IEEE P802.11
Wireless LANs

Codeword Description for the Harris-Lucent Updated Compromise Proposal for TGb

Date: July 1, 1998

Author: Mark Webster and Carl Andren
Harris Semiconductor

Jan Boer and Richard van Nee
Lucent Technologies

Abstract

HARRIS and LUCENT detail both the 16 chip codes presented in the initial compromise-proposal offering and their foundational 8 chip codes. They include 5.5Mbps and 11Mbps. The details on 8 chip codes and data dates are described in section 3 of this document.

The performance numbers remain unchanged because the packet-error-rate numbers truly represented the 8 chip complementary codes in the first place. Since the 16 chip codes are a simple extension of the 8 chip codes, HARRIS/LUCENT had already collected the full suite of 8 chip packet-error-rate performance curves. The new 16 chip packet-error-rate analysis was being developed in stepping-stone fashion, with these later results building off the earlier 8 chip results. As such, HARRIS/LUCENT were in the process of generating a full suite of 16 chip code performance results, which we planned on releasing shortly. Early-on test results show a small advantage for the 16 chip codes. The updated proposal directly uses the 8 chip codes to realise more MAC-workable data rates.

1. INTRODUCTION

This section highlights the advantages/disadvantages of the 16 chip versus 8 chip complementary codes. Since the MAC incompatibility is so dominant in the following trade table, the decision has been made to sacrifice the small performance/complexity advantage of 16 chip codes in favour of 8 chip codes.

CODE	ADVANTAGES	DISADVANTAGES
CCK-16 **16 chip code**	1. 0.5 dB better Eb/No. 2. Delay spread performance 5% better. 3. Automatic Japanese MKK approval.	1. MAC incompatible data rates produced. 2. Highest data rate only 10.3 Mbps 3. Encode/decoder slightly more complex.
CCK **8 chip code**	1. MAC compatible data rates. 2. Highest data rate 11 Mbps.	1. Insignificant reduction in performance.

2. UPDATED HIGHLIGHTS

This section presents the updated key features.

The performance numbers remain unchanged because the packet-error-rate numbers truly represented the 8 chip complementary codes in the first place. Since the 16 chip codes are a simple extension of the 8 chip codes, HARRIS/LUCENT had already collected the full suite of 8 chip packet-error-rate performance curves. The new 16 chip packet-error-rate analysis was being developed in stepping-stone fashion, with these later results building off the earlier 8 chip results. As such, HARRIS/LUCENT were in the process of generating a full suite of 16 chip code performance results, which we planned on releasing shortly. Early-on test results show a small advantage for the 16 chip codes.

Notice the four architectures listed below.

TRANSMIT MODULATION	1. Retains QPSK chips at 11 Mcps for interoperability. 2. Uses 8 chip codewords above 2 Mbps. 3. Constructs codewords from complementary codes. 4. Serial complex-chip encoding used to enable high performance/complexity ratio. 5. Symbol's phase is differentially encoded to enable receiver PLL simplification. 6. Many data-rates capable with highest rate 11 Mbps.
RECEIVE ARCHITECTURE	1. Multiple performance/complexity architectures possible • ISI/ICI - DFE Equalizer (Doc:IEEE P802.11-98/47) • RAKE • RAKE with ISI DFE Equalizer • RAKE with ISI/ICI DFE Equalizer 2. RAKE/Equalizer is symbol-decision-based not chip-decision-based. 3. RAKE/Equalization not needed at fallback rates. 4. Differentially-coherent-phase symbol reception minimises acquisition time. 5. Fast-Walsh-like transform used for codeword correlation. 6. Reception is possible using a limited receiver, but the highest data rates would be degraded beyond that acceptable for high delay spread environments.
PACKET ERROR PERFORMANCE *Note: ALL results based on 8 chip CCK codes.*	1. Excellent performance in all environments. 2. Tolerates high multipath (MP) spreads. 3. Tolerates low SNR- extends range. 4. Six-finger channel matched filter used in RAKE simulations. 5. ISI/ICI-DFE Equalizer (64 byte packets at 11Mbps) • Noise - *5.5 dB for 10% PER* • MP - *186 nsec for 10% PER* • Noise plus MP - *21.2 dB for 20% PER* 6. RAKE (64 byte packets at 11 Mbps) • Noise—*5.5 dB for 10% PER* • MP—*90 nsec for 10% PER* • Noise plus MP—*15 dB for 20% PER* 7. RAKE-ISI Equalizer (64 byte packets at 11 Mbps) • Noise—*5.5 dB for 10% PER* • MP—*144 nsec for 10% PER* • Noise plus MP—*15 dB for 20% PER* 8. RAKE-ISI/ICI Equalizer (64 byte packets at 11 Mbps) • Noise—*5.5 dB for 10% PER* • MP—*333 nsec for 10% PER*

