3GPP TSG RAN WG1 Meeting #9
Dresden, Germany, 30 Nov-3 Dec, 1999

Document R1-99I07
e.g. for 3GPP use the format TP-99xxx
or for SMG, use the format P-99-xxx

CHANGE REQUEST

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

25.212 CR 015 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
list expected approval meeting # here ↑

for approval ☒
for information ☐

strategic ☐
non-strategic ☐
(for SMG use only)

Form: CR cover sheet, version 2 for 3GPP and SMG The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc

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

Source: Siemens **Date:** 1999-11-22

Subject: TFCI coding

Work item:

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

F Correction ☒
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 ☒
Release 00 ☐

Reason for change:

Description of TFCI coding in split mode is currently quite different from TFCI coding in normal mode and does not specify coding/puncturing process completely. Also depends on OVSF sequences defined externally which should be removed.
- Editorial change in 4.3, BTFD, TFCI **or single TF** are possible.
- change 4.3.3 to more specification-like form.
- change 4.3.4 to more specification-like form.
- change 4.3.5.1 to show bit positions of 'b' more clearly.

Clauses affected: 4.3, 4.3.3, 4.3.4, 4.3.5.1

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:

4.3 Transport format detection

Transport format detection can be performed both with and without Transport Format Combination Indicator (TFCI). If a TFCI is transmitted, the receiver detects the transport format combination from the TFCI. When no TFCI is transmitted, so called blind transport format detection may be~~is~~ used, i.e. the receiver side detects the transport format combination using some information, e.g. received power ratio of DPDCH to DPCCH, CRC check results.

For uplink, the blind transport format detection is an operator option. For downlink, the blind transport format detection can be applied with convolutional coding, the maximum number of different transport formats and maximum data rates allowed shall be specified.

4.3.1 Blind transport format detection

Examples of blind transport format detection methods are given in Annex A.

4.3.2 Explicit transport format detection based on TFCI

The Transport Format Combination Indicator (TFCI) informs the receiver of the transport format combination of the CCTrCHs. As soon as the TFCI is detected, the transport format combination, and hence the individual transport channels' transport formats are known, and decoding of the transport channels can be performed.

4.3.3 Coding of Transport-F~~f~~ormat-C~~c~~ombination I~~i~~ndicator (TFCI)

~~The number of TFCI bits is variable and is set at the beginning of the call via higher layer signalling. For improved TFCI detection reliability, in downlink, repetition is used by increasing the number of TFCI bits within a slot.~~

The TFCI bits are encoded using a (30, 10) punctured sub-code of the second order Reed-Muller code. The coding procedure is as shown in figure 10.

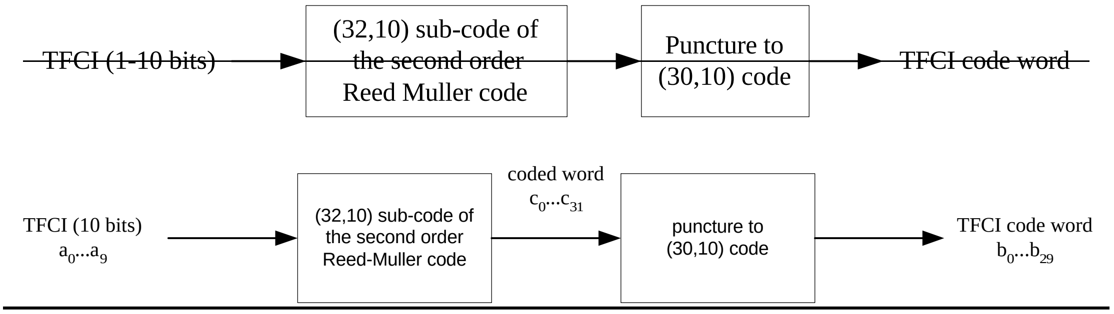

Figure 10: Channel coding of TFCI bits

If the TFCI consist of less than 10 bits, it is padded with zeros to 10 bits, by setting the most significant bits to zero. The receiver can use the information that not all 10 bits are used for the TFCI, thereby reducing the error rate in the TFCI decoder. The length of the TFCI code word is 30 bits. Thus there are 2 bits of (encoded) TFCI in every slot of the radio frame.

