0 TSG-RAN Working Group 1 meeting #3 **TSGR1#3(99)205**
1 Nynäshamn, Sweden
March 22-26, 1999

Agenda Item:

Source: Ericsson

Title: New RACH preambles with low auto-correlation sidelobes and reduced detector complexity

Document for:

---

1 Introduction

The preamble part of the random access burst signal format proposed for UTRA/FDD has the length of 4096 code chips [1]. The preamble consists of a *signature* of length 16 complex symbols, which are spread by a common, 256 chip long Orthogonal Gold sequence called *preamble spreading code*. In total there are 16 different signatures, obtained from the orthogonal set of binary Orthogonal Gold sequences of length 16, by multiplying each binary code with the constant complex number $C=(1+j)/\sqrt{2}$, where $j=\sqrt{-1}$.

The UE transmissions of the random access bursts can start at a number of well-defined time offsets (access slots), which are synchronised to the frame sync of the primary CCPCH. The primary CCPCH frame sync is extracted after the cell search procedure in the UE. Therefore the random access preambles are received at the base station at the beginning of each access slot interval with the time uncertainty equal to the round-trip propagation delay.

The current random access preamble construction allows simplified realisation of the bank of correlators required in the base station random access receiver if this time uncertainty is smaller than 255 chips. However, the aperiodic auto-correlation sidelobes of such codes are rather high, which means that the RACH preamble might be detected at wrong time positions. In other words, the preamble detection probability at correct time positions is deteriorated for moderate to high signal-to-noise ratios. Therefore it is desirable to find another random access preamble construction, which would also produce an *orthogonal set* of preamble codes with much lower *aperiodic auto-correlation* sidelobes, facilitating an efficient *matched filter implementation*.

2 Golay complementary sequences

The new RACH preambles are based on the application of binary sequences from the Golay complementary pairs. The major property of the binary sequences from the Golay complementary pair is that the sum of their aperiodic auto-correlation functions equals zero for all non-zero time shifts. The Golay sequences can be constructed for any length $L=2^N$, where N is any positive integer, and also for lengths 10 and 26, or for any combination of those three lengths. Besides the complementary property, such sequences exhibit some additional properties which make them attractive as synchronisation codes: they have low aperiodic auto-correlation sidelobes, and there is a large number of them for a given code length.

If the sequences are of length $L=2^N$, there is a general method for the construction of polyphase complementary pairs of sequences, where the Golay complementary sequences are just a special, binary case. That general construction is defined by the following recursive relation [2].

$$a_0(k) = \delta(k)$$
$$b_0(k) = \delta(k)$$
$$a_n(k) = a_{n-1}(k) + W_n \cdot b_{n-1}(k-D_n)$$

$b_n(k) = a_{n-1}(k) - W_n \cdot b_{n-1}(k-D_n)$, (1)

$k = 0, 1, 2, \ldots, 2^N-1,$
$n = 1, 2, \ldots, N,$
$D_n = 2^{P_n}$,

where

$a_n(k)$ and $b_n(k)$ are two complementary sequences of length 2^N,
$\delta(k)$ is the Kronecker delta function,
k is an integer representing the time scale,
n is the iteration number,
D_n is a delay,
P_n , $n = 1, 2, \ldots, N$, is any permutation of numbers $\{0, 1, 2, \ldots, N-1\}$,
W_n is an arbitrary complex number of unit magnitude.

If W_n has values +1 and –1, the binary (Golay) complementary sequences are obtained [3].

An efficient matched filter directly corresponding to the complementary sequences $a_N(k)$ and $b_N(k)$ defined by (1) is given in Figure 1. This filter performs the correlation of input signal $r(k)$ *simultaneousl*y with the two complementary sequences $a_N(k)$ and $b_N(k)$. The two matched filter outputs produce the two corresponding aperiodic cross-correlation functions $R_{ra}(\tau)$ and $R_{rb}(\tau)$. Such a digital filter will be called the Efficient Golay Correlator (EGC), although it is actually the filter matched also to any polyphase complementary pair defined by (1). The matched filter has complex conjugated coefficients W_n, denoted as W_n^*.