	• Noise plus MP—*15.5 dB for 20% PER*

3. 8 chip CCK

The fundamental building blocks of the 16 chip code were two 8 chip code segments. The updated proposal merely bypasses the concatenation of the two 8 chip segments and uses the 8 chip segments directly as the codewords. The 8 chip segments are complementary codes. The receiver merely uses the 8 chip correlates as detailed above, without the extra step of parity-bit resolution.

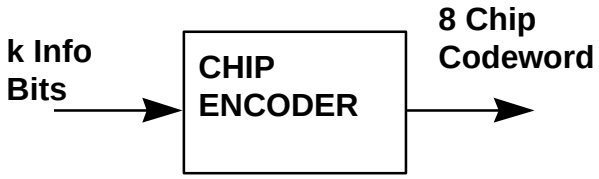

Figure 3.1 The basic encoder block.

PROPOSED DATA-RATE SPAN

The proposed data-rate capability of this coding scheme is listed in the Table 3.1. A symbol is 8 chips. In all the listed cases, 2 information bits select the quadriphase codeword sign. The phase is differentially encoded, easing receiver design.

There are additional rates that could be implemented using the coding approach. However, only 11 and 5.5 Mbps are proposed since the MAC requires data rates to be an integral of 500 KHz.

Table 3.1 Bit allocation.

# Info Bits Per Symbol	# Sign Bits for Symbol	# of Codeword Select Bits for Symbol	Rate Mbps
8	**2**	**6**	**11 Mbps**
4	**2**	**2**	**5.5**

3.1 CCK CODEWORD DESCRIPTION

This section describes the high-performing CCK codeword for both 5.5 Mbps and 11 Mbps. CCK is short for complementary code keying with 8 chips. 8 chips are used at all data rates except for the legacy 1 and 2 Mbps DSSS operation.

11 Mbps CCK

The description is given for the code length of 8 chips, where 256 possible sequences *c* can be constructed as follows, using 4 QPSK phases j_1 to j_4:

$$c = \{e^{j(j_1+j_2+j_3+j_4)}, e^{j(j_1+j_3+j_4)}, e^{j(j_1+j_2+j_4)},$$
$$-e^{j(j_1+j_4)}, e^{j(j_1+j_2+j_3)}, e^{j(j_1+j_3)}, -e^{j(j_1+j_2)}, e^{jj_1}\}$$

Note, j_1 is present in all 8 chips, so it simply rotates the entire code word. Hence, to decode these code set, one would need 64 correlators plus an additional phase estimation of the code that gave the largest correlation output. The correlation can be significantly simplified by using techniques like the fast Walsh transform. In fact, when the 4 input phases j_1 to j_4 are binary, then the complementary code set reduces to a modified Walsh code set, similar to the one used in Harris's original proposal.

Note that the information is encoded directly onto complex chips which cannot be cross-coupled corrupted by multipath since each channel finger has an $Ae^{j\theta}$ distortion. A single channel finger gain-scales and phase-rotates the signal. A gain scale and phase rotation of a complex chip still maintains I/Q orthogonality. This superior encoding technique avoids the corruption resulting from encoding half the information on the I-channel and the other half on the Q-channel, which easily cross-couple corrupts with the multipath finger's $Ae^{j\theta}$ phase rotation.

For 11 Mbps, j_1 provides the 2 bits quadriphase. Two information bits are used to generate 1-of-4 differential phases as done for the legacy 2 Mbps DSSS. j_2 to j_4: provide the 64 codewords, since 4^3 = 64.