Firstly, TFCI is encoded by the (32,10) sub-code of second order Reed-Muller code. The code words of the (32,10) sub-code of second order Reed-Muller code are linear combination of 10 basis sequences: all 1's, 5 Hadamard ~~OVSF~~ codes (H~~C~~$_{5,1}$, H~~C~~$_{5,2}$, H~~C~~$_{5,4}$, H~~C~~$_{5,8}$, H~~C~~$_{5,16}$), and 4 masks (Mask1, Mask2, Mask3, Mask4). The ~~4 mask~~ sequences are as in the following table 7.

Table 7: Mask and Hadamard sequences

	Mask1	Mask2	Mask3	Mask4		$H_{5,1}$	$H_{5,2}$	$H_{5,4}$	$H_{5,8}$	$H_{5,16}$
bit0	0	0	0	0		0	0	0	0	0

bit1	0	0	0	0		1	0	0	0	0
bit2	1	0	0	0		0	1	0	0	0
bit3	0	0	0	1		1	1	0	0	0
bit4	1	0	1	1		0	0	1	0	0
bit5	0	0	0	1		1	0	1	0	0
bit6	0	0	1	0		0	1	1	0	0
bit7	0	1	0	0		1	1	1	0	0
bit8	0	1	1	0		0	0	0	1	0
bit9	1	1	1	0		1	0	0	1	0
bit10	1	0	1	1		0	1	0	1	0
bit11	0	0	1	1		1	1	0	1	0
bit12	0	1	1	0		0	0	1	1	0
bit13	0	1	0	1		1	0	1	1	0
bit14	1	0	0	1		0	1	1	1	0
bit15	1	1	1	1		1	1	1	1	0
bit16	1	0	0	0		0	0	0	0	1
bit17	1	1	0	0		1	0	0	0	1
bit18	1	1	0	1		0	1	0	0	1
bit19	1	0	1	0		1	1	0	0	1
bit20	0	1	1	1		0	0	1	0	1
bit21	0	1	0	1		1	0	1	0	1
bit22	0	0	1	1		0	1	1	0	1
bit23	0	1	1	1		1	1	1	0	1
bit24	0	1	0	0		0	0	0	1	1
bit25	1	1	0	1		1	0	0	1	1
bit26	1	0	1	0		0	1	0	1	1
bit27	1	0	0	1		1	1	0	1	1
bit28	0	0	1	0		0	0	1	1	1
bit29	1	1	0	0		1	0	1	1	1
bit30	1	1	1	0		0	1	1	1	1
bit31	1	1	1	1		1	1	1	1	1

~~Mask 1~~	~~00101000011000111111000001110111~~
~~Mask 2~~	~~00000001110011010110110111000111~~
~~Mask 3~~	~~00001010111110010001101100101011~~
~~Mask 4~~	~~00011100001101110010111101010001~~

For information bits a_0 , a_1 , a_2 , a_3 , a_4 , a_5 , a_6 , a_7 , a_8 , a_9 (a_0 is LSB and a_9 is MSB), the encoder structure is as following figure 11 (summation is modulo 2).

