APPENDIX B - SAMPLE BOSE-CHAUDHURI-HOCQUENGHEM ERROR-CORRECTING CODE AND 23 HEX ID CALCULATIONS

NOTE: The information in the Appendix is not aligned with the current text in the main body of this document and requires an update.

B.1 Sample 48-Bit BCH Code Calculation

The error-correcting code used in 406 MHz messages is a shortened form of a (255,207) Bose-Chaudhuri-Hocquenghem (BCH) code. The shortened form (250,202) consists of 202 bits of data followed by a 48-bit sextuple error-correcting code. The code is used to detect and correct up to six errors in the entire 250-bit pattern (bits 1 through 250 of the 406 MHz message).

Note: For the purpose of error correction, all calculations shall be performed with the full 255 length code. Therefore, 5 zeros are placed before the 202 data bits to form the 207-bit pattern of the (255,207) BCH code. These padding zeros do not affect the generation of the BCH code as described below.

For the (250,202) BCH code, a generator polynomial g(X) (the same as for (255,207) BCH code) is defined as follows:

$$g(X) = LCM(m_1(X), m_3(X), m_5(X), m_7(X), m_9(X), m_{11}(X))$$

where LCM = Least Common Multiple.

In the above case:

$$m_1(X) = X^8 + X^4 + X^3 + X^2 + 1$$

$$m_3(X) = X^8 + X^6 + X^5 + X^4 + X^2 + X + 1$$

$$m_5(X) = X^8 + X^7 + X^6 + X^5 + X^4 + X + 1$$

$$m_7(X) = X^8 + X^6 + X^5 + X^3 + 1$$

$$m_9(X) = X^8 + X^7 + X^5 + X^4 + X^3 + X^2 + 1$$

$$m_{11}(X) = X^8 + X^7 + X^6 + X^5 + X^2 + X + 1$$

from which

$$\begin{aligned} g(X) &= m_1(X), m_3(X), m_5(X), m_7(X), m_9(X), m_{11}(X) \\ &= X^{48} + X^{47} + X^{46} + X^{42} + X^{41} + X^{40} + X^{39} + X^{38} + X^{37} + X^{35} + X^{33} + X^{32} + X^{31} \\ &\quad + X^{26} + X^{24} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{17} + X^{16} + X^{13} + X^{12} + X^{11} + X^{10} + X^7 + X^4 + X^2 \\ &\quad + X + 1 \end{aligned}$$

a determination of g(X) results in the following 49-bit binary number

$$g(X) = 1110001111110101110000101110111110011110010010111$$

To generate the BCH code, an information polynomial, m(x) is formed from the 202 data bits as follows:

$$m(X) = b_1 X^{201} + b_2 X^{200} + \cdots + b_{201} X + b_{202}$$

where b_1 - is the first bit and b_{202} - is the last bit of the digital message.

m (X) is then extended to 250 bits by filling the least significant bits with 48 "0". The resulting 250-bit binary string is then divided by g(X) and the remainder, r(X), becomes the BCH code (the quotient portion of the result of the module-2 binary division is discarded).

The above process may be clarified by the following example. Suppose, that the digital message in the Minimum Requirement main field consists of the following data (decimal notation):

Digital Message	Decimal Data	Binary Data	Bits in Message
TAC number	230	0000000011100110	1 to 16
Serial Number	573	00001000111101	17 to 30
Country code	201	0011001001	31 to 40
Status of homing device	1	1	41
RLS function	0	0	42
Test protocol	0	0	43
Encoded GNSS Location	See below	See below	44 to 90
Vessel ID	0	47 Bits all 0's	91 to 137
Beacon Type	0	000	138 to 140
Spare bits	16383 (all 1's)	11111111111111	141 to 154

Message also contains encoded GNSS location. This example uses the following values of position data:

Current latitude 48,793153539336956 °N
Current longitude 69,00875866413116 °E

Encoded GNSS location in binary notation (converted in accordance with Appendix C):

Current latitude 0 0110000 110010110000110
Current longitude 0 01000101 000000100011111

In this example Rotating Field 1 (C/S G.008 Objective Requirements) is used, containing the following data:

Digital Message	Decimal Data	Binary Data	Bits in Message
Rotating Field Identifier	0	0000	155 to 158
Elapsed Time since activation	1 hour 27 mins	000001	159 to 164
Time from last encoded location	6 mins 24 sec	00000000110	165 to 175
Altitude of encoded location	430,24 metres	0000110100	176 to 185
Dilution of precision	HDOP<1 VDOP <2	00000001	186 to 193
Activation notification	Manual	00	194 to 195
Remaining battery capacity	>75%	101	196 to 198
GNSS status	3D	10	199 to 200
Spare bits	0	00	200 to 202

Thus, digital message in binary and hexadecimal notation will be the following:

Bits 1-202	0000 0000 1110 0110 0000 1000 1111 0100 1100 1001 1000 0110 0001 1001 0110 0001 1000 1000 1010 0000 0100 0111 1100 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 1111 1111 1111 1100 0000 0001 0000 0000 1100 0001 1010 0000 0000 1001 0110 00
Hex Message From 202 bits	 00E608F4C986196188A047C000000000000FFFC0100C1A00960
Cospas-Sarsat Ground Segment Representation* (MF #90 Ref. C/S A.002)	0039823D32618658622811F0000000000003FFF004030680258 (two leading 0 bits plus bits 1-202)

* As the length of the message in bits is not divisible by 4, the message is augmented, for ground processing purposes, with two leading binary ‘0’s, which results in this revised hexadecimal message.
The division * described above is shown in Figure A1, and results in a 49-bit remainder of:

0010010010010101001001111110001010111101001001001

The most significant bit position of the remainder will always be a "0" and is deleted to obtain the 48-bit BCH code.