For convenience, MATLAB code to generate the set is:

```
% Matlab code to produce the length 8 complementary code set
% output codes are in rows of matrix codeSet
encoding = [ 1 1 1 1;
   1 0 1 1;
   1 1 0 1;
   1 0 0 1;
   1 1 1 0;
   1 0 1 0;
   1 1 0 0;
   1 0 0 0];
phases= 2 ^ nSignBits;
codeSet=ones(nCodeWords,nCodeChips);
cnt=1;
for p0=0:0    % p0=0:phases-1,  % Sign bit
    for p1=0:phases-1,
        for p2=0:phases-1,
            for p3=0:phases-1,
               pha=(encoding*[p0 p1 p2 p3]')';
               codeSet(cnt,:)=codeSet(cnt,:).*exp(j*pha*2*pi/phases);
               cnt=cnt+1;
            end;
        end;
    end;
end;
codeSet(:,4)=-1*codeSet(:,4); codeSet(:,7)=-1*codeSet(:,7);
```

5.5 Mbps CCK

The complementary codes proposed by Lucent in [1] show promising results at the 11 Mbps rate. In order to maximise the re-use of hardware, we would like to use a subset of the codes used at the 11 Mbps rate in the 5.5 Mbps rate. Based on a limited search of the possible subsets, this memorandum describes one such subset which has shown good performance in the presence of multipath.

The length 8 complementary codes can be written as a function of four phase elements $\phi_1, \phi_2, \phi_3, \text{and} \phi_4$:

$$\mathbf{c}(\phi_1, \phi_2, \phi_3, \phi_4) = \left[e^{j(\phi_1+\phi_2+\phi_3+\phi_4)}, e^{j(\phi_1+\phi_3+\phi_4)}, e^{j(\phi_1+\phi_2+\phi_4)} \right.$$
$$\left. -e^{j(\phi_1+\phi_4)}, e^{j(\phi_1+\phi_2+\phi_3)}, e^{j(\phi_1+\phi_3)}, -e^{j(\phi_1+\phi_2)}, e^{j\phi_1} \right]$$

To generate the $2^8 = 256$ codewords needed to transmit data at 11 Mbps from this expression, the four phase parameters are each allowed to take on one of the four values $0, \pi/2, \pi, 3\pi/2$. This is similar to allowing each phase to be drawn from a QPSK constellation. In order to achieve a data rate of 5.5 Mbps, only $2^4 = 16$ codewords out of these 256 possible words are needed. In order to minimise the number of codeword correlators, we assume that ϕ_1 , which is common to

all "chips" and hence looks like a complex sign modulation, is allowed to take on all four values -- that is $0, p/2, p, 3p/2$. This assumption reduces the number of codeword correlators to only 4 and may obviate the need for a Fast Walsh Transform at this data rate.

To determine the other 3 parameters, a limited search over the valid code word values was performed in Matlab. Based on that search, the codewords with:

$$\begin{aligned} f_2 &= \frac{p}{2}, \frac{3p}{2} \\ f_3 &= 0 \\ f_4 &= 0, \; p \end{aligned}$$

were able to tolerate a large RMS multipath delay .

Finally, we note that with these choices for the phase, the expression for the complementary codes can be simplified to:

$$\mathbf{c}(f_1, f_2, 0, f_4) = e^{jf_1}\left[e^{j(f_2+f_4)}, e^{jf_4}, e^{j(f_2+f_4)}, -e^{jf_4}, e^{jf_2}, 1, -e^{jf_2}, 1\right] \; .$$

The following code words are generated by applying the phase choices listed in the equation above with $f_1 = 0, p/2, p, 3p/2$. **The top 4 rows are the 4 base codewords.** The additional rows show the 2 bit quadriphase modulation. The last chip identifies the quadriphase sign.

Codeword 1:	$1j$	1	$1j$	-1	$1j$	1	$-1j$	1
Codeword 2:	$-1j$	-1	$-1j$	1	$1j$	1	$-1j$	1
Codeword 3:	$-1j$	1	$-1j$	-1	$-1j$	1	$1j$	1
Codeword 4:	$1j$	-1	$1j$	1	$-1j$	1	$1j$	1
Codeword 5:	-1	$1j$	-1	$-1j$	-1	$1j$	1	$1j$
Codeword 6:	1	$-1j$	1	$1j$	-1	$1j$	1	$1j$
Codeword 7:	1	$1j$	1	$-1j$	1	$1j$	-1	$1j$
Codeword 8:	-1	$-1j$	-1	$1j$	1	$1j$	-1	$1j$
Codeword 9:	$-1j$	-1	$-1j$	1	$-1j$	-1	$1j$	-1
Codeword 10:	$1j$	1	$1j$	-1	$-1j$	-1	$1j$	-1
Codeword 11:	$1j$	-1	$1j$	1	$1j$	-1	$-1j$	-1
Codeword 12:	$-1j$	1	$-1j$	-1	$1j$	-1	$-1j$	-1
Codeword 13:	1	$-1j$	1	$1j$	1	$-1j$	-1	$-1j$
Codeword 14:	-1	$1j$	-1	$-1j$	1	$-1j$	-1	$-1j$
Codeword 15:	-1	$-1j$	-1	$1j$	-1	$-1j$	1	$-1j$
Codeword 16:	1	$1j$	1	$-1j$	-1	$-1j$	1	$-1j$

