

**IEEE P802.11
Wireless LANs**

**HARRIS/LUCENT CCK DESCRIPTION:
Additional Covercode and Fast Transform Detail**

Date: Sept. 15, 1998

Author: Mark Webster and Carl Andren
Harris Semiconductor, M/S 62A-028
P.O. Box 883
Melbourne, Fl. 32902-0883
Phone: 407-724-7537
Fax: 407-724-7094
e-Mail: candren@harris.com

Abstract

This submission clarifies the data encoding/decoding aspects of the HARRIS/LUCENT CCK proposal.

1. Introduction

This document takes a backward look at HARRIS/LUCENT's July presentation. The presentations in July were somewhat terse, so this document attempts to clearly define the lower-level details of signal encoding/decoding.

The HARRIS/LUCENT proposal has the following features:

- Codewords defined as 8 chip CCK. 256 CCK codewords are used at 11 Mbps. 16 CCK codewords are used at 5.5 Mbps.
- The codewords have properties which enable high-performance RAKE receivers.
- At 11Mbps, 8 information bits per codeword are used in the following fashion: 6 information bits select one-of-64 subcodes, while 2 information bits select one-of-4 DQPSK phases.
- At 5.5 Mbps, 4 information bits per codeword are used in the following fashion: 2 information bits select one-of-4 subcodes, while 2 information bits select one-of-4 DQPSK phases.
- Differential-QPSK phase encoding across successive codewords is used to enable noncoherent carrier tracking loops.
- Codeword correlation can be broken into two steps: (1) noncoherently correlate for the 64 subcodes at 11 Mbps. (2) Coherently detect the codeword phase.
- Fast transform only required on the 64 subcodes.
- At 5.5 Mbps, the 4 codewords are a subset of the 64 subcodes. This allows the same fast transform to be used for both 5.5 and 11 Mbps, if desired.
- A covercode can be easily applied to the 64 subcodes in the transmitter. The covercode can be easily removed in the receiver.
- The covercode is independent of the codeword.
- The covercode still allows a fast transform on 64 subcodes.
- The covercode is merely a rotation of the codeword's QPSK chips. Each chip is rotated by one of the following: 0, 90, 180 or 270 degrees.
- The covercode is stripped in the receiver merely by derotating the codeword's samples output from the channel matched filter.
- 3 architectures possible: RAKE-only, RAKE-with ISI equalisation, RAKE with ISI and ICI equalisation.
- RAKE-only is minimally defined as only a channel matched filter (CMF) followed by a codeword correlator (fast transform with phase detect).
- The fast transform uses a positive rotation on all the butterflies.

2. Rake Receiver With Programmable Covercodes

HARRIS/LUCENT have proposed a signalling scheme which possesses a receiver architecture which is simple to implement. In fact the architecture can also be made covercode programmable without measurable complexity increase.

A covercode programmable modulator is illustrated in Figure 1. The corresponding covercode programmable demodulator is shown in Figure 2.

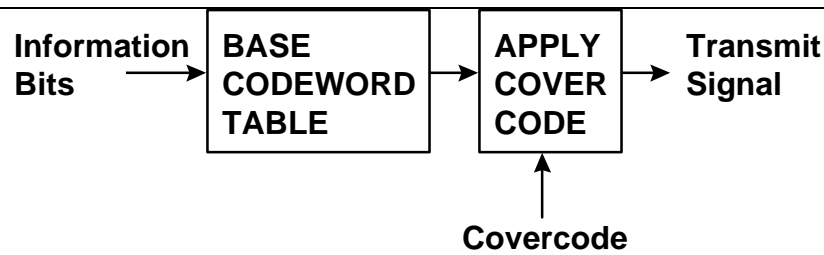


Figure 1 Applying a simple covercode in the transmitter.

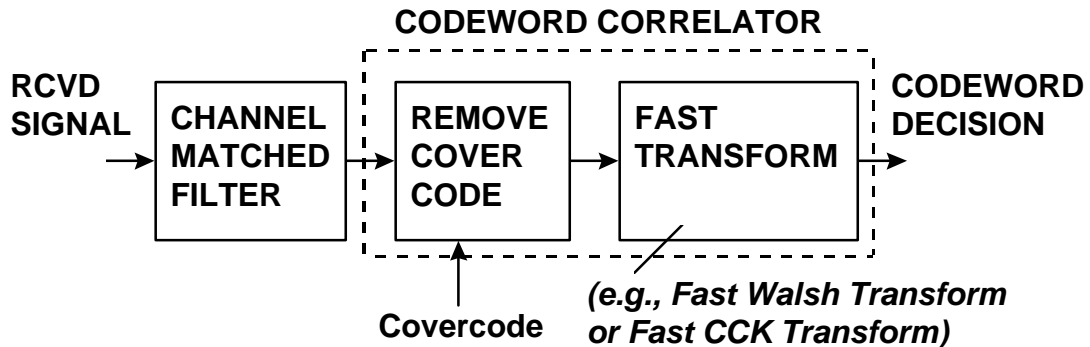


Figure 2 The preferred RAKE receiver architecture embraced by HARRIS/LUCENT.

The RAKE receiver can assume many different forms, and Figure 2 shows only one of them. The RAKE uses a correlator and channel-finger combiner. When the codewords possess good autocorrelation and cross-correlation properties the relative distances between the codewords at the channel output tend to become independent of the channel. The channel matched filter (CMF) coherently combines the channel fingers (discrete reflections with corresponding to time-of-arrivals, amplitude-and-phase of arrivals). Since the operations are linear, the CMF can be placed on either side of the codeword correlator.

It is convenient to use codewords which can be detected with a fast transform. Common fast transforms are the fast Fourier transform (FFT) and the fast Walsh transform (FWT). Unfortunately, codewords which possess a fast transform do not necessarily possess the good correlation properties required for high-performance RAKE reception.

3. What-Are and Why-Use Covercodes?

To generate codewords with both good correlation properties and a fast transform, covercodes are used.

In their simplest form, covercodes are an extra phase modulation of the base codeword's chips (e.g., Walsh codes), and the covercode is common to all the codewords. The transmitter places the covercode on all codewords before transmission as shown in Figure 1. Since the modulation is common to all the codewords, the receiver can unambiguously remove the covercode to obtain the base-codewords (e.g., Walsh codes). Once the covercode is removed, the fast transform can be run (e.g., fast Walsh transform).

The use of covercodes allows one to find codewords with good autocorrelation and cross-correlation properties by greatly increasing the size of the codeword space which is compatible with the basecode structure. The covercode is a simply transform on the basecodes applied in the modulator and removed in the receiver.

An example is now presented with a simple 8 chip, 8 codeword Walsh codes. The 8 chip Walsh codes have the values shown in Figure 3. A simple Fast Walsh Transform can be used to decode this set. Unfortunately, this is the one and only 8 chip Walsh code set, and its RAKE receiver properties are poor.

Fortunately, the set can be made much larger by using covercodes. For a binary covercode of $\{+1,-1\}$, $2^8 = 256$ different sets are generated (not all of them are unique. Now 256 different 8 chip, 8 word sets can be examined for the best when used with a RAKE receiver.

1	1	1	1	1	1	1	1
1	-1	1	-1	1	-1	1	-1
1	1	-1	-1	1	1	-1	-1
1	-1	-1	1	1	-1	-1	1
1	1	1	1	-1	-1	-1	-1
1	-1	1	-1	-1	1	-1	1
1	1	-1	-1	-1	-1	1	1
1	-1	-1	1	-1	1	1	-1

Figure 3 The Walsh code set for 8 chips.

HARRIS/LUCENT used proposed a covercode technique which provide $4^8 = 65536$ different 8-codeword sets (not all unique) to be generated from the base CCK, generalized Hadamard form. Since this set is so large, HARRIS/LUCENT have not examined all possible covercodes to determine we have selected the best.