Figure 1: Efficient Golay Correlator (EGC).

The boxes in Figure 1 represent the corresponding delay lines with D_n memory elements. The number of *multiplications* in the EGC is equal to $log_2(L)$, while in the straightforward matched filter implementation it would be L. The number of *additions* in the EGC is $2 \cdot log_2(L)$, while in the straightforward matched filter implementation it would be L-1. The number of memory elements required for the EGC is L-1 ($=D_1+D_2+\ldots+D_N$), the same as for the straightforward implementation of a single matched filter corresponding to one of the complementary sequences.

3 Efficient Golay correlator with reduced memory

In the case when the expected delays τ of input signal are limited to be $|\tau|<T_{max}$ chips, it is possible to derive another Efficient Golay correlator with reduced memory. The EGC with reduced memory is based on the representation of a Golay sequence of length $L=2^N=J \cdot T_{max}$ in the so-called "factored" form, i.e. as a function of two shorter constituent complementary sequences $A(k)$ and $B(k)$ of length T_{max}. This relation is a simple consequence of the general recursive construction (1), which can actually start from any complementary pair of sequences. Namely, if the initial vectors $a_0(k)$ and $b_0(k)$ are taken to be

$a_0(k) = A(k),$
$b_0(k) = B(k), \quad k=0, 1, 2, \ldots, T_{max}-1,$ (2)

where $A(k)$ and $B(k)$ are the two arbitrary complementary sequences of length T_{max}, the resulting pair of complementary sequences of length $L=2^N=J \cdot T_{max}$ is generated after J iterations. Note that all the delays D_n in (1) should be multiplied by the length of constituent sequences (T_{max}).

For example, if the constituent sequences are of length T_{max}=256, the permutation vector P_n and the weighting vectors W_n are of length 16 if the resulting complementary pair should be of length 4096. The resulting Golay sequences consists of 8 times repeated sequences $A(k)$ and $B(k)$, which are multiplexed according to some equivalent binary "interleaving" function $I_0(k)$ depending on the permutation vector P_n. The orthogonal set of 16 Golay sequences of length 4096, having the *common* constituent sequences $A(k)$ and $B(k)$ of length 256 (and a common interleaving function), can be obtained by choosing a single permutation vector of length 16, along with 8 appropriately chosen weighting vectors.

The additional 16 orthogonal Golay sequences of length 4096 having the same constituent sequences $A(k)$ and $B(k)$, but interleaved according to another interleaving function $I_1(k)$, can be obtained by taking

$$a_0(k) = B(k),$$
$$b_0(k) = A(k), \quad k=0, 1, 2,\ldots, T_{max}-1, \qquad (3)$$

and using the same permutation and weighting vectors as for the first set of 16 sequences. Therefore in total there are 32 orthogonal Golay sequences of length 4096 having the common constituent sequences of length 256. In principle it is possible to generate $2J$ such orthogonal Golay sequences, where $J=L/T_{max}$.

The set of 16 orthogonal Golay sequences of length 4096 which is obtained according to the above algorithm and which can be used in UTRA/FDD is given in Table 1, as a function of two shorter, constituent complementary sequences $A(k)$ and $B(k)$ of length 256. The additional set of 16 orthogonal Golay sequences of length 4096 can be obtained by replacing $A(k)$ with $B(k)$ and $B(k)$ with $A(k)$ in Table 1. As $A(k)$ and $B(k)$ are orthogonal, the additional set is orthogonal to the first one.