RAKE Architecture for 8 Chip

The conventional block diagram for a RAKE receiver is shown in Fig. 3.5. The signal is first channel matched filtered, coherently combining all the RAKE fingers. Second, codeword correlation is computed for a single finger output from the matched filter. Codeword correlation at a single finger is possible because the matched filter is located *in front* of the codeword correlator rather than *after* the codeword correlator.

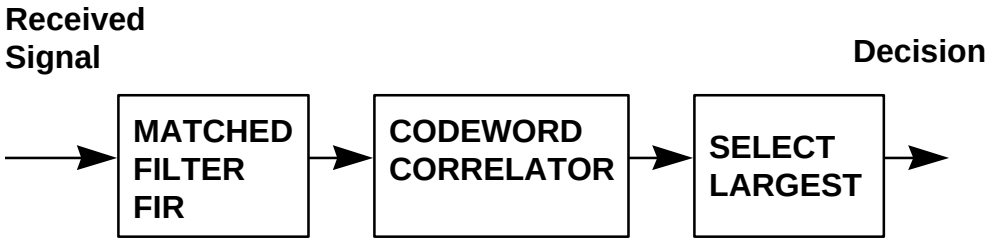

Figure 3.5 RAKE receiver architecture.

Usually the codeword consists of biphase chips, or +/-1. For complex chips the codewords are usually biphase on the I channel and biphase on the Q channel. In both these cases the codeword correlation are be computed using only adds and subtracts. Mathematically an N-chip vector dot-product is performed between the vector of received signal samples r_k and a the codeword vector c_k as shown in Eq. (3.1).

$$Correlation = \sum_{k=0}^{N-1} c_k^{*} r_k \qquad (3.1)$$

4. CONCLUSION

This submission has presented the details of the updated HARRIS/LUCENT proposal using 8 chip codewords. The older proposal with 16 chips is described in the appendix A. However, due to a data rate incompatibility with the MAC, HARRIS/LUCENT are updating the proposal to 8 chips complementary codes. The 8 chip codes have been detailed for both 5.5 Mbps and 11 Mbps.

In summary, HARRIS and LUCENT believe that the 8 chip codes are most optimum for the achieving superior 11Mbps data rate performance without any imposing any risks the MAC protocol or changes in the MAC specifications.

Appendix A 16 chip CODEWORD

This section provides additional details concerning the original 16 chip codes. The intent is to reveal how the 16 chip codes were fundamentally composed of two 8 chip code segments. This explains why HARRIS/LUCENT can effortlessly update our 16 chip proposal to 8 chips. These codes are an advanced version of the original QMBOK codes.

Hopefully, the creativity inherent in this approach is appreciated.

16 chip Encoder

For the 16 chips codes the natural data rates are listed in Table A.1. The column definitions will now become clear.

Table A.1 Bit allocation.

# Info Bits Per Symbol	# of Code Word Bits	# Bits per Segment	# Sign Bits for Segment	# of Codeword Select Bits for Segment	Rate Mbps
15	16	8	2	6	10.3125
13	14	7	2	5	8.9375
11	12	6	2	4	7.5625
9	10	5	2	3	6.1875
7	8	4	2	2	4.8125
5	6	3	2	1	3.4375

The highest level encoder block diagram is shown in Fig. A.1.

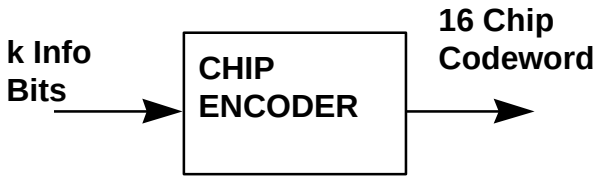

Figure A.1 The basic encoder block.