4. HARRIS/LUCENT's CCK Covercode

This section defines the concept HARRIS/LUCENT had in mind when describing covercodes for the proposed CCK.

HARRIS/LUCENT have found a codeword set which possesses both good correlation properties and a fast transform. The codewords are from a class of signals called complementary code keying (CCK). These codes have a generalized Hadamard structure, so they possess a fast transform. Conventional Hadamard codes are formed from binary (2-ary) elements (chips). For communication systems, Hadamard codes often use ± 1 signaling levels (BPSK). Generalized Hadamard codes possess a higher-level M-ary structure. For the 4-ary case, the codes can be

implemented with QPSK chips. HARRIS/LUCENT proposed 8 chip codewords use 4-ary QPSK chips.

The HARRIS/LUCENT CCK codes are synthesized from the set $\{e^{j(\phi_1+\phi_2+\phi_3+\phi_4)}, e^{j(\phi_1+\phi_3+\phi_4)}, e^{j(\phi_1+\phi_2+\phi_4)}, -e^{j(\phi_1+\phi_4)}, e^{j(\phi_1+\phi_2+\phi_3)}, e^{j(\phi_1+\phi_3)}, -e^{j(\phi_1+\phi_2)}, e^{j(\phi_1)}\}$, where ϕ_1, ϕ_2, ϕ_3 and ϕ_4 are taken from the 4-ary set $\{0, \pi/2, \pi, 3\pi/2\}$. ϕ_1, ϕ_2, ϕ_3 and ϕ_4 correspond to the information bits, where 2 information bits are mapped to each phase.

The codewords possess a structure which enables fast synthesis/analysis transforms to be used. Assume the codeword's chips are numbered starting with 1. Then, the Hadamard structure is created by placing ϕ_1 on all the chips, ϕ_2 on all the odd chips, ϕ_3 on all the odd chip-pairs and ϕ_4 on all odd chip-quads. Later, it will be shown how this enables a fast transform.

There are a total of $4^4=256$ codewords in this 8-chip set for 11 Mbps.

The covercode can be identified as $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$ for CCK codes $\{c_1 e^{j(\phi_1+\phi_2+\phi_3+\phi_4)}, c_2 e^{j(\phi_1+\phi_3+\phi_4)}, c_3 e^{j(\phi_1+\phi_2+\phi_4)}, c_4 e^{j(\phi_1+\phi_4)}, c_5 e^{j(\phi_1+\phi_2+\phi_3)}, c_6 e^{j(\phi_1+\phi_3)}, c_7 e^{j(\phi_1+\phi_2)}, c_8 e^{j(\phi_1)}\}$. The covercode used by HARRIS/LUCENT is $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\} = \{1, 1, 1, -1, 1, 1, -1, 1\}$.

HARRIS/LUCENT have not searched all possible covercodes for the CCK set. It is possible that there is a superior covercode to the one we have selected $\{1, 1, 1, -1, 1, 1, -1, 1\}$. HARRIS/LUCENT is open to using a superior covercode if one is found.

Since the CCK codewords are QPSK based, implementation is easiest if the covercodes are formed from the complex values $\{1, j, -1, -j\}$.

An example of what HARRIS/LUCENT had in mind for allowed covercode is $\{1, j, -j, -1, j, 1, -1, -j\}$, where the QPSK chips are merely rotated by the covercode. In the transmitter, the base codewords are multiplied by the covercode. In the receiver, the codeword is multiplied by the complex-conjugate of the covercode. All the multiplications can be implemented merely by exchanging inphase and quadrature chip values (rotations). The rotations are made in one direction in the transmitter and in the opposite direction in the receiver. No multiplications are actually needed.

Making this type of covercode $\{1, j, -1, -j\}$ programmable is easy because only 2 bits of covercode information per chip are needed. With 8-chip CCK only a total of 16 bits are needed to specify the covercode. The number of gates required to implement the programmable covercode is small in both the modulator and demodulator.

HARRIS/LUCENT stated they were open to covercodes of this type. The *APPLY COVERCODE* block of Figure 1 is illustrated in Figure 4. The *REMOVE COVERCODE* block of is illustrated in Figure 5. In these figures the parameters cc1 through cc8 are 4-ary $\{0, 1, 2, 3\}$.

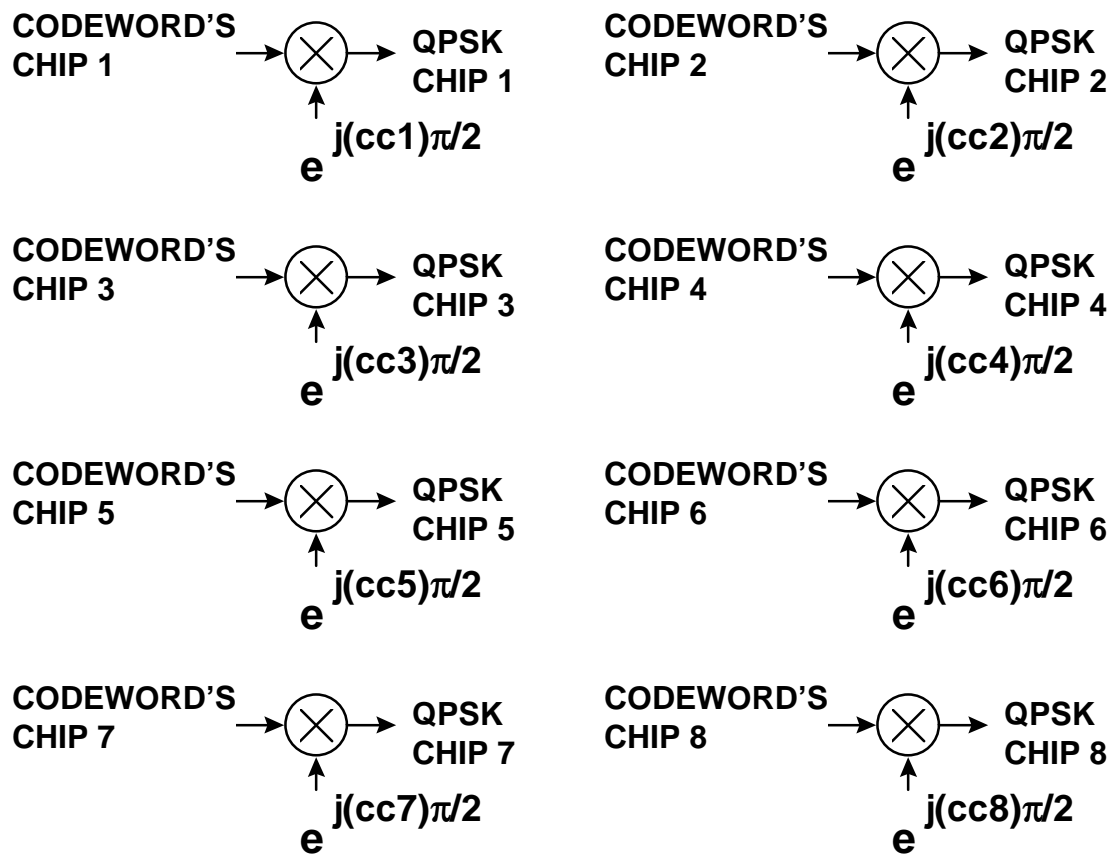


Figure 4 Covercode application show as a pure rotation of the QPSK chips of the codeword.

