TSG-RAN Working Group 1 meeting #9
Dresden, Germany
November 30 – December 3, 1999

TSGR1#9(99)L12

Agenda item:

Source: Ericsson

Title: CR 25.213-019: Correction to code allocation for compressed mode

Document for: Decision

This CR requests a correction to the description of downlink spreading in Section 5.2 of 25.213.

It has been noted that the current rule for code allocation in case of compresed mode using alternative scrambling codes may lead to collisions in some cases. This CR proposes a somewhat different rule that avoids this.

In the current allocation scheme, channelization codes in a left-branch are mapped to a "left" code tree and codes in a right-branch are mapped to a "right" code tree. The channelization-code allocation is the same as for the case of ordinary scrambling code for compressed frames. Figure 1 illustrates a possible collision in case of the current scheme.

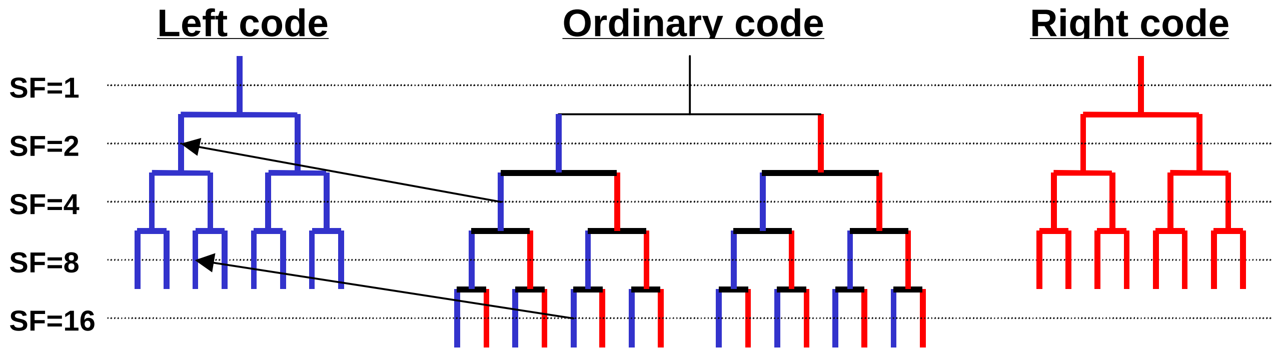

Figure 1

Figure 2 illustrates the proposed mapping in case of alternative scrambling codes. Channelization codes in the *left half-part* of the ordinary code tree are mapped to a "left"-code tree and codes in the *right half-part* of the ordinary code tree are mapped to a "right"-code tree. The mapping is such that the position in the alternative code tree is identical to the position in the ordinary code tree for non-compressed frames, except that the spreading factor is reduced by 50%. With this allocation, all codes that are mapped to the same alternative scrambling code keep their relative position. Collisions can thus not occur.

