1 WiMax Convolutional Encoder

In the OFDMA mode of operation the use of the convolutional encoder is mandatory in the tail-biting mode, whereas its utilization in the zero-tailed mode is optional. This is a binary, constraint length 7, native code rate 1/2 convolutional encoder, whose generator sequences are defined as (octal representation):

$G_1 = 171_o$, $G_2 = 133_o$

Figure 1 shows the shift register diagram of the convolutional encoder in its native coding rate 1/2 form.

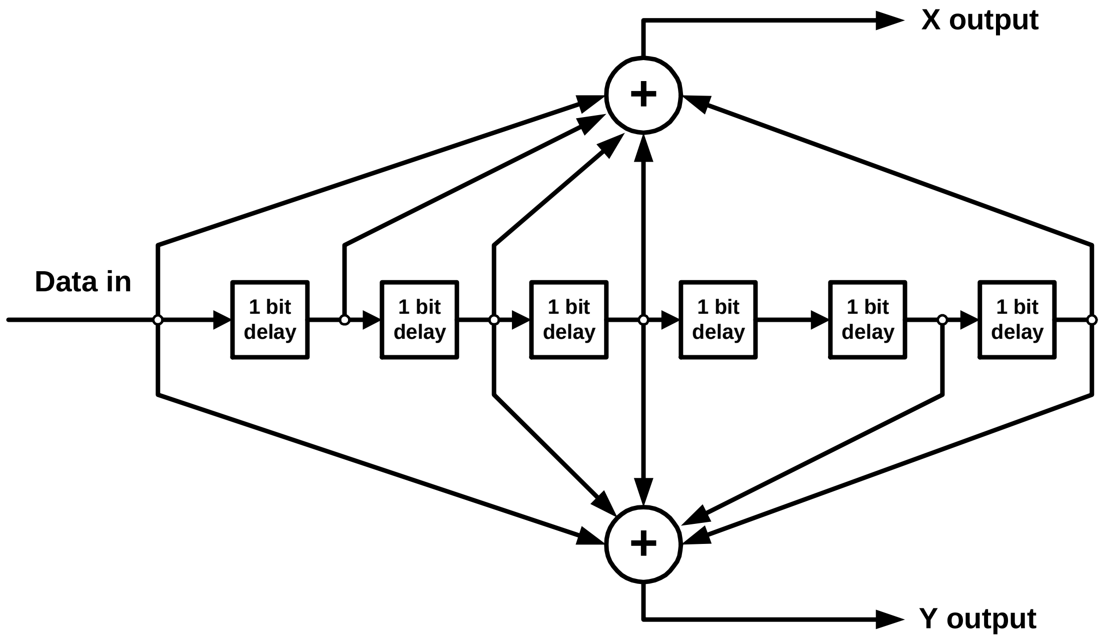

Figure 1. IEEE 802.16 rate 1/2 convolutional encoder

Higher coding rates are obtained by applying puncturing operation on the X and Y output pair. This operation removes some of the output bits following a pattern which depends on the coding rate (Figure 2).

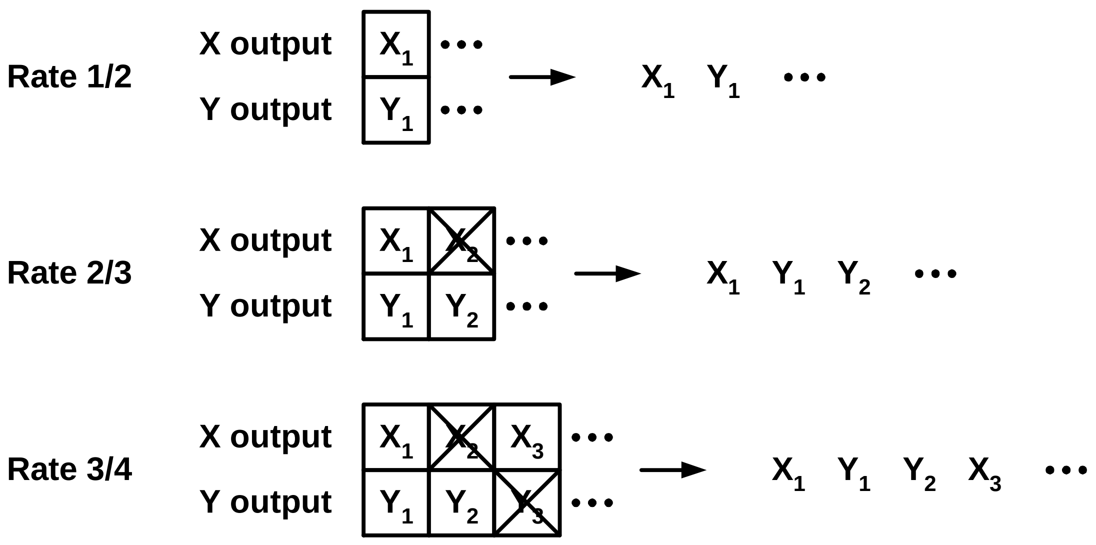

Figure 2. WiMax convolutional code puncturing patterns

For the mandatory tail-biting convolutional encoding (Clause 8.4.9.2.1 in [1]) valid combinations of the information block size, coding rate and the modulation type is shown in Table 1.