The codeword possesses an underlying structure as shown in Fig. A.2. Two 8 chip segments are jointly encoded.

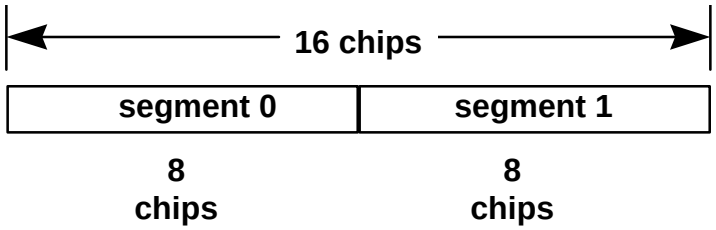

Figure A.2 The underlying codeword structure.

The two subsegments are jointly encoded by generating a parity bit. The parity bit correlates all the codeword bits together. The codewords bits are split into two halves as shown in Fig. A.3. (k+1)/2 bits are used to generate the first 8 chip segment, and similarly for the second 8 chips.

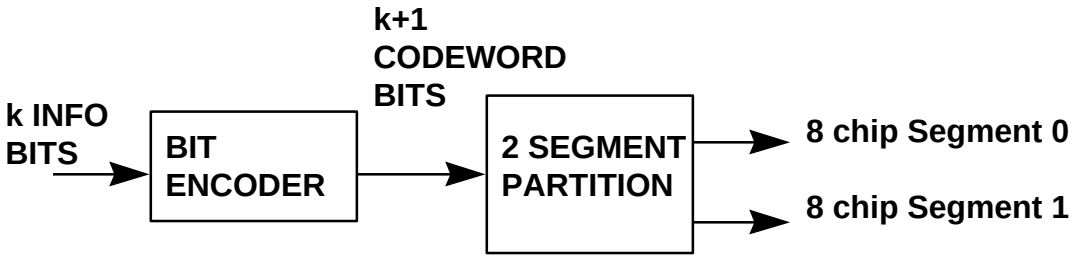

Figure A.3 Each segment uses (k+1)/2 bits.

The m=(k+1)/2 bits assigned to each 8 chip segment are encoded as shown in Fig. A.4. One or two bits n can be used to either biphase modulate or quadriphase modulate the codeword. The remaining segment bits select one of 2^(m-n) 8-chip codewords.

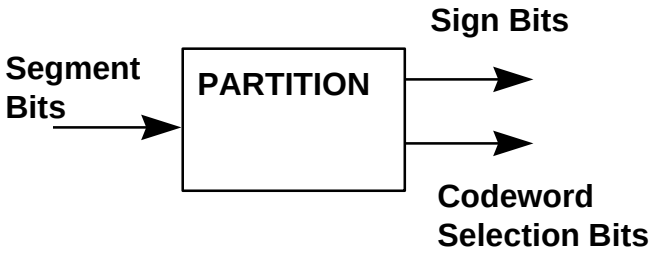

Figure A.4 Decomposition of the segment bits into a complex sign and a codeword.

Theoretically, there is no doubt that the codewords are indeed 16 chips, since optimal detection requires all 16 chips be jointly detected. However, the underlying structure allows optimal 16-chip detection with low complexity. This is also ideal for simplified equalizer implementation.

Optimal 16 chip RAKE Correlator

The conventional block diagram for a RAKE receiver is shown in Fig. A.5. The signal is first channel matched filtered, coherently combining all the RAKE fingers. Second, codeword

correlation is computed for a single finger output from the matched filter. Codeword correlation at a single finger is possible because the matched filter is located *in front* of the codeword correlator rather than *after* the codeword correlator.

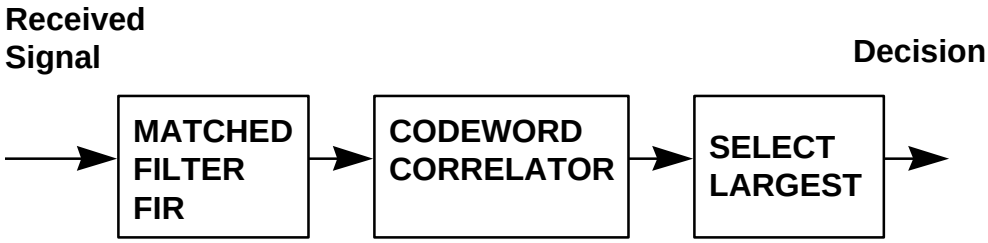

Figure A.5 RAKE receiver architecture.