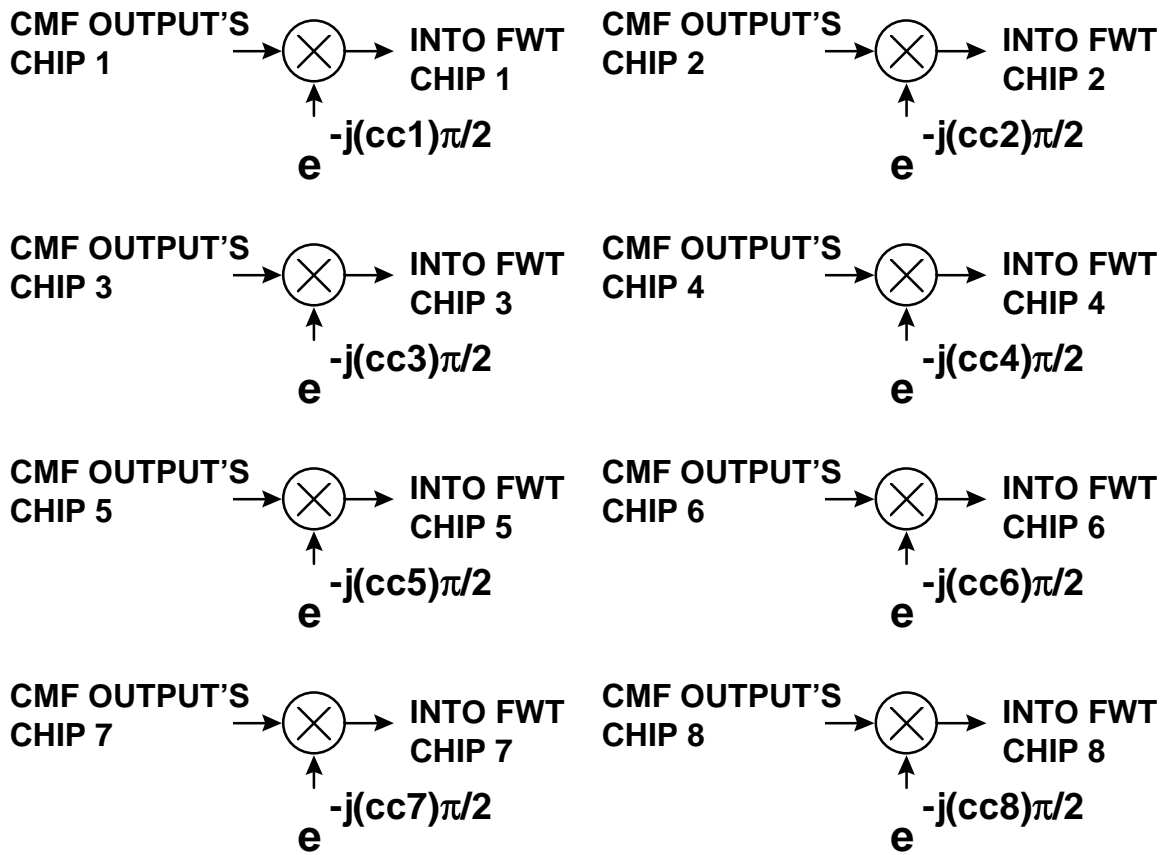


Figure 5 Covercode removal show as a pure derotation of the samples output by the channel matched filter (CMF).

5. Subcode Correlations

The HARRIS/LUCENT codewords allow the receiver to correlate over only a one-quarter subset, greatly simplifying implementation. This is shown in Figure 6.

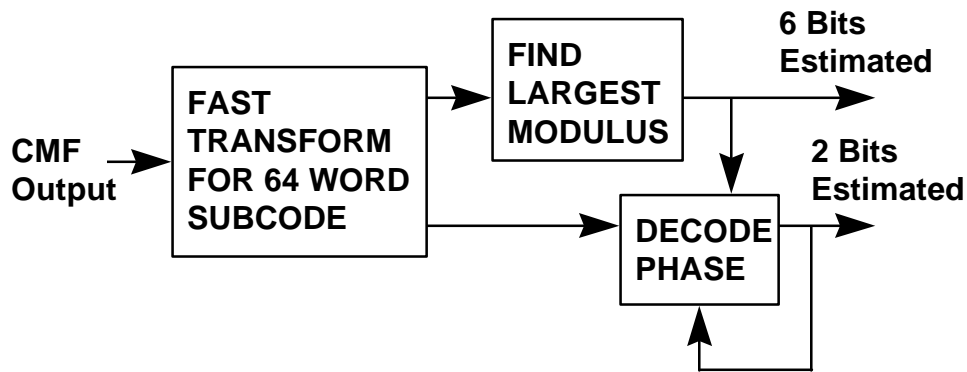


Figure 6 Subcode correlation simplifies the receive processing.

At 11 Mbps per second there are 256 codewords according to the CCK encoding formula $\{c_1 e^{j(\phi_1+\phi_2+\phi_3+\phi_4)}, c_2 e^{j(\phi_1+\phi_3+\phi_4)}, c_3 e^{j(\phi_1+\phi_2+\phi_4)}, c_4 e^{j(\phi_1+\phi_4)}, c_5 e^{j(\phi_1+\phi_2+\phi_3)}, c_6 e^{j(\phi_1+\phi_3)}, c_7 e^{j(\phi_1+\phi_2)}, c_8 e^{j(\phi_1)}\}$, where c_1 through c_8 define the covercode. However the receiver need only correlate for 64 subcodes. These subcodes are obtained by removing the covercode and removing the common term $e^{j\phi_1}$ to give $\{e^{j(\phi_2+\phi_3+\phi_4)}, e^{j(\phi_3+\phi_4)}, e^{j(\phi_2+\phi_4)}, e^{j(\phi_4)}, e^{j(\phi_2+\phi_3)}, e^{j(\phi_3)}, e^{j(\phi_2)}, 1\}$. The 64 subcodes are established by the 4-ary terms ϕ_2, ϕ_3 and ϕ_4 .

The fast transform is only needed on the 64 subcodes, rather than the full 256 codewords. The transform output with the largest magnitude identifies the transmitted 1-of-64 subcodes. The phase corresponding to that largest output identifies the DQPSK value $e^{j\phi_1}$, resolving the last 4-ary information element.

6. Differential- Phase Encoding of Codewords

It is preferable to define the high-rate IEEE 802.11 standard with a transmission scheme that uses differential-phase encoding. With differential-phase encoding the receiver need not use a PLL if preferred. HARRIS/LUCENT's proposal defines DQPSK encoding across successive codewords. This is shown in Figure 7.

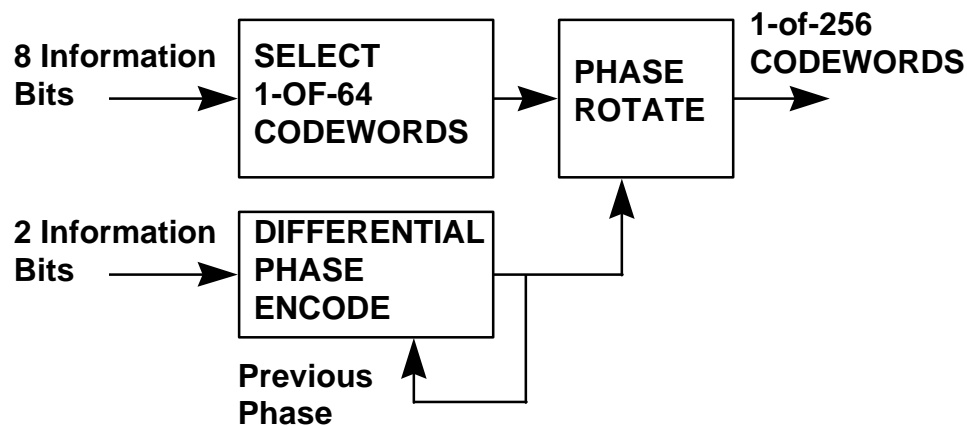


Figure 7 Differential-QPSK phase encoding of the 11 Mbps signal. The same encoding is used at 5.5 Mbps.