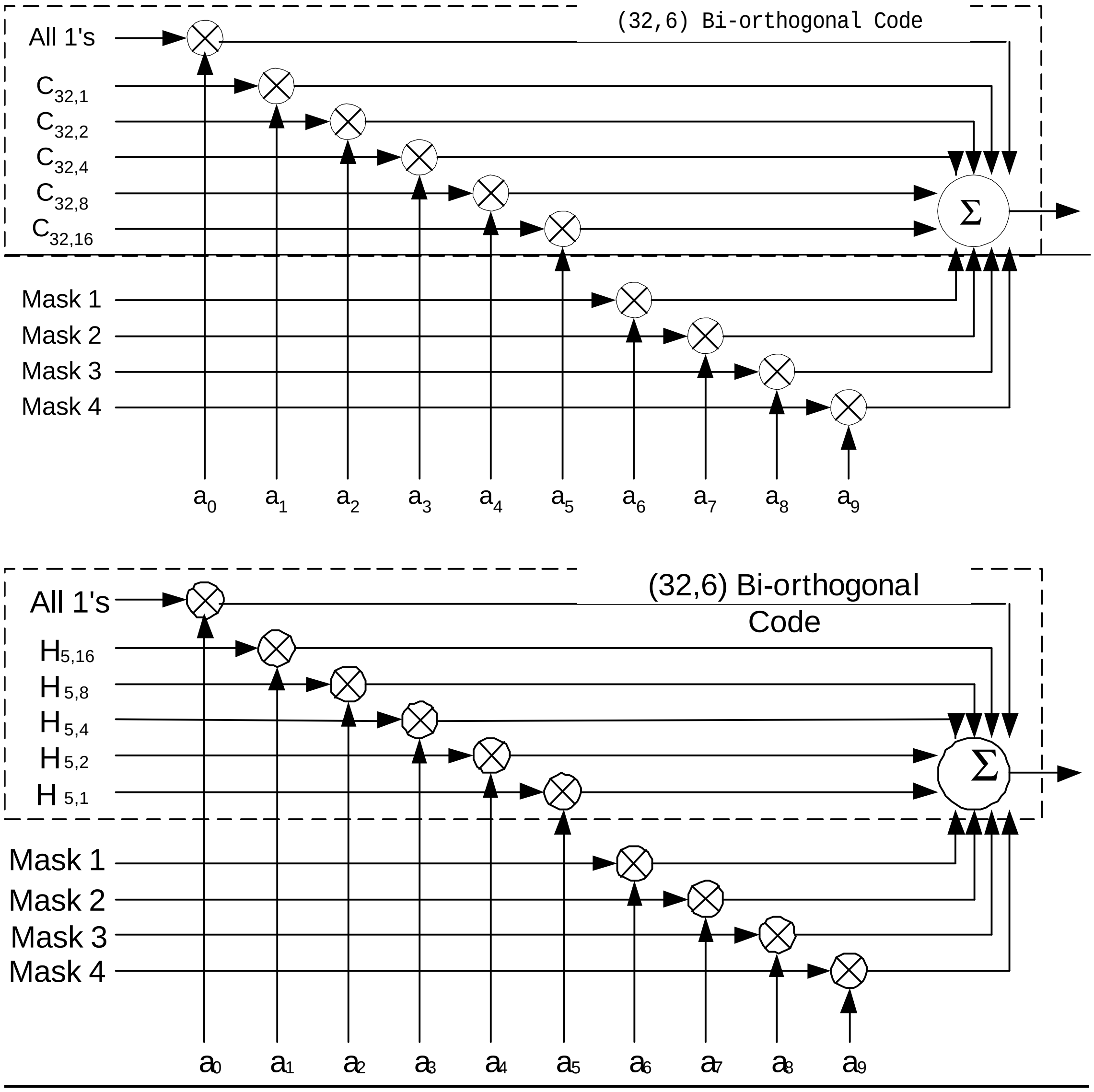

Figure 11: Encoder structure for (32,10) sub-code of second order Reed-Muller code

Then, the output~~code~~ words of the (32,10) sub-code of second order Reed-Muller coder are punctured into length 30 by puncturing output bits c_0 and c_{16}~~1st and 17th bits~~. The remaining punctured bits are denoted by b_k, k = 0, 1, 2, …, 29 (k = 29 corresponds to the MSB bit).

In downlink, when the SF is lower then 128 the encoded and punctured TFCI code words are repeated four times yielding 8 encoded TFCI bits per slot. Mapping of repeated bits to slots is explained in section 4.3.5.

4.3.4 Operation of Transport-F~~f~~ormat-C~~c~~ombination I~~i~~ndicator (TFCI) in Split Mode

In the case of DCH in Split Mode, the UTRAN shall operate with as follows:

- If one of the links is associated with a DSCH, the TFCI code word may be split in such a way that the code word relevant for TFCI activity indication is not transmitted from every cell. The use of such a functionality shall be indicated by higher layer signalling.

The TFCI bits are encoded using a (15, 5) punctured bi-orthogonal code. The coding procedure is as shown in figure 12.