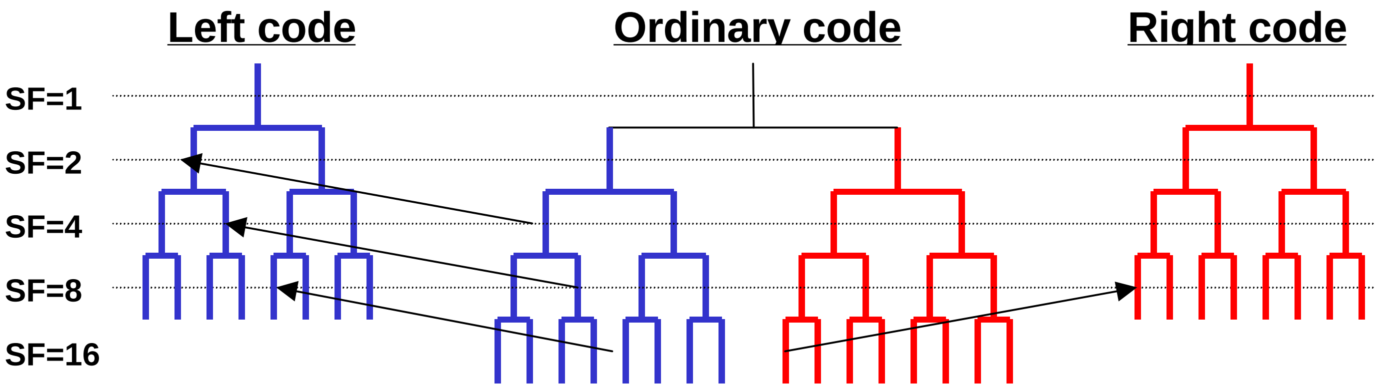

Figure 2

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

25.213	**CR**	**019**	Current Version: 3.0.0

GSM (AA.BB) or 3G (AA.BBB) specification number ↑ *↑ CR number as allocated by MCC support team*

For submission to:	TSG-RAN #6	for approval	X	strategic	
		for information		non-strategic	

list expected approval meeting # here ↑ *(for SMG use only)*

Proposed change affects: (U)SIM [ ] ME [X] UTRAN / Radio [X] Core Network [ ]
(at least one should be marked with an X)

Source: Ericsson **Date:** 1999-12-03

Subject: Correction to code allocation for compressed mode

Work item:

Category: *(only one category shall be marked with an X)*

F	Correction	X
A	Corresponds to a correction in an earlier release	
B	Addition of feature	
C	Functional modification of feature	
D	Editorial modification	

Release:

Phase 2	
Release 96	
Release 97	
Release 98	
Release 99	X
Release 00	

Reason for change: The current code allocation in case of alternative scrambling code in compressed mode is incorrect and may lead to collision in some rare case. This change request corrects this incorrectness.

Clauses affected: 5.2

Other specs affected:

Other 3G core specifications		→ List of CRs:
Other GSM core specifications		→ List of CRs:
MS test specifications		→ List of CRs:
BSS test specifications		→ List of CRs:
O&M specifications		→ List of CRs:

Other comments:

help.doc

<--------- double-click here for help and instructions on how to create a CR.

5.2 Code generation and allocation

5.2.1 Channelization codes

The channelization codes of figures 8 and 9 are the same codes as used in the uplink, namely Orthogonal Variable Spreading Factor (OVSF) codes that preserve the orthogonality between downlink channels of different rates and spreading factors. The OVSF codes are defined in figure 4 in section 4.3.1.

The channelization code for the Primary CPICH is fixed to $C_{ch,256,0}$ and the channelization code for the Primary CCPCH is fixed to $C_{ch,256,1}$.The channelization codes for all other physical channels are assigned by UTRAN.

When compressed mode is implemented by reducing the spreading factor by 2, the OVSF code used for compressed frames is:

- $C_{ch,SF/2,\lfloor n/2 \rfloor}$ if ordinary scrambling code is used
- $c_{ch,SF/2,n \bmod SF/2}$ if alternative scrambling code is used (see section 5.2.2)

where $c_{ch,SF,n}$ is the channelization code used for non-compressed frames.

~~of spreading factor SF/2 on the path to the root of the code tree from the OVSF code assigned for normal frames is used in the compressed frames. For the case where the scrambling code is changed during compressed frames, an even numbered OVSF code used in normal mode results in using the even alternative scrambling code during compressed frames, while an odd numbered OVSF code used in normal mode results in using the odd alternative scrambling code during compressed frames. The even and odd alternative scrambling codes are described in the next section.~~

In case the OVSF code on the PDSCH varies from frame to frame, the OVSF codes shall be allocated such a way that the OVSF code(s) below the smallest spreading factor will be from the branch of the code tree pointed by the smallest spreading factor used for the connection. This means that all the codes for UE for the PDSCH connection can be generated according to the OVSF code generation principle from smallest spreading factor code used by the UE on PDSCH.

In case of mapping the DSCH to multiple parallel PDSCHs, the same rule applies, but all of the branches identified by the multiple codes, corresponding to the smallest spreading factor, may be used for higher spreading factor allocation.