This is harmonious with the current 1 and 2 Mbps DSSS waveforms, where the 11-chip Barker codeword carries information using phase differences between successive codewords. This feature allows one to use noncoherent carrier-tracking techniques in the receiver, if desired. The drawback is there is an increase in bit-error-rate due to the inherent memory. The 1 Mbps mode uses differential BPSK. The 2 Mbps mode uses differential QPSK.

Differential phase encoding of the CCK is the recommended approach. To provide the same feature with CCK, one can write the CCK codewords as $e^{j(\phi_1)}\{c_1 e^{j(\phi_2+\phi_3+\phi_4)}, c_2 e^{j(\phi_3+\phi_4)}, c_3 e^{j(\phi_2+\phi_4)}, c_4 e^{j(\phi_4)}, c_5 e^{j(\phi_2+\phi_3)}, c_6 e^{j(\phi_3)}, c_7 e^{j(\phi_2)}, c_8\}$, where the ϕ_1 term has been pulled-out in front. Since ϕ_1 is common to all the 8 CCK chips, it can be used to specify the codeword phase. ϕ_1 can be defined differentially between codewords.

There are now 64 codewords prior to the addition of the codeword phase. The addition of the codeword phase creates 256 codewords.

7. 5.5 Mbps Definition

At 5.5 Mbps, the 4 codewords are a subset of the 64 subcodes. This allows the same fast transform to be used for both 5.5 and 11 Mbps. At 5.5 Mbps only four outputs for the 64 word fast transform would be examined. Alternatively, a pruned version of the 64 word transform could be used at 5.5 Mbps.

The 5.5 Mbps codewords specified by HARRIS/LUCENT are formed by encoding ϕ_1 as in the 11 Mbps case, setting ϕ_3 to zero, letting ϕ_2 take on only the two values $\{\pi/2, 3\pi/2\}$ and ϕ_4 take on only 2-ary values $\{0, \pi\}$. Since both ϕ_1 and ϕ_2 take on two values from the set $\{0, \pi/2, \pi, 3\pi/2\}$, these codewords are a subset of those used for 11 Mbps. This reuse minimizes implementation complexity.

8. HARRIS/LUCENT Codeword Encoder Forms

This section examines various codeword encoding schemes which implement the HARRIS/LUCENT proposal. One of the 64 subcodes is selected, then the covercode and differential phase term is applied. The last implementation was suggested by ALANTRO in a private communication.

8.1 CODEWORD-PHASE AND COVERCODE COMPLEX

This implementation most closely matches what HARRIS/LUCENT had in mind during earlier presentations. It is illustrated in Figure 8. The 4-ary values a_1, a_2, a_3 and a_4 take on one of the values $\{0, 1, 2, 3\}$. The terms a_2, a_3 and a_4 select one of 64 subcodes to generate 8 complex QPSK chips. In practice, these would be probably be generated via a table look-up. The differential phase term a_1 is applied directly to the complex chips. The covercode is applied by rotating the chips.

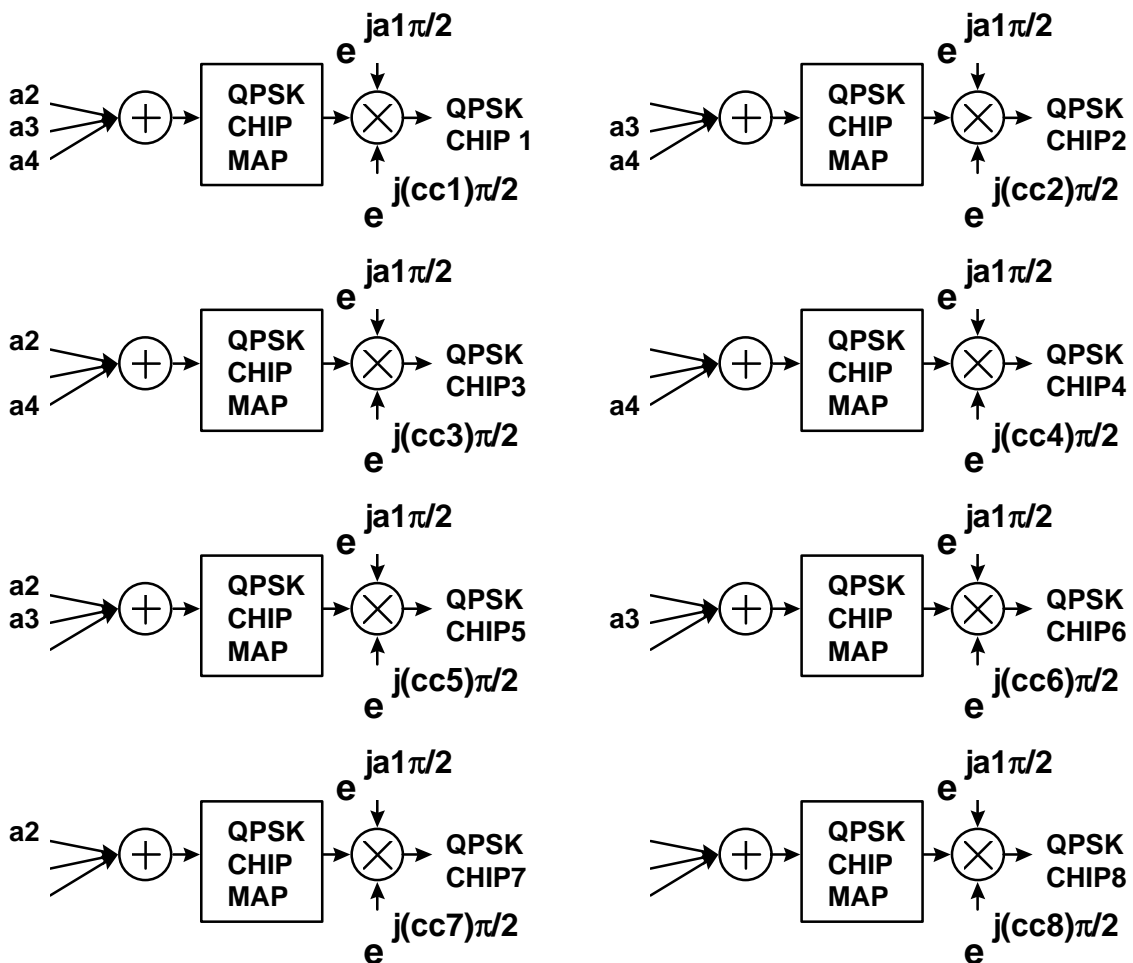


Figure 8 Implementation showing the covercode as a rotation of the QPSK chip. Since the covercode is 4-ary, the rotation is from the set { 0, 90, 180, 270} degrees.

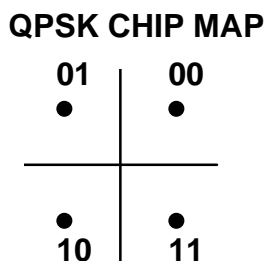


Figure 9 The QPSK chip map used in Figure 8.

8.2 PRE-CHIP MAP ENCODING

In the implementation of Figure 10, all the values sit before the QPSK chip map. The chip map in Figure 9 is still used.