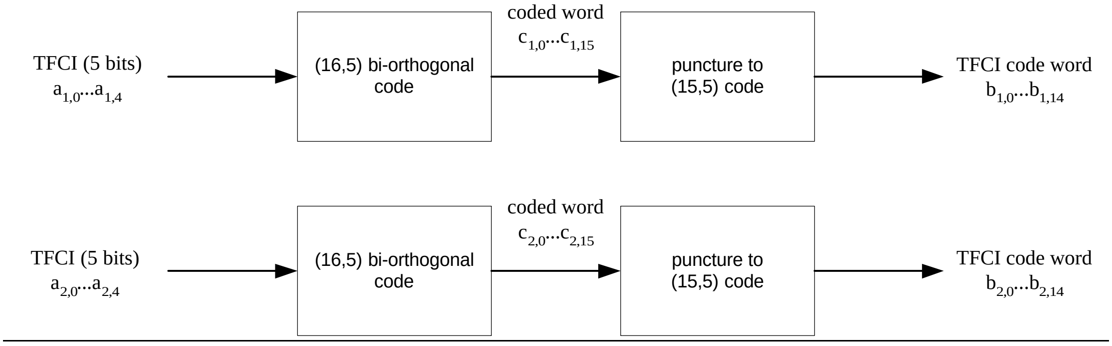

Figure 12: Channel coding of split mode TFCI bits

Firstly, TFCI is encoded by the (16,5) bi-orthogonal (or first order Reed-Muller) code. The code words of the (16,5) bi-orthogonal code are linear combinations of 5 basis sequences: the all 1's sequence and 4 Hadamard codes ($H_{4,1}$, $H_{4,2}$, $H_{4,4}$, $H_{4,8}$) as defined in table 8 below.

Table 8: Hadamard sequences

	$H_{4,1}$	$H_{4,2}$	$H_{4,4}$	$H_{4,8}$
bit0	0	0	0	0
bit1	1	0	0	0
bit2	0	1	0	0
bit3	1	1	0	0
bit4	0	0	1	0
bit5	1	0	1	0
bit6	0	1	1	0
bit7	1	1	1	0
bit8	0	0	0	1
bit9	1	0	0	1
bit10	0	1	0	1
bit11	1	1	0	1
bit12	0	0	1	1
bit13	1	0	1	1
bit14	0	1	1	1
bit15	1	1	1	1

~~TFCI information is encoded by biorthogonal (16, 5) block code. The code words of the biorthogonal (16, 5) code are from two mutually biorthogonal sets,~~ $S_{C_{16}} = \{C_{16,0}, C_{16,1}, ..., C_{16,15}\}$ ~~and its binary complement,~~ $\bar{S}_{C_{16}} = \{\bar{C}_{16,0}, \bar{C}_{16,1}, ..., \bar{C}_{16,15}\}$~~. Code words of set~~ $S_{C_{16}}$ ~~are from the level 16 of the code three of OVSF codes defined in document TS 25.213. The mapping of information bits to code words is shown in the table 8.~~

~~Table 8: Mapping of information bits to code words for biorthogonal (16, 5) code~~

~~Information bits~~	~~Code word~~
~~00000~~	~~$C_{16,0}$~~
~~00001~~	~~$\overline{C}_{16,0}$~~
~~00010~~	~~$C_{16,1}$~~
~~…~~	~~…~~
~~11101~~	~~$\overline{C}_{16,14}$~~
~~11110~~	~~$C_{16,15}$~~
~~11111~~	~~$\overline{C}_{16,15}$~~

~~Biorthogonal code words, $C_{16,i}$ and $\overline{C}_{16,i}$, are then punctured into length 15 by puncturing the 1st bit~~.

For information bits a_0, a_1, a_2, a_3, a_4 (a_0 is LSB and a_4 is MSB), the encoder structure is as following figure 13 (summation is modulo 2).

