Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**
Title	**Errata and points for clarification in 802.16a OFDM PHY specification**
Date Submitted	**2003-01-09**
Source(s)	Neil Shipp Commsonic Ltd St. John's Innovation Center, Cambridge, CB4 0WS,UK Voice: +44 1223 871071 Fax: +44 1223 871071 mailto:nrs@commsonic.com
Re:	Call for Contributions on Project 802.16d: IEEE Standard 802.16: 2-11 GHz System Profiles and Corrections of Errors in Base Standard. 18 Dec. 2002. **Errata** in IEEE 802.16a or 802.16c, or those **in 802.16 not corrected in 802.16a or 802.16c**.
Abstract	Corrections for some Errata in 802.16a and also some suggestions for clarification.
Purpose	Provide text that should be inserted by the 802.16d amendment to 802.16a.
Notice	
Release	
Patent Policy and Procedures	

Errata and points for clarification in 802.16a OFDM PHY specification

Neil Shipp
Commsonic Ltd

Section 8.4.3.1 p147 to p148

Use of the word ‘allocation’ has become a bit confusing in the OFDM PHY section. I think using symbol, burst and frame as appropriate will be much clearer. In this section I believe ‘burst’ should replace ‘allocation’.

Section 8.4.3.2.1 p150 lines 26 to 29

‘A single 0x00 tail byte is appended to the end of each . In this section I believe ‘burst’ should replace ‘allocation’.

Section 8.4.3.3 p156 line 26

‘…coded bits per the specified allocation, Ncbps’ – this defines ‘allocation’ as a single symbol. Ncbps would seem to stand for Number of Coded Bits Per Symbol so use ‘symbol’ rather than ‘allocation’.

Section 8.4.3.3 p157 line 15 in Table 116al

Several mistakes in the table, 98 should be 96, 768 should be 576 and 384 should be 288.

Section 8.4.3.3 p157 line 22

I think this should read ‘first bit out of the interleaver shall map to the MSB of the constellation’.
This is consistent with the example data on p160. For all other blocks the MSB is always passed out first so for consistency it should really be the same here.

Section 8.4.3.4.1 p158 line 33-34

This should explicitly define b0 as the LSB to make the constellations consistent with 802.16-2001.

8.4.3.5 p160 line 7 onwards

I believe the interleaved data example is wrong and should be:
Interleaved Data
EE 73 2F A7 38 26 2A 66 BB F4 98 A7 38 46 B6 FB 59 90 7C ED CD 8D FA D5 23 AC EE 14 8F AD D0 67 B8 68 A7 D4 D3 10 23 8D C0 63 BB F2 06 2B 4F E0
I think the sequence in D7 has comes from a misinterpretation of the interleaving spec in Section 8.4.3.3. Equation (44) in combination with the definitions of ‘m’ and ‘k’ means take every 16^{th} bit of the input array and put it in the output. If instead you allow ‘m’ to index the input array and ‘k’ to index the output then you will wrongly form the output from every 24^{th} bit of the input (for QPSK with Ncbps=384) which is what the current example has done.

And of course you’ll need some new Mapped data:

Carrier Mapped Data (Index: I Q) x 1/sqrt(2)
-100:-1 -1, -99:-1 1, -98:-1 -1, -97:-1 1, -96:1 -1, -95:-1 -1, -94:1 1, -93:-1 -1, -92:1 1, -91:-1 1,
-90:-1 -1, -89:-1 -1, -88:-1 1, -87:-1 1, -86:1 -1, -85:-1 -1, -84:1 0, -83:1 1, -82:-1 -1, -81:-1 1,
-80:1 1, -79:1 1, -78:-1 1, -77:1 -1, -76:-1 1, -75:1 1, -74:-1 1, -73:-1 1, -72:-1 1, -71:1 -1,