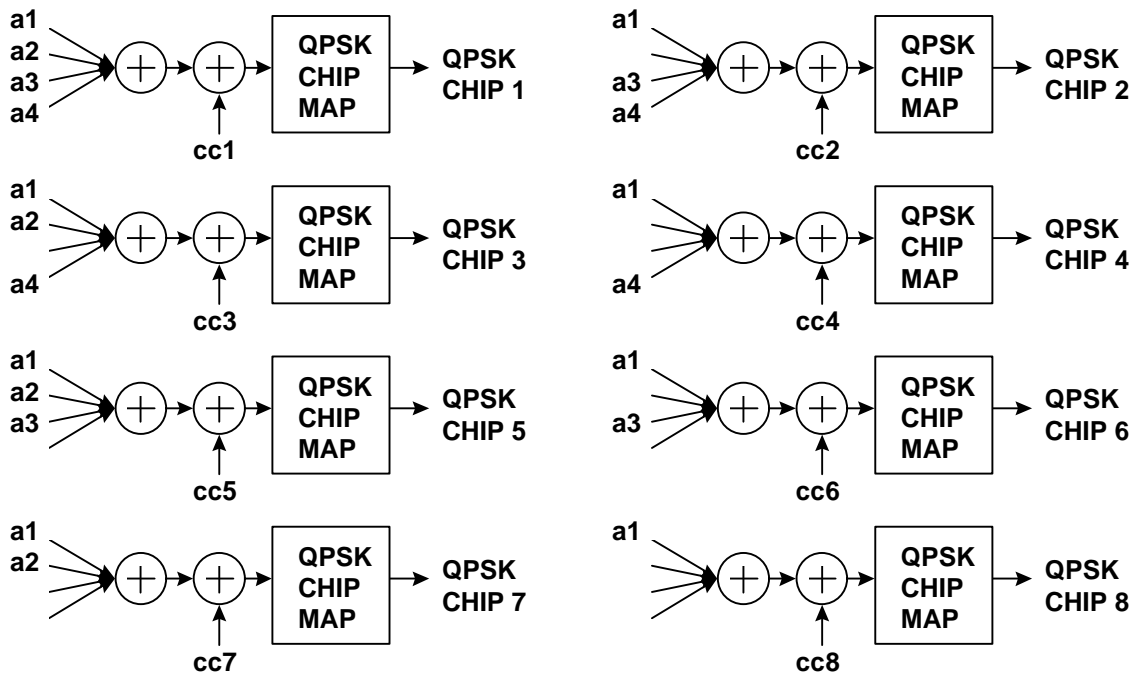


Figure 10 The eight CCK chip definitions formulated using four 4-ary information elements {a1, a2, a3, a4} and a 4-ary cover specified for each chip. The QPSK chip map is shown in Fig. 2-2.

8.3 USING MULTIPLE CHIP MAPS

This particular implementation shown in Figure 11 was derived using insights provided by ALANTRO. Here the covercode variable for each chip is used to select a particular chip map. The chip maps are shown in Figure 12. The chip map 1 is a 90 degree rotation of chip map 0. Chip map 2 is a 180 degree rotation of chip map 0. Chip map 3 is a 270 degree rotation of chip map 0.

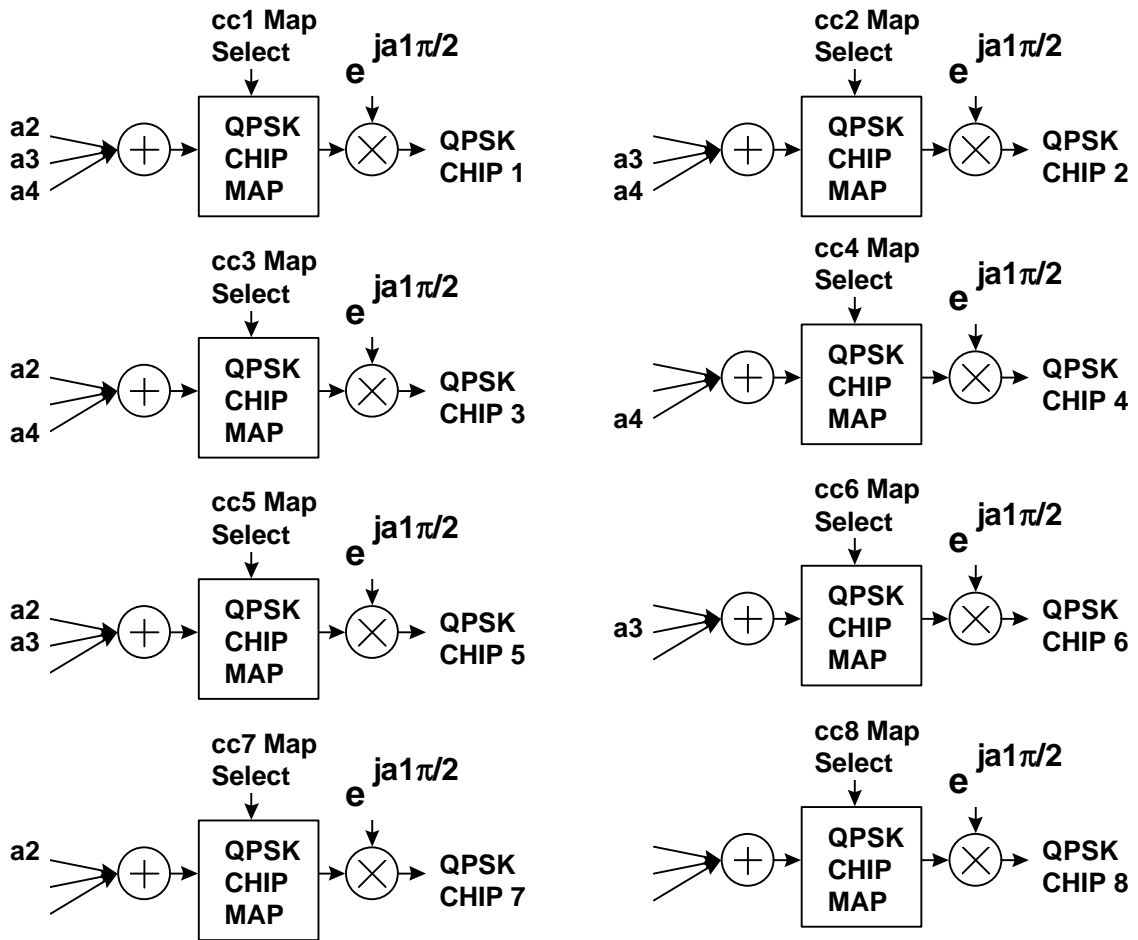


Figure 11 Using the covercode to select different QPSK chip maps.

QPSK CHIP MAPS

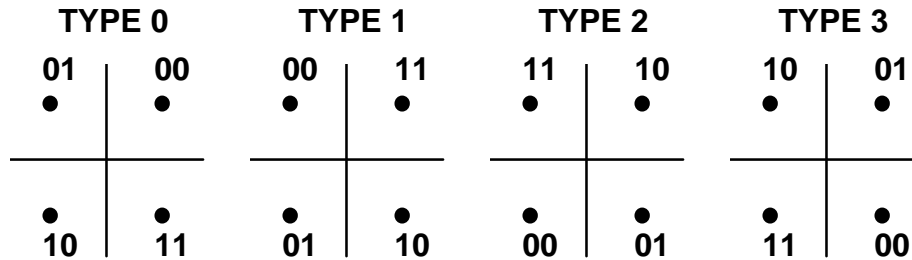


Figure 12 The chip maps which would be used in the HARRIS/LUCENT definition. Type 1 is a 90 degree rotation of Type 0. Type 2 is a 180 degree rotation of Type 0. Type 3 is a 270 degree rotation of Type 0.

9. FAST TRANSFORM STRUCTURES

9.1 FAST TRANSFORM FOR HARRIS/LUCENT's CCK

This section develops the architecture proposed by HARRIS/LUCENT for taking a fast transform of the 64 subcodes.

The codeword structure opens the door to a fast transform. CCK codes are created by ϕ_1 encoding across all 8 samples. Assume the chips are numbered with the first chip having number one. ϕ_2 is encoded across all odd chips. ϕ_3 is encoded across all odd pairs of chips. ϕ_4 is encoded across all odd quads of chips. This structure is clearly evident in Figure 13.

$$\{ e^{j(\phi_1+\phi_2+\phi_3+\phi_4)}, e^{j(\phi_1+\phi_3+\phi_4)}, e^{j(\phi_1+\phi_2+\phi_4)}, e^{j(\phi_1+\phi_4)}, e^{j(\phi_1+\phi_2+\phi_3)}, e^{j(\phi_1+\phi_3)}, e^{j(\phi_1+\phi_2)}, e^{j(\phi_1)} \}$$

Figure 13 The CCK codeword without a covercode.

