x$  sequence is constructed using the primitive (over GF(2)) polynomial  $1+X^7+X^{18}$ . The  $y$  sequence is constructed using the polynomial  $1+X^5+X^7+X^{10}+X^{18}$ .

<Editor's note: [ ] is due to the fact that only 4.096Mcps is a working assumptions. 1.024, 8.192, and 16.384Mcps are ffs.>

~~Let  $n_1, \dots, n_0$  be the binary representation of the scrambling code number  $n$  (decimal) 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:

$$x_n(0)=n_0, x_n(1)=n_1, \dots, x_n(16)=n_{16}, x_n(17)=n_{17}$$

$$y(0)=y(1)=\dots=y(16)=y(17)=1$$

Recursive definition of subsequent symbols:

$$x_n(i+18) = x_n(i+7) + x_n(i) \text{ modulo } 2, i=0, \dots, 2^{18}-20,$$

$$y(i+18) = y(i+10)+y(i+7)+y(i+5)+y(i) \text{ modulo } 2, i=0, \dots, 2^{18}-20.$$

The  $n$ .th Gold code sequence  $z_n$  is then defined as

$$z_n(i) = x_n(i) + y(i) \text{ modulo } 2, i=0, \dots, 2^{18}-2.$$

~~$x_0$  is constructed with  $x_0(0) = x_0(1) = \dots x_0(16) = 0, x_0(17) = 1$  as initial conditions..~~

~~$x_n$  is constructed with  $n$  phase shift from  $x_0$ ..~~

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

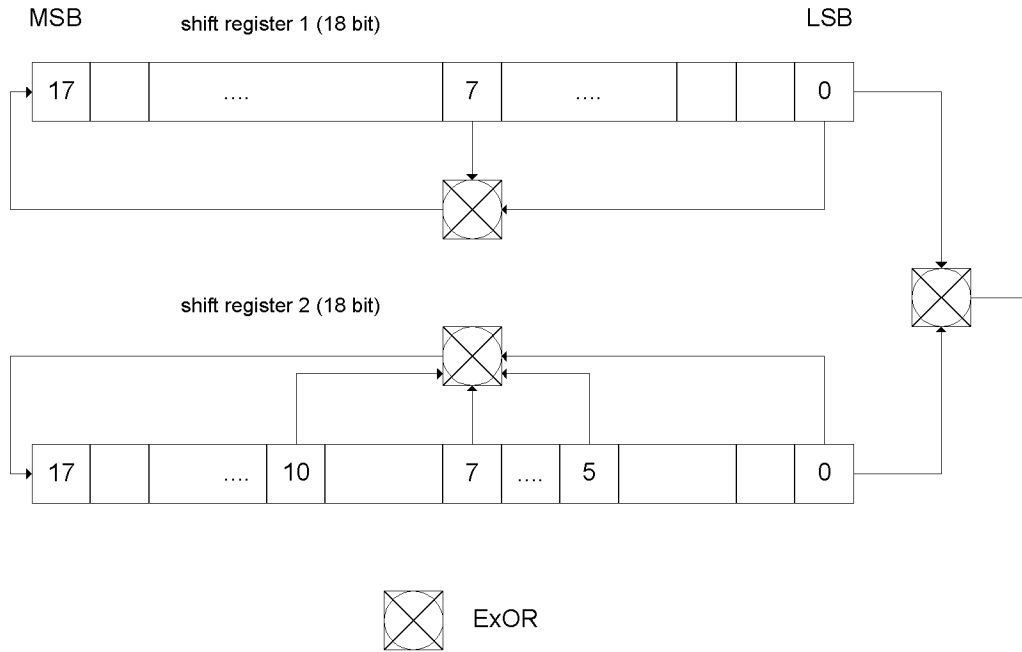
Finally, the  $n$ .th complex scrambling code sequence  $C_{scramb}$  is defined as (the lowest index corresponding to the chip scrambled first in each radio frame): (see Table 1 for definition of  $N$

and M)

$$C_{scramb}(i) = z'_n(i) + j z'_n(i+M), i=0,1,\dots,N-1.$$

<Editor's note: the values 40960 is based on an assumption of a chip rate of 4.096 Mcps.>

Note that the pattern from phase 0 up to the phase of 10 msec is repeated.



**Figure 1. Configuration of downlink scrambling code generator**

<Editor's note: a replacement figure for the above is to be prepared showing both I & Q generation.>

chip rate (Mcps)	Period N	IQ Offset M	Range of phase (chip)	
			for in-phase component	for quadrature component
[1.024]	[10240]	[131072]	0 – N-1	M – N+M-1
4.096	40960	131072		
[8.192]	[81920]	[131072]		
[16.384]	[163840]	[131072]		

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

## Reference

- [1] 3GPP TSGR1#6 (99)924, 'Multiple scrambling code', Source: Samsung
- [2] 3GPP TSGR1#7 (99)a86, 'TS 25.213 V2.0.1 (1999-08) Spreading and modulation(FDD)'