Table 1: First 16 orthogonal Golay sequences of length 4096.

k	0...255… …4095															
S(0,*k*)	*A*	*A*	*B*	*B*	*A*	*-A*	*-B*	*B*	*A*	*-A*	*B*	*-B*	*A*	*A*	*-B*	*-B*
S(1,*k*)	*A*	*A*	*B*	*B*	*A*	*-A*	*-B*	*B*	*-A*	*A*	*-B*	*B*	*-A*	*-A*	*B*	*B*
S(2,*k*)	*A*	*-A*	*B*	*-B*	*A*	*A*	*-B*	*-B*	*A*	*A*	*B*	*B*	*A*	*-A*	*-B*	*B*
S(3,*k*)	*A*	*-A*	*B*	*-B*	*A*	*A*	*-B*	*-B*	*-A*	*-A*	*-B*	*-B*	*-A*	*A*	*B*	*-B*
S(4,*k*)	*A*	*A*	*B*	*B*	*-A*	*A*	*B*	*-B*	*A*	*-A*	*B*	*-B*	*-A*	*-A*	*B*	*B*
S(5,*k*)	*A*	*A*	*B*	*B*	*-A*	*A*	*B*	*-B*	*-A*	*A*	*-B*	*B*	*A*	*A*	*-B*	*-B*
S(6,*k*)	*A*	*-A*	*B*	*-B*	*-A*	*-A*	*B*	*B*	*A*	*A*	*B*	*B*	*-A*	*A*	*B*	*-B*
S(7,*k*)	*A*	*-A*	*B*	*-B*	*-A*	*-A*	*B*	*B*	*-A*	*-A*	*-B*	*-B*	*A*	*-A*	*-B*	*B*
S(8,*k*)	*A*	*A*	*-B*	*-B*	*A*	*-A*	*B*	*-B*	*A*	*-A*	*-B*	*B*	*A*	*A*	*B*	*B*
S(9,*k*)	*A*	*A*	*-B*	*-B*	*A*	*-A*	*B*	*-B*	*-A*	*A*	*B*	*-B*	*-A*	*-A*	*-B*	*-B*
S(10,*k*)	*A*	*-A*	*-B*	*B*	*A*	*A*	*B*	*B*	*A*	*A*	*-B*	*-B*	*A*	*-A*	*B*	*-B*
S(11,*k*)	*A*	*-A*	*-B*	*B*	*A*	*A*	*B*	*B*	*-A*	*-A*	*B*	*B*	*-A*	*A*	*-B*	*B*
S(12,*k*)	*A*	*A*	*-B*	*-B*	*-A*	*A*	*-B*	*B*	*A*	*-A*	*-B*	*B*	*-A*	*-A*	*-B*	*-B*
S(13,*k*)	*A*	*A*	*-B*	*-B*	*-A*	*A*	*-B*	*B*	*-A*	*A*	*B*	*-B*	*A*	*A*	*B*	*B*
S(14,*k*)	*A*	*-A*	*-B*	*B*	*-A*	*-A*	*-B*	*-B*	*A*	*A*	*-B*	*-B*	*-A*	*A*	*-B*	*B*
S(15,*k*)	*A*	*-A*	*-B*	*B*	*-A*	*-A*	*-B*	*-B*	*-A*	*-A*	*B*	*B*	*A*	*-A*	*B*	*-B*

The RACH preamble correlator with reduced memory, corresponding to the above set of 32 orthogonal preambles is shown in Figure 2. The number of memory elements per received signature in this scheme is the same as for the RACH preamble correlator described in UMTS XX.07. However, the total number of adders and multipliers is significantly reduced due to the use of EGC instead of preamble spreading code matched filter.