The fast transform can be developed through a series of factorizations. Figure 14 shows the codewords with ϕ_1 factored out. This step defines the 64 subcodes. Figure 15 shows ϕ_4 factored out of the first quad. Figure 16 shows ϕ_3 factored out of the odd pairs. ϕ_2 already is now factored out of the odd chips.

$$e^{j(\phi_1)} \{ e^{j(\phi_2+\phi_3+\phi_4)}, e^{j(\phi_3+\phi_4)}, e^{j(\phi_2+\phi_4)}, e^{j(\phi_4)}, e^{j(\phi_2+\phi_3)}, e^{j(\phi_3)}, e^{j(\phi_2)}, 1 \}$$

Figure 14 The CCK codes with ϕ_1 factored out.

$$e^{j(\phi_1)} \{ e^{j(\phi_4)} \{ e^{j(\phi_2+\phi_3)}, e^{j(\phi_3)}, e^{j(\phi_2)}, 1 \}, e^{j(\phi_2+\phi_3)}, e^{j(\phi_3)}, e^{j(\phi_2)}, 1 \}$$

Figure 15 The CCK codes with ϕ_1 and ϕ_4 factored out.

$$e^{j(\phi_1)} \{ e^{j(\phi_4)} \{ e^{j(\phi_3)}, e^{j(\phi_2)}, 1 \}, \{ e^{j(\phi_2)}, 1 \} \}, \{ e^{j(\phi_3)}, e^{j(\phi_2)}, 1 \}, \{ e^{j(\phi_2)}, 1 \} \}$$

Figure 16 The CCK codes with ϕ_1 , ϕ_3 and ϕ_4 factored out.

A fast correlation structure can be defined for each of the factorisations. Each pair of braces and center comma $\{\bullet,\bullet\}$ define the elements over which the correlation iterates. The correlation's butterfly structure is shown in Figure 17. There are 2 input elements and 4 output elements for the butterfly. An input element consists of a set of correlations from a preceding transform stage.

ϕ_1 is a special term. Since it resides on all chips a butterfly is not needed for computation.

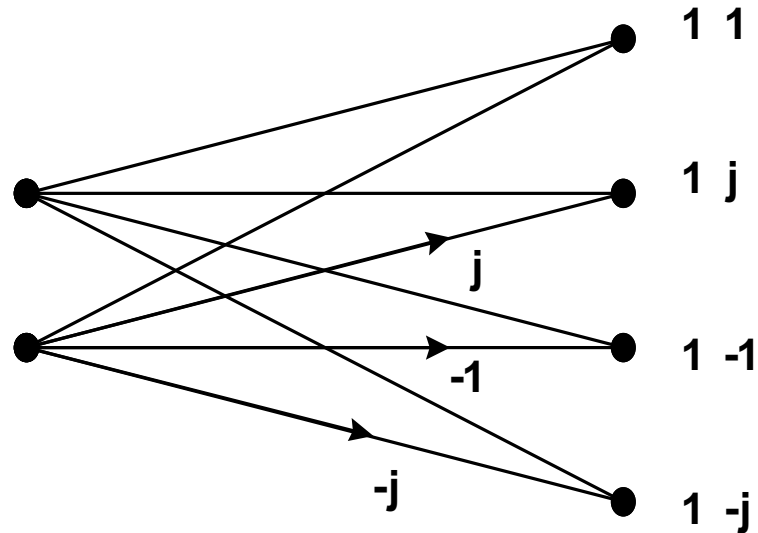


Figure 17 The basic Butterfly structure used for CCK with QPSK chips. On the right are listed the corresponding correlation weights for the top and the bottom inputs. This butterfly is used in a 2-chip, 4-chip and 8-chip form as explained below.

The butterfly which is applied to the ϕ_2 terms is shown in Figure 18. For pairs of received chips, ϕ_2 appears on every other sample. Since ϕ_2 can assume the values $\{0, \pi/2, \pi, 3\pi/2\}$, the correlation on every other sample must occur over the possible values 1, j , -1 , and $-j$.

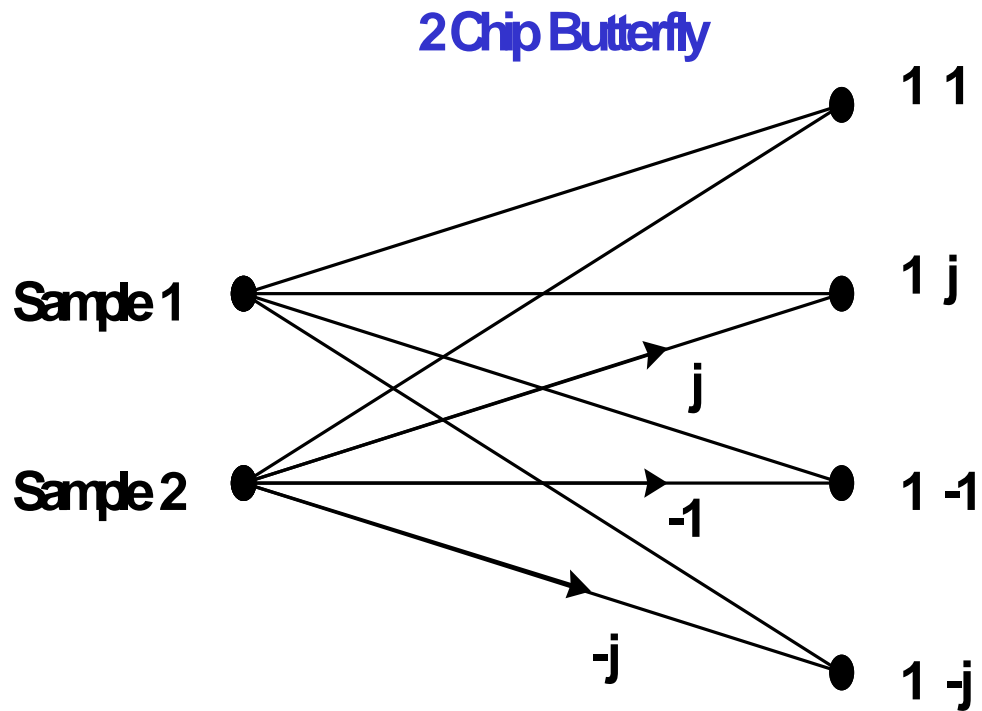


Figure 18 The butterfly used to process pairs of input samples.

The butterfly which is applied to the ϕ_3 terms is shown in Figure 19. For pairs of 2-chip butterfly outputs, ϕ_3 appears on every other sample. Since ϕ_3 can assume the values $\{0, \pi/2, \pi, 3\pi/2\}$, the correlation on every other sample must occur over the possible values 1, j , -1 , and $-j$.