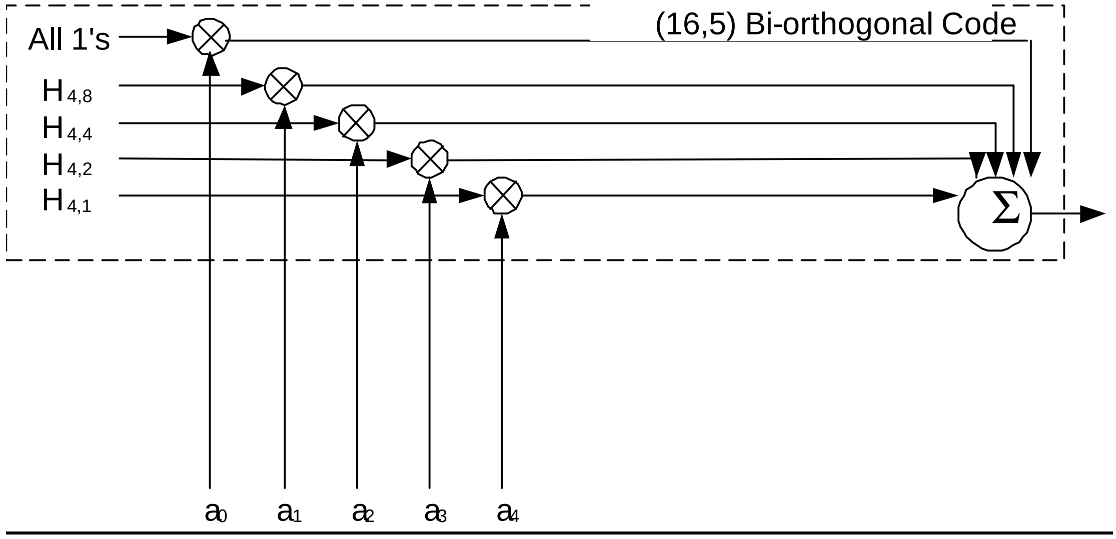

Figure 13: Encoder structure for (16,5) bi-orthogonal code

Then the output words of the (16,5) bi-orthogonal coder are punctured to length 15 by puncturing bits $c_{j,0}$ (the lsbs).

-The bits in the punctured code words are denoted by $b_{j,k}$, where subscript *j* indicates the code word and subscript *k* indicates bit position in the code word (*k* =14 corresponds to the MSB bit).

4.3.5 Mapping of TFCI words

4.3.5.1 Mapping of TFCI word

As only one code word for TFCI is needed no channel interleaving for the encoded bits are done. Instead, the bits of the code word are directly mapped to the slots of the radio frame as depicted in the figure 12. Within a slot the more significant bit is transmitted before the less significant bit.