5.2.2 Scrambling code

A total of 2^{18}-1 = 262,143 scrambling codes, numbered 0…262,142 can be generated. However not all the scrambling codes are used. The scrambling codes are divided into 512 sets each of a primary scrambling code and 15 secondary scrambling codes.

The primary scrambling codes consist of scrambling codes n=16*i where i=0…511. The i:th set of secondary scrambling codes consists of scrambling codes 16*i+k, where k=1…15.

There is a one-to-one mapping between each primary scrambling code and 15 secondary scrambling codes in a set such that i:th primary scrambling code corresponds to i:th set of scrambling codes.

Hence, according to the above, scrambling codes k = 0, 1, …, 8191 are used. Each of these codes are associated with a~~n~~ ~~even~~ left alternative scrambling code and a~~n odd~~ right alternative scrambling code, that may be used for compressed frames. The ~~even~~left alternative scrambling code corresponding to scrambling code k is scrambling code number k + 8192, while the ~~odd~~right alternative scrambling code corresponding to scrambling code k is scrambling code number k + 16384. The alternative scrambling codes can be used for compressed frames. In this case, the left alternative scrambling code is used if n<SF/2 and the right alternative scrambling code is used if n≥SF/2, where $c_{ch,SF,n}$ is the channelization code used for non-compressed frames. The usage of alternative scrambling code for compressed frames is signalled by higher layers for each physical channel respectively.

The set of primary scrambling codes is further divided into 64 scrambling code groups, each consisting of 8 primary scrambling codes. The j:th scrambling code group consists of primary scrambling codes 16*8*j+16*k, where j=0..63 and k=0..7.

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.

The mixture of primary scrambling code and secondary scrambling code for one CCTrCH is allowable.

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 38400 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 $1+X^5+X^7+ X^{10}+X^{18}$.

The sequence depending on the chosen scrambling code number n is denoted z_n, in the sequel. Furthermore, let $x(i)$, $y(i)$ and $z_n(i)$ denote the *i:*th symbol of the sequence x, y, and z_n, respectively

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

Initial conditions:

x is constructed with x (0)=1, x(1)= x(2)=...= x (16)= x (17)=0

y(0)=y(1)= … =y(16)= y(17)=1

Recursive definition of subsequent symbols:

$x(i+18) =x(i+7) + x(i)$ modulo 2, $i=0,\ldots,2^{18}-20$,

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

The n:th Gold code sequence z_n, $n=0,1,2,\ldots,2^{18}-2$, is then defined as

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

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 $S_{dl,n}$ is defined as (the lowest index corresponding to the chip scrambled first in each radio frame)(where N is the period in chips and M is 131,072):

$S_{dl,n}(i) = z_n(i) + j\, z_n(i+M)$, $i=0,1,\ldots,N-1$.

Note that the pattern from phase 0 up to the phase of 38399 is repeated.