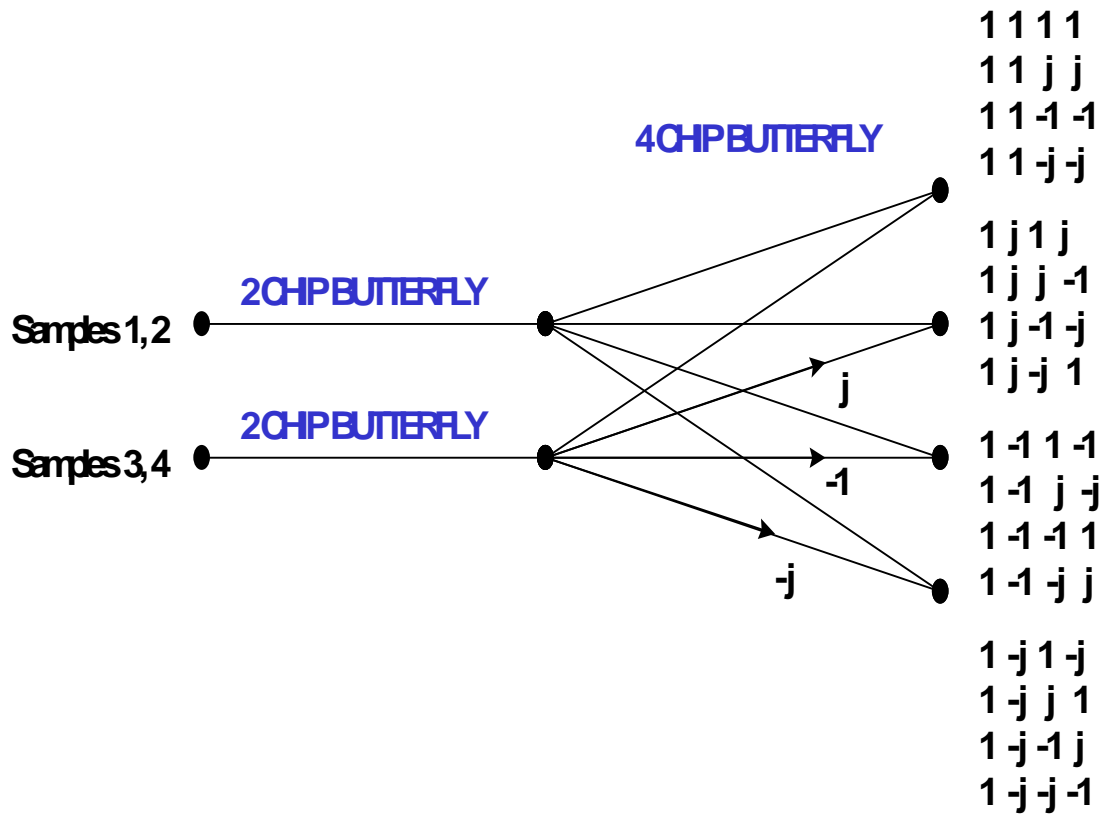


Figure 19 Transform structure for 4 received QPSK CCK chips.

The butterfly which is applied to the ϕ_4 terms is shown in Figure 20. For pairs of 2-chip butterfly outputs, ϕ_4 appears on every other sample. Since ϕ_4 can assume the values $\{0, \pi/2, \pi, 3\pi/2\}$, the correlation on every other sample must occur over the possible values 1, j, -1, and -j. This processing is hidden inside the 8 chip butterfly block.

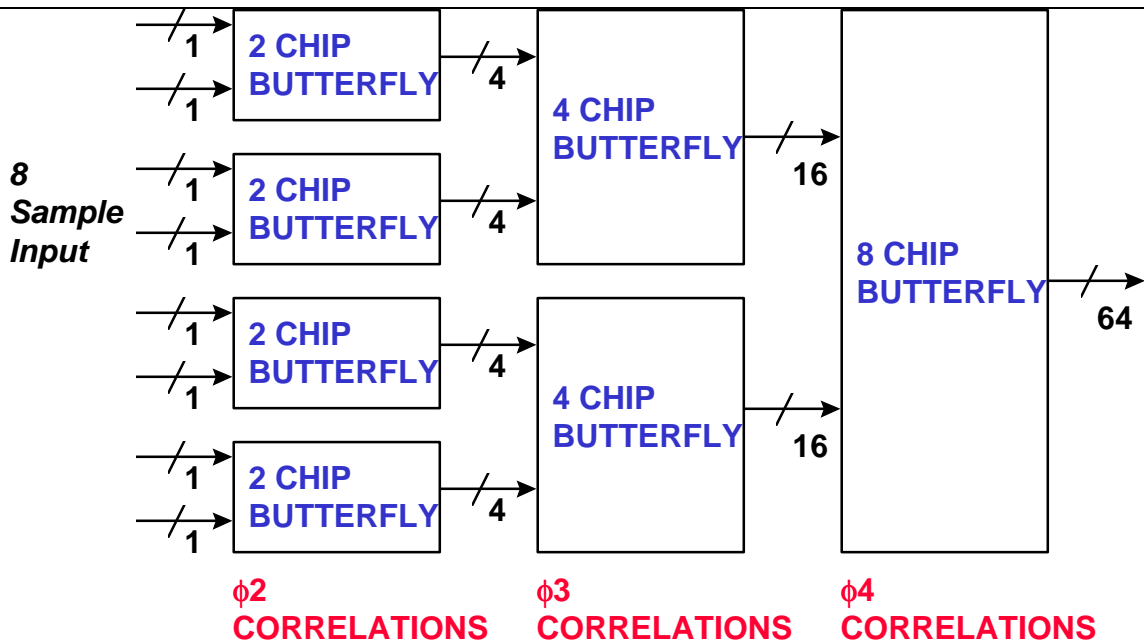


Figure 20 Complete fast-transform structure for 8 received QPSK CCK chips. The transform is used on 8 samples output from the channel matched filter.

9.2 FAST TRANSFORM CONSTRAINTS

One may ask the question, “What constraints are required to maintain a fast transform structure?” This section attempts to answer that question from the HARRIS/LUCENT understanding of the transform.

The following will show that the a fast transform exists for a larger code space than what HARRIS/LUCENT proposed. This however is not a recommendation to change the CCK defined in July. It is merely an illumination of degrees of freedom one could exploit if one wanted to do so. Designing a fast transform which is programmable over the full code space is probably undesirable, since this is one of most intensive processing elements in the receiver.

First, notice the covercode as defined above does not impact the transform, since the samples are rotated and derotated independent of the 64 subcodes. Hence, the constraints examined in this section focuses on the 64 subcodes themselves.

Given the decomposed codeword shown in Figure 16, it appears the codeword can be generalized to that shown in Figure 21. Each pair of braces and center comma {•,•} define a correlation butterfly for an instance of a odd, odd-pair or odd-quad. It appears each of these could be independently sign-specified.

$$e^{+/-j(\phi1)} \{ e^{+/-j(\phi4)} \{ e^{+/-j(\phi3)} \{ e^{+/-j(\phi2)}, 1 \}, \{ e^{+/-j(\phi2)}, 1 \} \}, \{ e^{+/-j(\phi3)} \{ e^{+/-j(\phi2)}, 1 \}, \{ e^{+/-j(\phi2)}, 1 \} \} \}$$

Figure 21 The codeword structure showin a optional sign on each factored butterfly correlation term.

ϕ_1 is a special term. Since it resides on all chips a butterfly is not needed for computation. Since it can only assume the values $\{0, \pi/2, \pi, 3\pi/2\}$, the same codewords are derived for both a positive rotation and a negative rotation. Consequently, it makes no sense to make the sign a free parameter for ϕ_1 .

The butterfly corresponding to a positive rotation is shown in Figure 22. The butterfly corresponding to a negative rotation is shown in Figure 23.