Usually the codeword consists of biphase chips, or +/-1. For complex chips the codewords are usually biphase on the I channel and biphase on the Q channel. In both these cases the codeword correlation are be computed using only adds and subtracts. Mathematically an N-chip vector dot-product is performed between the vector of received signal samples r_k and a the codeword vector c_k as shown in Eq. (A.1).

$$Correlation = \sum_{k=0}^{N-1} c_k^* r_k \qquad \text{(A.1)}$$

The dot-product can optimally be partitioned without since it is an additive operation as shown in Eq. (A.2). The miniature dot-product can be computed for the first codeword half. A miniature dot-product can be computed for the second codeword half. Afterwards, the output of the two halves can be added as shown in Fig. A.6.

$$Correlation = \sum_{k=0}^{N/2-1} c_k^* r_k + \sum_{k=N/2}^{N-1} c_k^* r_k \qquad \text{(A.2)}$$

The recommended encoding structure allows this first-half/second-half dot-product with a tentative first-half/second-half detection. The parity bit is used to combine the two halves optimally to form the best 16 chip decision.

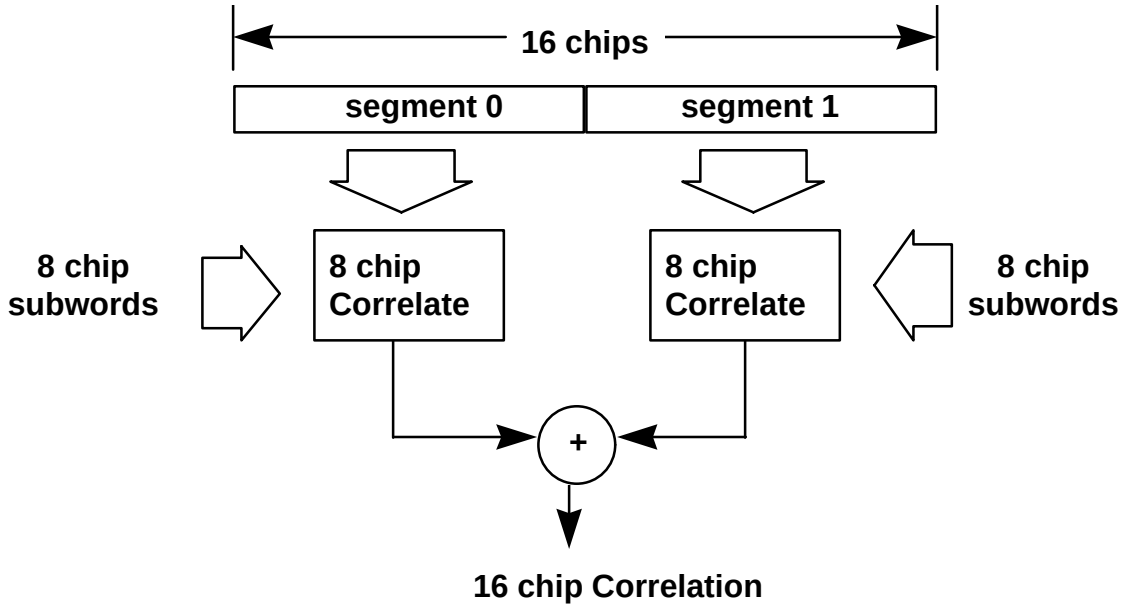

Figure A.6 Forming a 16 chip correlation using a half-word decomposition.

First, the soft decisions are generated for the first 8 chips. Only two soft decisions are retained from the first half—the largest even-bit-count decision and the largest odd-bit-count decision. Second, the soft decisions are generated for the last 8 chips. Again, only two soft decisions are retained—the largest even-bit-count decision and the largest odd-bit-count decision. The final decision is made by computing the possible combinations of first-half/second-half soft decisions. The optimal decision is the largest combination consistent with the parity-encoding scheme—even parity or odd parity.

If even parity encoding is used, the largest between odd-odd combination and the even-even combination is the best decision. If odd parity encoding is used, the largest between odd-even combination and the even-odd combination is the best decision. This operation is illustrated in Fig. A.7.

