IEEE P802.11
Wireless LANs

Replacement Description of CCK

Date: November 4, 1998

Author: Matthew B. Shoemake, Chris Heegard and Stanley Ling

Alantro Communications
141 Stony Circle, Suite 210
Phone: 707-521-3060
Fax: 707-521-3066
e-mail: shoemake@alantro.com, heegard@alantro.com, and sling@alantro.com

Abstract

This document presents a replacement description of the CCK coding scheme in P802.11B/D1.0. This description uses standard notation that is widely used in industry and academia. This text has been written to replace replace sections 18.4.6.4 – 18.4.6.6.

The authors believe that this description of CCK is clearer than the current desciption in the draft text and thus minimizes the possibility of erroneous interpretation. This will increase the likelihood of interoperability of IEEE 802.11 based products that implement the high rate PHY.

18.4.6.4 DSSS/CCK Data Modulation and Modulation Rate

The CCK modulation scheme may be described in terms of a block code over the integers modulo 4. Figure A shows the structure of the CCK encoding scheme.

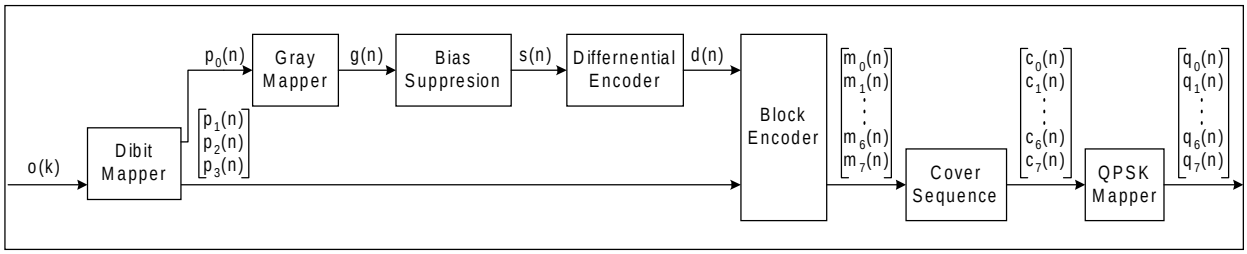

Figure A - CCK Encoder Block Diagram

18.4.6.4.1 CCK Dibit Mapper

The CCK Dibit Mapper is one of two sections of the CCK encoder that is a function of the data rate, i.e. 5.5 Mbps or 11 Mbps. The other section that is dependent on the data rate is the CCK Cover Sequence.

Each octet of data to be CCK encoded is represented by *o(k)*, where *k* is an integer time index that starts at zero. The least significant bit of octet *k* is represent by $b_0(k)$, and the most significant bit of octet *n* is reprsented by $b_7(k)$. This is shown in Equation A below. The CCK Dibit mapper maps input octets, represented as a vector of bits, to elements of Z_4, the integers modulo 4. All sections of the CCK Encoder after the Dibit Mapper do arithmetic modulo 4.

Equation A – Octet Representation

$$o(k) = [b_0(k)\ b_1(k)\ b_2(k)\ b_3(k)\ b_4(k)\ b_5(k)\ b_6(k)\ b_7(k)]^T, \text{ where } b_i(k) \in \{0,1\} = Z_2$$

In 11Mbps mode, the CCK Dibit Mapper is defined by Equation B. While the input octets are indexed by *k*, the output is indexed by *n*. In 11Mbps CCK mode, the Dibit Mapper produces one set of outputs for each octet input, thus *n = k*. The output of the Dibit Mapper is represented as *o(k)* times as matrix. For every eight bits that are input to the Dibit Mapper in 11Mbps mode, eight QPSK chips are produced at the output of the QPSK Mapper to yield a data rate of one one bit per complex chip.

Equation B - 11Mbps CCK Precoder

$$\begin{bmatrix} p_0(n) \\ p_1(n) \\ p_2(n) \\ p_3(n) \end{bmatrix} = \begin{bmatrix} 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} b_0(n) \\ b_1(n) \\ b_2(n) \\ b_3(n) \\ b_4(n) \\ b_5(n) \\ b_6(n) \\ b_7(n) \end{bmatrix}, \text{ where } p_i(n) \in \{0,1,2,3\} = Z_4$$

In 5.5 Mbps CCK mode, the encoding of one octet of data generates a pair of outputs from the CCK Dibit Mapper, where one output is defined as a vector of four dibits as shown in Equation C. The first output is indexed by *n = 2k*, and the second is indexed by *n = 2k + 1*. Each output of the Dibit Mapper is operated on independently by the following CCK encoder stages. Thus for every 8 bits input to the Dibit Mapper, there are 16 QPSK chips produced at the output of the QPSK Mapper, which yields a data rate of one-half bit per complex chip.

Each of the two outputs of the CCK Dibit mapper for 5.5Mbps CCK are shown in Equation C. The first output is a function of the least significant four bits of the input octet, *o(k)*, and the second output of the CCK Dibit Mapper is a function of the most significant four bits of the input octet, *o(k)*.