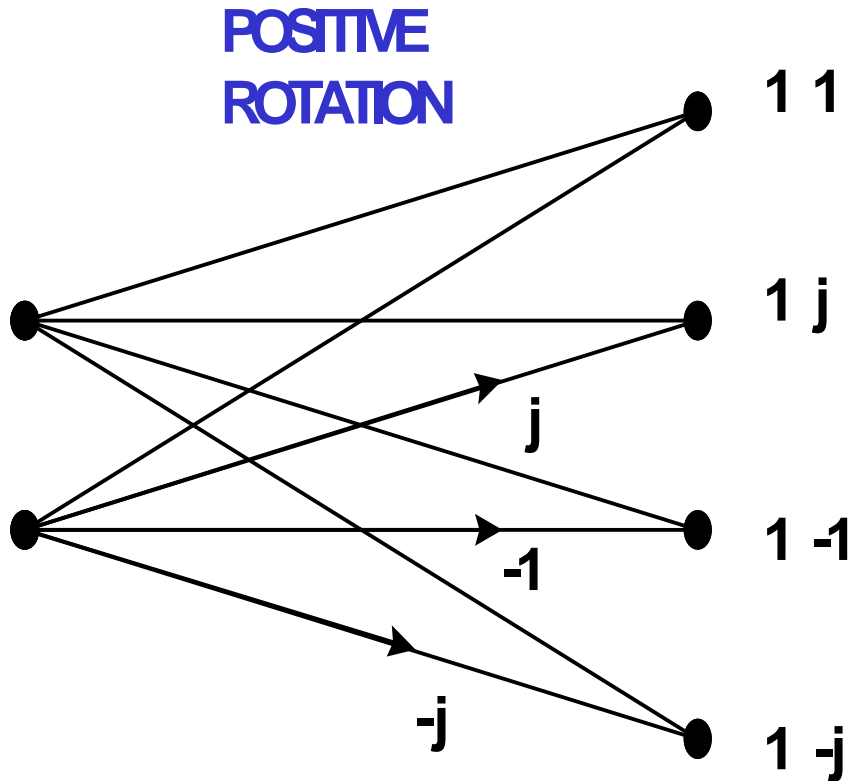


Figure 22 The butterfly structure associated with a positive phase rotation 0, 90, 180 and 270 degree for each 4-ary parameter.

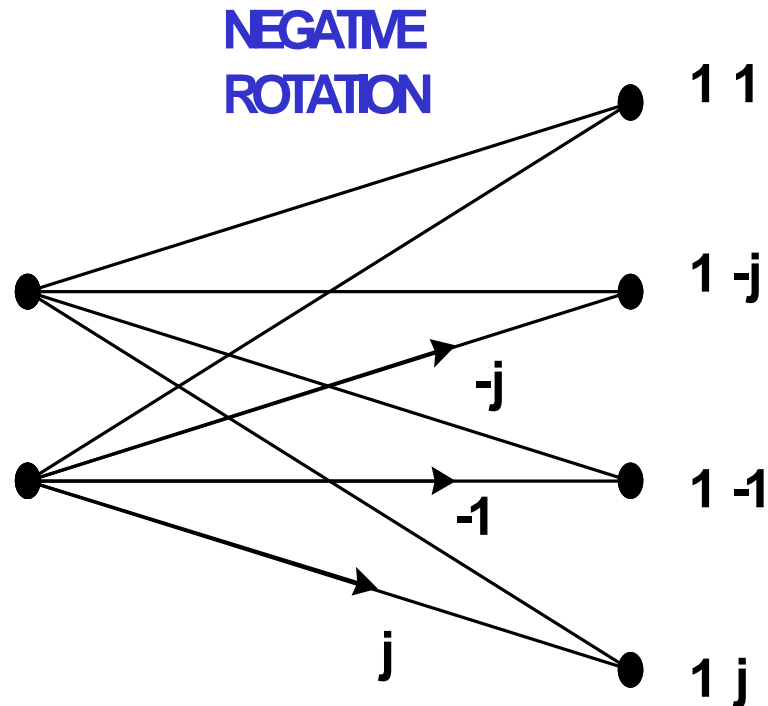


Figure 23 The butterfly structure associated with a negative phase rotation 0 , -90 , -180 and -270 degree for each 4-ary parameter relative to that shown in **Figure 22**.

The sign option on each butterfly is shown in Figure 24. This corresponds to 7 bits control, one bit for each butterfly identifying the rotation sense, positive or negative. Designing hardware be controllable does increase the complexity by a small amount. If a fixed specification of sign on each free parameter, the transform does not increase in complexity.

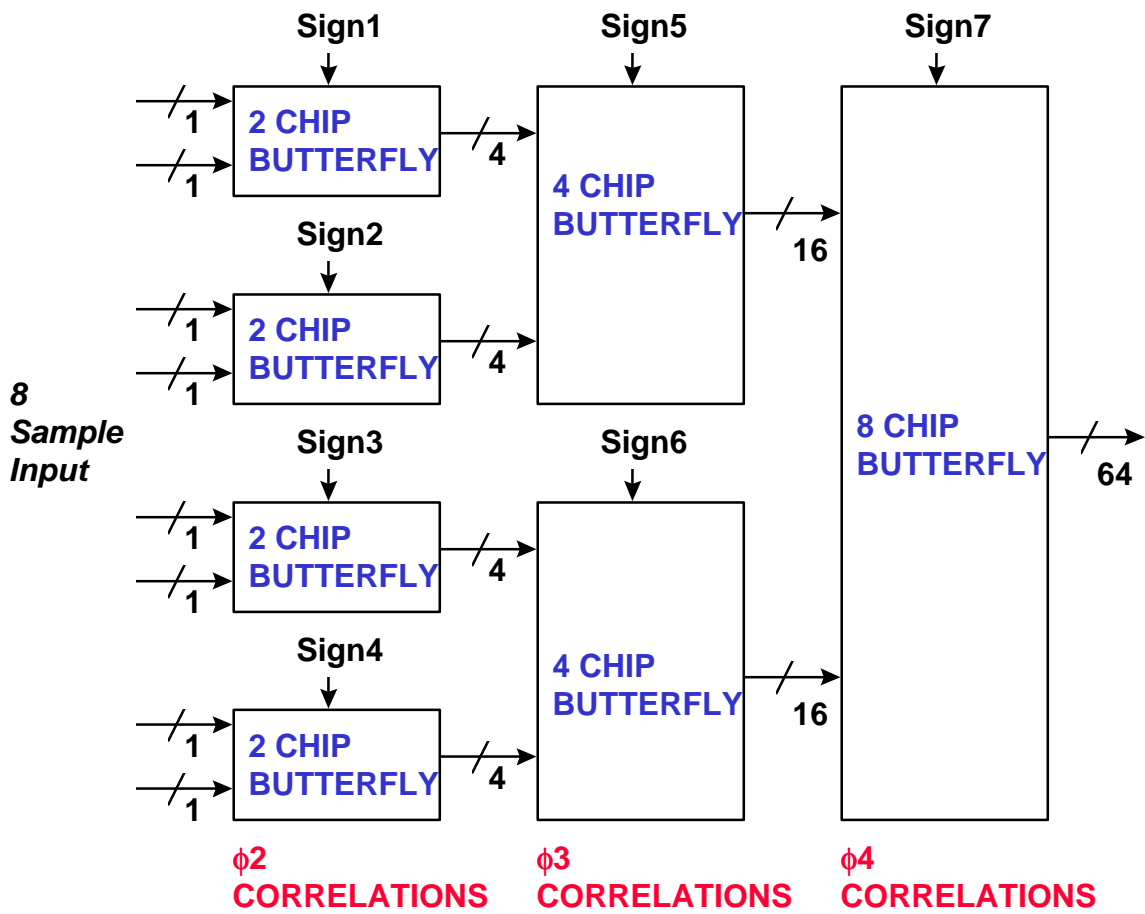


Figure 24 The fast transform shown with a control bit for each butterfly specifying the butterfly's rotation direction.

9.3 CODE SPACE INCREASE

Just as a covercode provides a increased code space, so doe the additional bits of specification described in Section 9.2. A much larger space exists to find codes with good autocorrelation and cross-correlation properties.

HARRIS/LUCENT's codes lie in a subspace of this larger set. It is not known if significantly superior codes lie in this increased space.

Making these parameters free for future discussion is probably too difficult, since designing a programmable, controllable transform is too expensive.

Searching this code space is difficult, consuming lots of computer time, since it is so large.