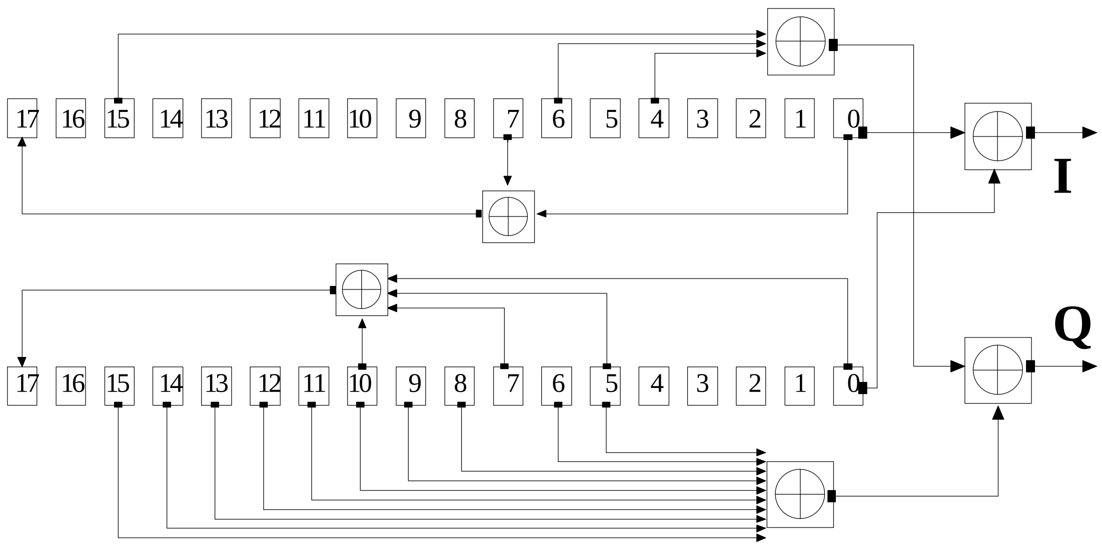

Figure 11: Configuration of downlink scrambling code generator

5.2.3 Synchronisation codes

5.2.3.1 Code Generation

The primary code sequence, C_{psc} 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, …, x_{16}> = <0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0> and

$$b = < x_1, x_2, .., x_8, \bar{x}_9, \bar{x}_{10}, .., \bar{x}_{16} >$$

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_{psc} follows (the left most index corresponds to the chip transmitted first in each time slot):

C_{psc} = <y(0),y(1),y(2),...,y(255)>.

Let the sequence $Z = \{b, b, b, \bar{b}, b, b, \bar{b}, \bar{b}, b, \bar{b}, b, \bar{b}, \bar{b}, \bar{b}, \bar{b}, \bar{b}\}$. Then the Secondary Synchronization code words, $\{C_{ssc,1}, \ldots, C_{ssc,16}\}$ 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 16[th] row of the matrix H_8 implying 16 possible code words given by n =0,16,32,48,64,80,96,112,128,144,160,176,192,208,224,240.

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

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 SCH code words are defined in terms of $C_{sch,n}$:

$C_{ssc,i} = C_{sch,i}$, i=1,…,16

5.2.3.2 Code Allocation

The 64 sequences are constructed such that their cyclic-shifts are unique, i.e., a non-zero cyclic shift less than 15 of any of the 64 sequences is not equivalent to some cyclic shift of any other of the 64 sequences. Also, a non-zero cyclic shift less than 15 of any of the sequences is not equivalent to itself with any other cyclic shift less than 15. The following sequences are used to encode the 64 different scrambling code groups (note that c_i indicates the i'th secondary code of the 16 codes). Note that a secondary code can be different from one time slot to another and that the sequence pattern can be different from one cell to another, depending on Scrambling Code Group the cell uses.

Table 5: Spreading Code allocation for Secondary SCH Code, the index "i" of the code Ci