-70:-1 1, -69:1 -1, -68:-1 1, -67:-1 1, -66:-1 -1, -65:-1 1, -64:-1 -1, -63:-1 -1, -62:-1 -1, -61:1 -1,
-60:-1 0, -59:1 1, -58:-1 1, -57:1 -1, -56:-1 1, -55:1 1, -54:-1 1, -53:-1 1, -52:1 -1, -51:-1 -1,
-50:1 1, -49:-1 -1, -48:-1 1, -47:1 1, -46:1 -1, -45:1 1, -44:1 -1, -43:-1 1, -42:-1 1, -41:-1 -1,
-40:1 -1, -39:-1 1, -38:-1 -1, -37:-1 -1, -36:1 0, -35:-1 1, -34:-1 -1, -33:1 -1, -32:1 -1, -31:-1 1,
-30:1 -1, -29:-1 1, -28:1 -1, -27:1 1, -26:1 1, -25:1 -1, -24:-1 -1, -23:-1 -1, -22:1 1, -21:-1 -1,
-20:-1 1, -19:-1 -1, -18:1 -1, -17:-1 -1, -16:1 1, -15:-1 -1, -14:1 -1, -13:-1 1, -12:-1 0, -11:1 1,
-10:-1 -1, -9:1 -1, -8:-1 -1, -7:-1 -1, -6:-1 1, -5:-1 1, -4:-1 -1, -3:1 -1, -2:1 -1, -1:1 -1,
0:0 0, 1:1 1, 2:-1 1, 3:1 1, 4:-1 -1, 5:-1 1, 6:-1 1, 7:-1 -1, 8:1 1, 9:-1 -1,
10:-1 1, 11:-1 -1, 12:1 0, 13:-1 1, 14:1 1, 15:1 -1, 16:1 -1, 17:1 1, 18:-1 1, 19:1 1,
20:-1 -1, 21:-1 -1, 22:-1 1, 23:-1 1, 24:-1 -1, 25:1 -1, 26:-1 -1, 27:1 -1, 28:1 1, 29:1 1,
30:1 -1, 31:-1 1, 32:1 -1, 33:-1 -1, 34:-1 1, 35:-1 -1, 36:1 0, 37:-1 1, 38:1 1, 39:1 -1,
40:-1 1, 41:-1 1, 42:1 1, 43:-1 1, 44:-1 1, 45:1 -1, 46:-1 -1, 47:-1 -1, 48:1 -1, 49:1 -1,
50:1 1, 51:-1 -1, 52:1 -1, 53:1 1, 54:-1 -1, 55:1 1, 56:1 -1, 57:1 1, 58:1 1, 59:1 1,
60:1 0, 61:-1 1, 62:1 1, 63:-1 -1, 64:-1 1, 65:1 1, 66:-1 -1, 67:1 -1, 68:-1 -1, 69:1 1,
70:1 1, 71:1 1, 72:1 -1, 73:-1 1, 74:1 1, 75:-1 -1, 76:-1 1, 77:-1 -1, 78:-1 1, 79:-1 -1,
80:-1 -1, 81:-1 -1, 82:1 1, 83:-1 1, 84:1 0, 85:1 1, 86:1 1, 87:1 -1, 88:-1 1, 89:1 1,
90:-1 1, 91:-1 1, 92:-1 -1, 93:1 -1, 94:1 1, 95:-1 -1, 96:-1 -1, 97:-1 -1, 98:-1 1, 99:1 1, 100:1 1,

8.4.4.1 p165 line 12

The HCS code definition would benefit from an example. I think there is more than one way to interpret this and of course if you do the reverse at your demodulator then you'll be fine but won't interwork with someone who designed it the other way.

My interpretation of this is that it is a Hamming type polynomial division, where the transmitter takes the RateID and Length bytes, divides by the generator polynomial and uses the remainder as the HCS code. At the receiver dividing the RateID+Length+HCS bytes by the generator then gives remainder 0 if correct.

So if RateID=1 and Length=204 symbols then we encode the byte sequence [0x10 0xCC] and should get 0x3D (decimal 61) as the HCS byte. The hardware to do this is a simple shift register and 3 XOR gate structure.