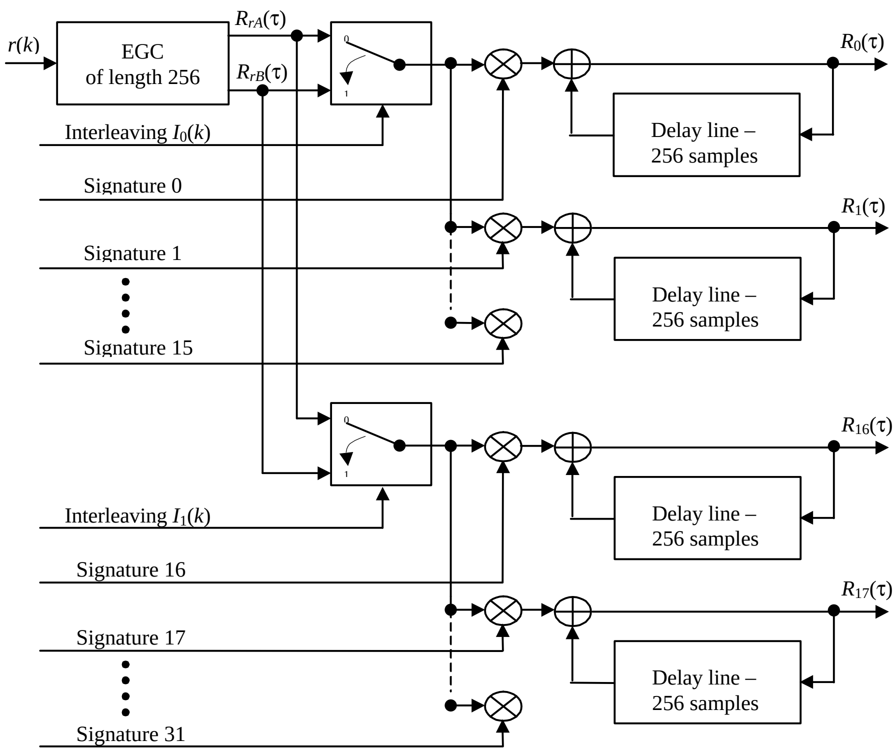

Figure 2: The bank of RACH preamble correlators with reduced memory, matched to 32 orthogonal Golay sequences of length 4096.

The interleaving function $I_0(k)$ is common for the 16 orthogonal preambles, while the interleaving function $I_1(k)$ is common for the other 16 orthogonal preambles. From Table 1 it can easily be seen that

$I_0(k)$ = {0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, }, and $I_1(k)$ = {1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0}.

Each preamble has a unique "signature" sequence, which can also be easily derived from Table 1. For example,

Signature_0 = Signature_16 = {1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, -1, -1}.

The set of 256 cell-specific pairs of constituent Golay sequences $A(k)$ and $B(k)$ (corresponding to the set of 256 cell specific preamble spreading codes) is defined by (1), where the permutation vector P_n is common for all pairs and is given by

$$P_n = \{0, 2, 1, 5, 6, 4, 7, 3\}, \qquad (4)$$

while the corresponding 256 weighting vectors $W(v,n)$, $v = 0, 1, \ldots, 255$, are defined as the 8-bit binary representations of integers {0, 1, 2, …, 255}, i.e.

$$W(v,n) = (-1)^{B_n(v)}, \quad v = 0, 1, \ldots, 255, \quad n = 1, 2, 3, \ldots, 8, \qquad (5a)$$

where $B_n(x)$ is the n-th bit in the 8-bits long binary representation of some positive integer x, i.e.

$$x = \sum_{n=1}^{8} B_n(x) \cdot 2^{n-1}. \qquad (5b)$$

Note that all 256 constituent pairs can be detected by using the same correlator shown in Figure 1, by adapting only the weighting coefficients W_n^*.

4 Implementation complexity

The implementation complexity of the bank of RACH preamble correlators is significantly reduced due to the use of EGC instead of preamble spreading code matched filter. Assuming that the number of new orthogonal preambles based on Golay complementary sequences (GCS) used is the same (16) as in the case of the current preambles based on concatenated orthogonal Gold sequences (OGS), the implementation complexity of the corresponding banks of correlators can be compared in the following way:

a) The number of adders is 32 (=16+16) for the GCS, compared to 271 (=255+16) adders for the OGS.
b) The number of multipliers is 24 (=8+16) for the GCS, compared to 272 (=256+16) multipliers for the OGS.
c) There is a multiplexer (switch) for GCS while there is no multiplexer for OGS.
d) The lengths of delay lines are the same in both cases.