Scrambling Code Group	slot number														
	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14
Group 1	1	1	2	8	9	10	15	8	10	16	2	7	15	7	16
Group 2	1	1	5	16	7	3	14	16	3	10	5	12	14	12	10
Group 3	1	2	1	15	5	5	12	16	6	11	2	16	11	15	12
Group 4	1	2	3	1	8	6	5	2	5	8	4	4	6	3	7
Group 5	1	2	16	6	6	11	15	5	12	1	15	12	16	11	2
Group 6	1	3	4	7	4	1	5	5	3	6	2	8	7	6	8
Group 7	1	4	11	3	4	10	9	2	11	2	10	12	12	9	3
Group 8	1	5	6	6	14	9	10	2	13	9	2	5	14	1	13
Group 9	1	6	10	10	4	11	7	13	16	11	13	6	4	1	16
Group 10	1	6	13	2	14	2	6	5	5	13	10	9	1	14	10
Group 11	1	7	8	5	7	2	4	3	8	3	2	6	6	4	5
Group 12	1	7	10	9	16	7	9	15	1	8	16	8	15	2	2
Group 13	1	8	12	9	9	4	13	16	5	1	13	5	12	4	8
Group 14	1	8	14	10	14	1	15	15	8	5	11	4	10	5	4
Group 15	1	9	2	15	15	16	10	7	8	1	10	8	2	16	9
Group 16	1	9	15	6	16	2	13	14	10	11	7	4	5	12	3
Group 17	1	10	9	11	15	7	6	4	16	5	2	12	13	3	14
Group 18	1	11	14	4	13	2	9	10	12	16	8	5	3	15	6
Group 19	1	12	12	13	14	7	2	8	14	2	1	13	11	8	11
Group 20	1	12	15	5	4	14	3	16	7	8	6	2	10	11	13
Group 21	1	15	4	3	7	6	10	13	12	5	14	16	8	2	11
Group 22	1	16	3	12	11	9	13	5	8	2	14	7	4	10	15
Group 23	2	2	5	10	16	11	3	10	11	8	5	13	3	13	8
Group 24	2	2	12	3	15	5	8	3	5	14	12	9	8	9	14
Group 25	2	3	6	16	12	16	3	13	13	6	7	9	2	12	7
Group 26	2	3	8	2	9	15	14	3	14	9	5	5	15	8	12
Group 27	2	4	7	9	5	4	9	11	2	14	5	14	11	16	16
Group 28	2	4	13	12	12	7	15	10	5	2	15	5	13	7	4
Group 29	2	5	9	9	3	12	8	14	15	12	14	5	3	2	15
Group 30	2	5	11	7	2	11	9	4	16	7	16	9	14	14	4
Group 31	2	6	2	13	3	3	12	9	7	16	6	9	16	13	12
Group 32	2	6	9	7	7	16	13	3	12	2	13	12	9	16	6
Group 33	2	7	12	15	2	12	4	10	13	15	13	4	5	5	10
Group 34	2	7	14	16	5	9	2	9	16	11	11	5	7	4	14
Group 35	2	8	5	12	5	2	14	14	8	15	3	9	12	15	9
Group 36	2	9	13	4	2	13	8	11	6	4	6	8	15	15	11
Group 37	2	10	3	2	13	16	8	10	8	13	11	11	16	3	5
Group 38	2	11	15	3	11	6	14	10	15	10	6	7	7	14	3
Group 39	2	16	4	5	16	14	7	11	4	11	14	9	9	7	5
Group 40	3	3	4	6	11	12	13	6	12	14	4	5	13	5	14
Group 41	3	3	6	5	16	9	15	5	9	10	6	4	15	4	10
Group 42	3	4	5	14	4	6	12	13	5	13	6	11	11	12	14
Group 43	3	4	9	16	10	4	16	15	3	5	10	5	15	6	6
Group 44	3	4	16	10	5	10	4	9	9	16	15	6	3	5	15
Group 45	3	5	12	11	14	5	11	13	3	6	14	6	13	4	4
Group 46	3	6	4	10	6	5	9	15	4	15	5	16	16	9	10
Group 47	3	7	8	8	16	11	12	4	15	11	4	7	16	3	15
Group 48	3	7	16	11	4	15	3	15	11	12	12	4	7	8	16
Group 49	3	8	7	15	4	8	15	12	3	16	4	16	12	11	11
Group 50	3	8	15	4	16	4	8	7	7	15	12	11	3	16	12