Thus BCH Error-Correcting Code:

010010010010101001001111110001010111101001001001

REFERENCE

[*] Modulo 2 division prohibits a "borrow" in the subtraction portion of the long division.

An Introduction to Error Correcting Codes, Shu Lin, Prentice Hall 1970

```
       5 "0"                                    Bits 0-202                                                         Bits (203-250)
                                               (Data bits)                                                           48 «0»
m(x)=  00000 000000001110011000001000111101001100100110000110000110010110000110001000101000001000111   …   01000000000100101 1000 000000000000000000000000000000000000000000000000
g(X)=             1110001111110101110000101110111110011110010010111
                  000001011111110100110110001001100001100001010010111000
                       1110001111110101110000101110111110011110010010111
                       01011100010100110000011000101100100101000001011110
                        1110001111110101110000101110111110011110010010111
                        01011011010100111100111010110110101101100110010011
                         1110001111110101110000101110111110011110010010111
                         01010101010100100101111110000010111100101000001001
                          1110001111110101110000101110111110011110010010111
                          01001001010100010111110111101010011110110100111100
                           1110001111110101110000101110111110011110010010111
                           01110001010101110011100100111011011010001101010110
                            1110001111110101110000101110111110011110010010111
                            00000001010110111011000010011001010011111110000010100010
                                   1110001111110101110000101110111110011110010010111
                                   01001110001011011000111001001000011011100001101011
                                    1110001111110101110000101110111110011110010010111
                                    0111111110101110110111100111111110100001001111111000
                                     1110001111110101110000101110111110011110010010111
                                     0001110010101000011111100001000100011010101101111000
                                        1110001111110101110000101110111110011110010010111
                                        000001101011011000110010011001110100101111110111100100
                                             1110001111110101110000101110111110011110010010111
                                             001101010011001110001110000001101110000010111001101
                                               1110001111110101110000101110111110011110010010111
                                               0011011100111011111111010111101000001110010101101011

…
                                                              01110000011000011010000011010100110011001010010110
                                                               1110001111110101110000101110111110011110010010111
                                                               0000001100110110100000110100011000000111000000001000000
                                                                     1110001111110101110000101110111110011110010010111
                                                                     001011100101010100010011011011100101111001101011100
                                                                       1110001111110101110000101110111110011110010010111
                                                                       010110101010000110001111010101101110011111110010110
                                                                        1110001111110101110000101110111110011110010010111
                                                                        01010110101101101101110001000010010100011000000010
                                                                         1110001111110101110000101110111110011110010010111
                                                                         01001110100110000111101001101011001111010100101010
                                                                          1110001111110101110000101110111110011110010010111
                                                                          01111110110001010011011000111001111001001101111010
                                                                           1110001111110101110000101110111110011110010010111
                                                                           0001111001111111101011101001110001010111111101101000
                                                                              1110001111110101110000101110111110011110010010111
                                                                              0001000000000100010110110000011010010000111111111000
                                                                                 1110001111110101110000101110111110011110010010111
                                                                                 01100011101100000111001010000110100100011011011110
                                                                                  1110001111110101110000101110111110011110010010111
                                                           49 bits remainder =    0010010010010101001001111110001010111101001001001
                                                                                 |<--------  BCH Code (last 48 bits)  ------->|
```

Figure B-1: Sample 48-Bit BCH Error-Correcting Code Calculation
(for easier viewing the calculation is reduced and divided into 2 parts)

B.2 Sample 23 Hex ID Calculation

Using the same data as above the 23 Hex ID can be generated as shown below. It should be noted that the 23 Hex ID cannot be formed directly from the transmitted message as additional bits have to be added to it and the order of the digital message data has to be changed.

Referring to T.018 Section 3.6 and Table 3.9 and using the digital message data from above results in the following:

23 Hex ID Bit	No Bits	Data Content	Message Content
1	1	Fixed Binary ‘1’	1
2 to 11	10	C/S Country Code ‘201’	0011001001
12	1	Fixed Binary ‘1’	1
13	1	Fixed Binary ‘0’	0
14	1	Fixed Binary ‘1’	1
15 to 30	16	C/S TAC No ‘230’	0000000011100110
31 to 44	14	Beacon Serial Number ‘573’	00001000111101
45	1	Test Protocol Flag	0
46 to 48	3	Aircraft / Vessel ID Type ‘N/A’	000
49 to 92	44	Aircraft / Vessel ID	44 Bits all ‘0’s

1001 1001 0011 0100 0000 0011 1001 1000 0010 0011 1101 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

Which in turn gives a final 23 Hex ID of:

23 Hex ID **9934039823D000000000000**

Which it can be seen is very different to the first 23 Hex characters of the Digital Message Data

00E608F4C986196188A047C

Finally, for completeness it can be noted that the unique 15 Hex ID for the same beacon is obtained by truncating the 23 Hex ID, as described in section 3.6, which would give a 15 Hex ID of:

9934039823D0000

- END OF APPENDIX B -