5 Aperiodic autocorrelation properties

Besides the improved implementation efficiency, the new preambles based on Golay sequences offer much better performances in term of the maximum absolute aperiodic autocorrelation sidelobes (MAS), when compared with the current preamble codes based on concatenated orthogonal Gold sequences.

As a first example, the aperiodic auto-correlation function for one of the concatenated orthogonal Gold preambles generated with the old scheme for preamble spreading code n=1 is shown in Figure 3 in the annex. This should be compared with the new construction using Golay sequences with constituent sequences A and B defined by (1), (4) and, as an example, W_n={1, -1, 1, -1, 1, -1, -1, 1}. The aperiodic auto-correlation function of this new code is shown in Figure 4 in the annex. As can be seen the Golay sequences have much better auto-correlation properties.

The MAS for all the preambles based on the above preamble spreading code are listed in Table 2. The benefits of the Golay sequences in terms of reduced MAS is clear.

Table 2: MAS for preambles corresponding to one particular preamble spreading code.

Golay sequences		Concatenated Orthogonal Gold sequences	
Number of occurrences	MAS	Number of occurrences	MAS
4	**161**	1	**1024**
4	**181**	4	**1280**
8	**183**	7	**1536**
-	**-**	2	**1792**
-	-	2	**2048**

The random access preambles are not completely asynchronous to the base station receiver because the UE has the basic information about base station timing, but with an uncertainty introduced by the round-trip propagation delay between the base station and UE. The current assumption in UTRA/FDD is that the round-trip delay is at most 255 chips to be able to use the proposed simplified receiver structure, so the aperiodic auto-correlation function of random access preambles is actually of most interest only in the region +/- 255 chips around the main lobe. The maximum absolute values of aperiodic auto-correlation sidelobes in the region +/- 255 chips around the main lobe are shown in Table 3 for the previously described Golay and concatenated Orthogonal Gold sequences of length 4096.

Table 3: MAS in the +/- 255 chips region for preambles corresponding to one particular preamble spreading code.

Golay sequences		Concatenated Orthogonal Gold sequences	
Number of occurrences	MAS	Number of occurrences	MAS
16	**31**	1	**731**
-	-	2	**737**
-	-	3	**743**
-	-	3	**755**
-	-	6	**761**
-	-	1	**767**

From Table 3 it can be noticed that Golay sequences have about 25 times lower auto-correlation sidelobes than the concatenated Orthogonal Gold sequences, in the region +/- 255 chips around the main lobe.

It is clear that for the particular codes evaluate above, the Golay sequences are superior. Finally, the maximum absolute values of aperiodic auto-correlation sidelobes in the region +/- 255 chips around the main lobe are evaluated for *all* preambles. Both the Golay based 256 pairs of constituent sequences *A* and *B* defined by (4) and (5) for all 32 orthogonal preambles of length 4096 corresponding to each such pair of constituent sequences, and the current preambles based on concatenated Orthogonal Gold sequences have been investigated. The results are shown in Table 4.

Table 4: MAS in the +/- 255 chips region for all preambles.

Golay sequences		Concatenated Orthogonal Gold sequences
Number of occurrences	MAS	MAS
64	**27**	MAS values are plotted in Figure 5 in the annex. Average MAS is **669**, largest MAS is **1080**, smallest MAS is **286**. 95% of MAS values are above **500**.
128	**29**	
1280	**31**	
1024	**33**	
1600	**35**	
1280	**37**	
832	**39**	
512	**41**	
576	**43**	
256	**45**	
320	**47**	
192	**49**	
64	**51**	

Table 4 shows that all 8192 possible Golay preambles of length 4096, have extremely low maximum auto-correlation sidelobes. The average MAS is 37, and 65% of the MAS values are between 27 and 37. A simple, but rather fair, comparison between the two different preamble designs can be done by comparing the average MAS. The old concatenated orthogonal Gold preambles have an average MAS 18 times (669/37) higher than the Golay based preambles.