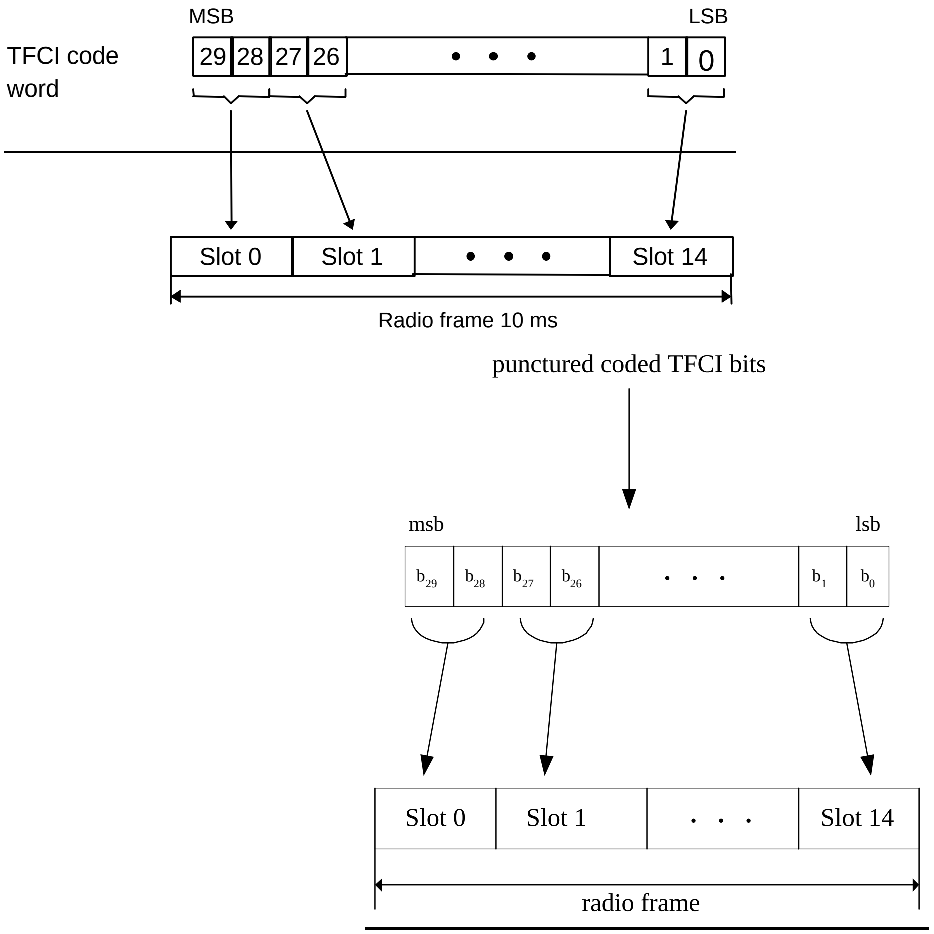

Figure 14~~2~~: Mapping of TFCI code words to the slots of the radio frame

For downlink physical channels whose SF is lower than 128, bits of the TFCI code words are repeated and mapped to slots as shown in the table 9. Code word bits are denoted as b_k^l, where subscript *k*, indicates bit position in the code word (*k* =29 is the MSB bit) and superscript *l* indicates bit repetition. In each slot transmission order of the bits is from left to right in the table 9.

Table 8~~9~~: Mapping order of repetition encoded TFCI code word bits into slots.

Slot	TFCI code word bits							
0	b_{29}^1	b_{29}^2	b_{29}^3	b_{29}^4	b_{28}^1	b_{28}^2	b_{28}^3	b_{28}^4
1	b_{27}^1	b_{27}^2	b_{27}^3	b_{27}^4	b_{26}^1	b_{26}^2	b_{26}^3	b_{26}^4
2	b_{25}^1	b_{25}^2	b_{25}^3	b_{25}^4	b_{24}^1	b_{24}^2	b_{24}^3	b_{24}^4
3	b_{23}^1	b_{23}^2	b_{23}^3	b_{23}^4	b_{22}^1	b_{22}^2	b_{22}^3	b_{22}^4
4	b_{21}^1	b_{21}^2	b_{21}^3	b_{21}^4	b_{20}^1	b_{20}^2	b_{20}^3	b_{20}^4
5	b_{19}^1	b_{19}^2	b_{19}^3	b_{19}^4	b_{18}^1	b_{18}^2	b_{18}^3	b_{18}^4
6	b_{17}^1	b_{17}^2	b_{17}^3	b_{17}^4	b_{16}^1	b_{16}^2	b_{16}^3	b_{16}^4
7	b_{15}^1	b_{15}^2	b_{15}^3	b_{15}^4	b_{14}^1	b_{14}^2	b_{14}^3	b_{14}^4
8	b_{13}^1	b_{13}^2	b_{13}^3	b_{13}^4	b_{12}^1	b_{12}^2	b_{12}^3	b_{12}^4
9	b_{11}^1	b_{11}^2	b_{11}^3	b_{11}^4	b_{10}^1	b_{10}^2	b_{10}^3	b_{10}^4
10	b_{9}^1	b_{9}^2	b_{9}^3	b_{9}^4	b_{8}^1	b_{8}^2	b_{8}^3	b_{8}^4
11	b_{7}^1	b_{7}^2	b_{7}^3	b_{7}^4	b_{6}^1	b_{6}^2	b_{6}^3	b_{6}^4
12	b_{5}^1	b_{5}^2	b_{5}^3	b_{5}^4	b_{4}^1	b_{4}^2	b_{4}^3	b_{4}^4
13	b_{3}^1	b_{3}^2	b_{3}^3	b_{3}^4	b_{2}^1	b_{2}^2	b_{2}^3	b_{2}^4
14	b_{1}^1	b_{1}^2	b_{1}^3	b_{1}^4	b_{0}^1	b_{0}^2	b_{0}^3	b_{0}^4