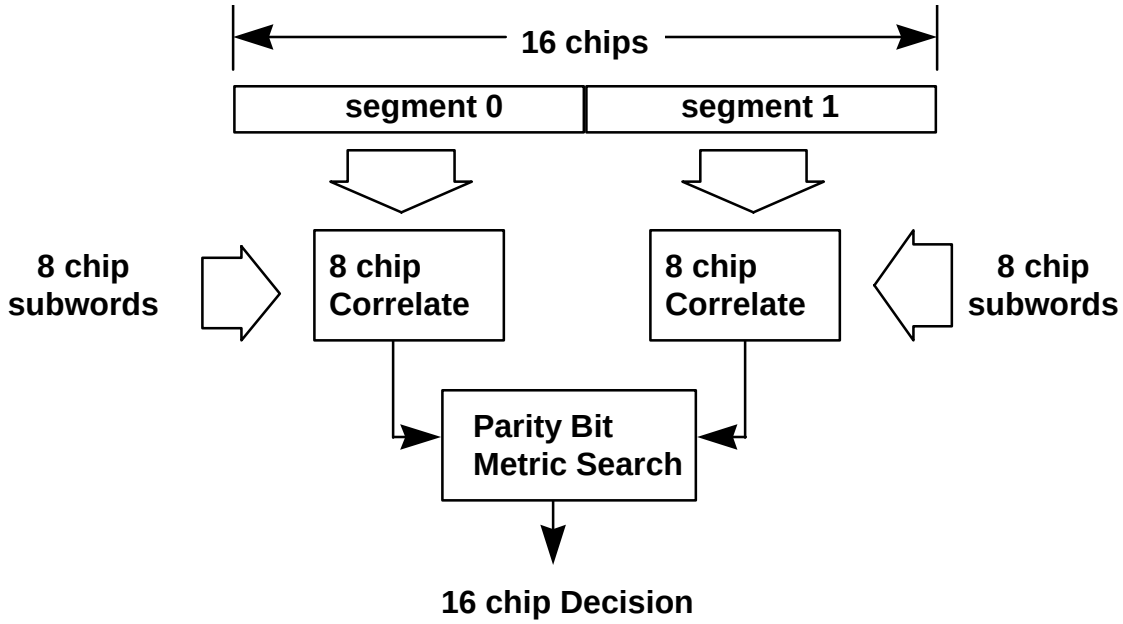

Figure A.7 Optimal decoding architecture.

How can one be certain that after discarding all the first-half/second-half soft decisions except the largest even-bit-count soft-decision and the largest odd-bit-count soft-decision, the optimal 16 chip decision is made? The reason is simple. The largest soft-decision for the complete 16 chips must result from one of 4 possible combinations (additions): odd-bit/odd-bit, odd-bit/even-bit, even-bit/odd-bit or even-bit/even-bit. This is an exhaustive set. Since the largest soft decisions were retained from both halves, the largest combination is one of the four. The parity-bit encoding across 16 chips makes the result unique and identifiable from the two simple sets.

COMPLEXITY CONTRAST

At the highest data rate, with 15 information bits encoded across the 16 chips, the receiver needs to use 2^15 = 32768 codeword correlators. With the sign bits decoded separately, 2^13 = 8192 codeword correlators are needed, followed by the sign decision. This is a huge, impractical number.

However, by employing the recommended underlying codeword structure, the number of correlators is greatly reduced. The number of correlators needed for a single half is 2^6 = 64, with 2 bits sign decoded separately. To decode the full 16 chips, only 64 + 64 = 128 correlations need be computed, rather than 8192. This is an additive increase in complexity, rather than multiplicative.

This codeword structure is novel and optimal.

Unmasking the "Parity-Bit Encode The Whole Packet" Trick

Some may argue that this 8-chip concatenation to form 16 chip symbols using parity-bit binding is completely arbitrary. The potent criticism follows this line of thought: "Why not just add a single parity bit to the whole packet? Then the whole packet can be called a symbol with hundreds of chips."

If one were to parity-bit concatenate the whole packet, how would one optimally process the packet? Extracting all the information? A practical implementation is not possible. Nor would real performance enhancement be realised.

The 16 chip scheme described in this memo not only describes the encoding element, the optimal-and-practical decoding (detector) has been also described which fully extracts all the information in the transmitted 16 chip symbols. Consequently, both Japan and the FCC should feel comfortable that this is not a ruse, but a potent worthwhile scheme.