6 Conclusion

A new set of RACH preambles is proposed for inclusion in UTRA/FDD. The benefits of the preambles codes, based on Golay complementary sequences, are:

- The new preambles offer significantly more efficient preamble detector hardware implementation, measured in terms on the number of multipliers and adders required.
- The number of available preambles is doubled, to 8192.
- All 8192 of the new preambles have good auto-correlation properties, while the span for the old preambles is quite large and many of those codes exhibit very bad correlation properties.
- The new preambles have about 18 times lower aperiodic auto-correlation sidelobes than the present RACH preambles, offering potentially better Eb/No performance.

References

[1] S1.13, UTRA FDD Spreading and modulation.
[2] S.Z.Budisin, "Efficient pulse compressor for Golay complementary sequences", Electronics Letters, Vol.27, No.3, pp.219-220, Jan. 1991.
[3] M.J.E.Golay, "Complementary Series", IRE Trans. on Information Theory, Vol.IT-7, pp.82-87, April 1961.

Annex - Figures

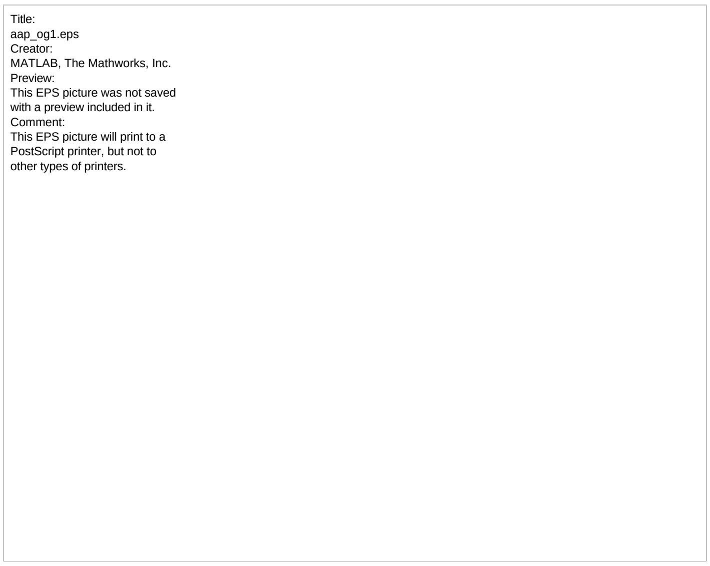

Figure 3: Aperiodic auto-correlation function for one of the present RACH preambles (signature + preamble spreading code).

Figure 4: Aperiodic auto-correlation function for one of the new RACH preambles (Golay complementary sequence from Table 1)

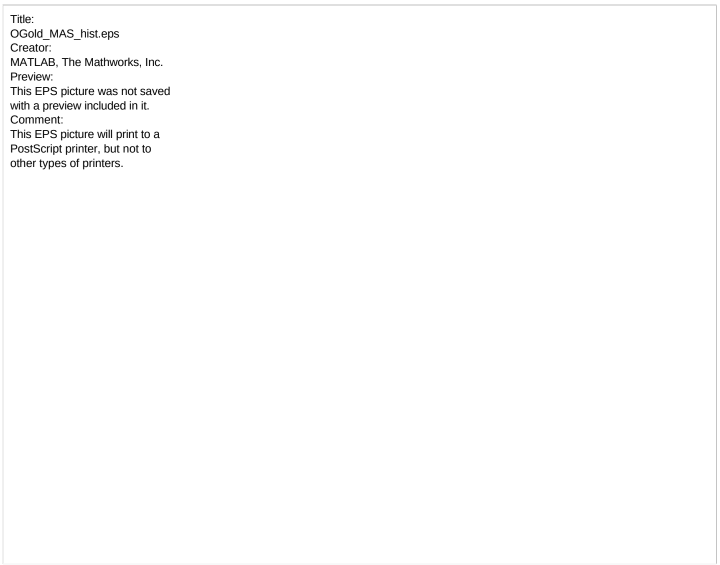

Figure 5: Distribution of MAS values for the old orthogonal Gold based preambles.