Equation C - 5.5 Mbps CCK Dibit Mapper

$$\begin{bmatrix} p_0(n) \\ p_1(n) \\ p_2(n) \\ p_3(n) \end{bmatrix} = \begin{bmatrix} 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} b_0(n) \\ b_1(n) \\ b_2(n) \\ 0 \\ 0 \\ 0 \\ b_3(n) \\ 0 \end{bmatrix}$$

First Output ($n = 2k$)

$$\begin{bmatrix} p_0(n) \\ p_1(n) \\ p_2(n) \\ p_3(n) \end{bmatrix} = \begin{bmatrix} 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} b_4(n) \\ b_5(n) \\ b_6(n) \\ 0 \\ 0 \\ 0 \\ b_7(n) \\ 0 \end{bmatrix}$$

Second Output ($n = 2k + 1$)

18.4.6.4.2 CCK Gray Mapping

Each $p_0(n)$ output of by the CCK Dibit Mapper is Gray encoded using the map shown in Table A.

Table A - CCK Gray Mapping

$p_0(n)$	$g(n)$
0	0
1	1
2	3
3	2

18.4.6.4.3 CCK Bias Suppressor

The CCK Bias Suppresor adds 2 to $g(n)$ modulo 4, if n is even. This may be equivalently represented by Equation D.

Equation D – CCK Bias Suppressor

$$s(n) = (g(n) + 2*n) \bmod 4$$

Figure B shows an implementation of the bias suppressor.

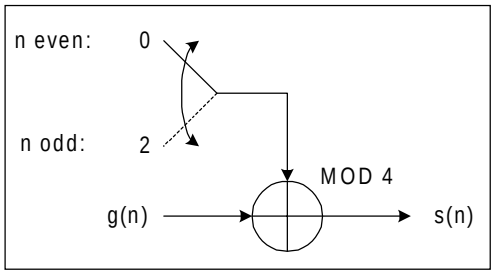

Figure B - CCK Bias Supressor

18.4.6.4.4 CCK Differential Encoder

The output of the bias supressor is differentially encoded by Equation E.

Equation E – CCK Differential Encoder

$$d(n) = (s(n) + s(n-1)) \bmod 4$$

Figure C shows one relization of this differentiator.

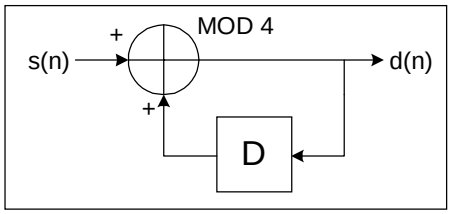

Figure C - CCK Differential Encoder

Prior to encoding the first octet of data, *o(k)*, the CCK Differential Encoder must be intialized. The CCK Differential Encoder shall be initialized to the phase of the last Barker symbol as mapped according to Figure D.

18.4.6.4.5 CCK Block Encoder

The output of the differential encoder, *d(n)*, and the outputs of the precoder that are not input to the Gray mapper, $p_1(n)$, $p_2(n)$, and $p_3(n)$, are input to the CCK Block encoder. The CCK Block encoder may be represented as a matrix multiply. Equation F shows the multiplication operation. All arithmetic is performed modulo 4.

Equation F - CCK Block Encoder

$$\begin{bmatrix} m_0(n) \\ m_1(n) \\ m_2(n) \\ m_3(n) \\ m_4(n) \\ m_5(n) \\ m_6(n) \\ m_7(n) \end{bmatrix} = \begin{bmatrix} d(n) & p_1(n) & p_2(n) & p_3(n) \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

18.4.6.4.6 CCK Cover Sequence

The cover sequence varies based on the modulation rate. For CCK at 11Mbps, the constant sequence [0 0 0 2 0 0 2 0] is added to the output of the CCK Block encoder. This is shown in Equation G.

Equation G - 11Mbps CCK Cover Sequence

$$\begin{bmatrix} c_0(n) \\ c_1(n) \\ c_2(n) \\ c_3(n) \\ c_4(n) \\ c_5(n) \\ c_6(n) \\ c_7(n) \end{bmatrix} = \begin{bmatrix} m_0(n) \\ m_1(n) \\ m_2(n) \\ m_3(n) \\ m_4(n) \\ m_5(n) \\ m_6(n) \\ m_7(n) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2 \\ 0 \\ 0 \\ 0 \\ 2 \end{bmatrix}$$

In 5.5 Mbps CCK mode, the cover sequence is [1 0 1 2 1 0 3 0]. This is shown in Equation H below.

Equation H - 5.5Mbps CCK Cover Sequence

$$\begin{bmatrix} c_0(n) \\ c_1(n) \\ c_2(n) \\ c_3(n) \\ c_4(n) \\ c_5(n) \\ c_6(n) \\ c_7(n) \end{bmatrix} = \begin{bmatrix} m_0(n) \\ m_1(n) \\ m_2(n) \\ m_3(n) \\ m_4(n) \\ m_5(n) \\ m_6(n) \\ m_7(n) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 2 \\ 1 \\ 0 \\ 3 \\ 0 \end{bmatrix}$$

18.4.6.4.7 CCK QPSK Mapper

The output of the CCK Cover Sequence block is mapped onto QPSK. Each $c_i(n)$ is mapped onto one QPSK symbol in order of increasing *i*. Each $c_i(n)$ takes on values in Z_4, the integers modulo 4. Figure D shows how each value is mapped onto QPSK.

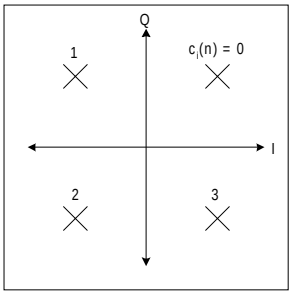

Figure D – CCK QPSK Mapping