Group 51	3	10	10	15	16	5	4	6	16	4	3	15	9	6	9
Group 52	3	13	11	5	4	12	4	11	6	6	5	3	14	13	12
Group 53	3	14	7	9	14	10	13	8	7	8	10	4	4	13	9
Group 54	5	5	8	14	16	13	6	14	13	7	8	15	6	15	7
Group 55	5	6	11	7	10	8	5	8	7	12	12	10	6	9	11
Group 56	5	6	13	8	13	5	7	7	6	16	14	15	8	16	15
Group 57	5	7	9	10	7	11	6	12	9	12	11	8	8	6	10
Group 58	5	9	6	8	10	9	8	12	5	11	10	11	12	7	7
Group 59	5	10	10	12	8	11	9	7	8	9	5	12	6	7	6
Group 60	5	10	12	6	5	12	8	9	7	6	7	8	11	11	9
Group 61	5	13	15	15	14	8	6	7	16	8	7	13	14	5	16
Group 62	9	10	13	10	11	15	15	9	16	12	14	13	16	14	11
Group 63	9	11	12	15	12	9	13	13	11	14	10	16	15	14	16
Group 64	9	12	10	15	13	14	9	14	15	11	11	13	12	16	10

5.3 Modulation

5.3.1 Modulating chip rate

The mQAodulating chip rate is 3.84 Mcps.

5.3.2 Modulation

QPSK modulation is used.

Annex A (informative): Generalised Hierarchical Golay Sequences

A.1 Alternative generation

The generalised hierarchical Golay sequences for the PSC described in 5.2.3.1 may be also viewed as generated (in real valued representation) by the following methods:

Method 1.

The sequence y is constructed from two constituent sequences x_1 and x_2 of length n_1 and n_2 respectively using the following formula:

$$y(i) = x_2(i \bmod n_2) * x_1(i \operatorname{div} n_2),\ i = 0 \dots (n_1 * n_2) - 1$$

The constituent sequences x_1 and x_2 are chosen to be the following length 16 (i.e. $n_1 = n_2 = 16$) sequences:

- x_1 is defined to be the length 16 ($N^{(1)}=4$) Golay complementary sequence obtained by the delay matrix $D^{(1)} = [8, 4, 1,2]$ and weight matrix $W^{(1)} = [1, -1, 1,1]$.

- x_2 is a generalised hierarchical sequence using the following formula, selecting s=2 and using the two Golay complementary sequences x_3 and x_4 as constituent sequences. The length of the sequence x_3 and x_4 is called n_3 respectively n_4.

  $$x_2(i) = x_4(i \bmod s + s*(i \operatorname{div} sn_3)) * x_3((i \operatorname{div} s) \bmod n_3),\ i = 0 \dots (n_3 * n_4) - 1$$

  x_3 and x_4 are defined to be identical and the length 4 ($N^{(3)}= N^{(4)}=2$) Golay complementary sequence obtained by the delay matrix $D^{(3)} = D^{(4)} = [1, 2]$ and weight matrix $W^{(3)} = W^{(4)} = [1, 1]$.

The Golay complementary sequences x_1,x_3 and x_4 are defined using the following recursive relation:

$a_0(k) = \delta(k)$ and $b_0(k) = \delta(k)$

$a_n(k) = a_{n-1}(k) + W^{(j)}_n \cdot b_{n-1}(k-D^{(j)}_n)$,

$b_n(k) = a_{n-1}(k) - W^{(j)}_n \cdot b_{n-1}(k-D^{(j)}_n)$,

$k = 0, 1, 2, \ldots, 2^{**}N^{(j)} -1$,

$n = 1, 2, \ldots, N^{(j)}$.

The wanted Golay complementary sequence x_j is defined by a_n assuming $n=N^{(j)}$. The Kronecker delta function is described by δ, k,j and n are integers.

<u>Method 2</u>

The sequence y can be viewed as a pruned Golay complementary sequence and generated using the following parameters which apply to the generator equations for a and b above:

(a) Let j = 0, $N^{(0)} = 8$

(b) $[D_1^0,D_2^0,D_3^0,D_4^0,D_5^0,D_6^0,D_7^0,D_8^0] = [128, 64, 16, 32, 8, 1, 4, 2]$

(c) $[W_1^0,W_2^0,W_3^0,W_4^0,W_5^0,W_6^0,W_7^0,W_8^0] = [1, -1, 1, 1, 1, 1, 1, 1]$

(d) For n = 4, 6, set $b_4(k) = a_4(k)$, $b_6(k) = a_6(k)$.