	QPSK		16QAM		64QAM		
Coding rate	**1/2**	**3/4**	**1/2**	**3/4**	**1/2**	**2/3**	**3/4**
Information data payload (bytes)	6						
		9					
	12		12				
	18	18		18	18		
	24		24			24	
		27					27
	30						
	36	36	36	36	36		

Table 1. Defined payload size/coding rate/modulation combinations

In this mandatory tail-biting mode the delay line of the convolutional encoder is initialized with the values of the last 6 info data bits.

In the case of the optional zero-tailed convolutional encoding (Clause 8.4.9.2.4 in [1]) valid values of the information block size are not specified. Rather, there is just a sentence, which reads "The convolutional code and the puncturing shall be applied to the whole burst without partitioning it into blocks." A naïve interpretation of this sentence would lead to conclusion that the support for the information block sizes corresponding to the maximum burst may be needed. In the uplink case example that would mean potentially dealing with information block sizes in excess of 40k bits. It is not likely, however, that one would use large blocks associated with convolutional code. Frame Error Rate (FER) performance would be poor in that case. Rather, one would expect that the zero-tailed encoding would be applied to the information blocks of the size similar to those in Table 1.

2 General considerations (apply to both VCP and VCP2)

The documents providing information on how to use the Viterbi co-processors are Reference Guides, [3] for VCP and [4] for VCP2. Reference to other useful supporting documents is also included in the last section of this document: [5], [6], [7], [8], [9], [10]. By following descriptions, instructions and rules listed in [3] and [4] one can employ effectively VCP/VCP2 coprocessors for decoding the WiMax convolutional code.

Associated programming of VCP/VCP2 registers and EDMA operation is quite involved and utilization of provided relevant Chip Support Library (CSL) functions is recommended. There are a few particular notes associated with utilization of those library functions which will be addressed in subsequent sections dedicated to VCP and VCP2.

It is assumed that VCP/VCP2 needs to provide hard decisions only. This is based on the primary assumption that there is no need to send the soft decisions back to the receiver chain for iterative processing ("turbo approach"). If this becomes needed, utilization of soft decisions will be considered.

2.1 Two modes of the WiMax convolutional code operation: tail-biting and zero-tailed

Tail-biting mode

In this mode the delay line of the convolutional encoder is initialized with the value of the last 6 info data bits (6 is the encoder delay line length). Therefore, the initial state of the trellis depends on the data contents and varies from frame to frame. The decoder does not have any knowledge about what this initial state might be. However, it does know that the first and the last states are the same ("tail-biting"). With this in mind and considering how the VCP and VCP2 coprocessors work we have decided on the following strategy. The coprocessors will be presented with the soft input (branch metrics) corresponding to a slightly larger block than the original one. The way the new block is going to look is shown in Figure 3.

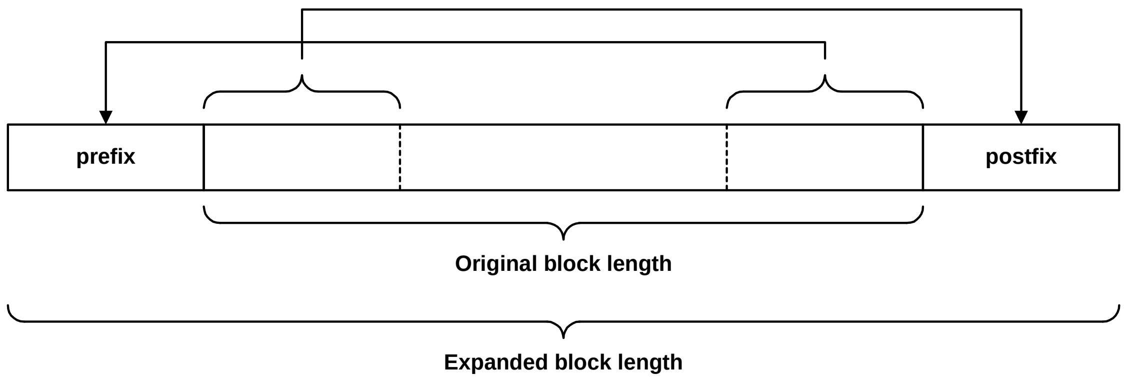

Figure 3. Block expansion in the tail-biting mode

VCP/VCP2 is expected to send back hard decisions corresponding to the expanded block length. However, only middle portion of the expanded block will be taken as the decoded information bits (original block length). The prefix and postfix lengths may be 3-5 delay line lengths (3x6 - 5x6). These are the complexity/performance trade-off factors. Considering the maximum WiMax required information block size, 36 bytes (288 bits) as well as the addition of prefix and postfix, one can compute that the maximum block size VCP/VCP2 may be presented with, is: 288+30+30 = 348. This is smaller than what the maximum allowed block size is for either VCP or VCP2 for the constraint length 7 code in the hard decision mode. Therefore, there is no need to split FEC block and use the sliding window approach (Mixed mode for VCP/VCP2). Rather, the Convergent mode can be used. In the Convergent mode the VCP/VCP2 is expected to provide hard decisions corresponding to the expanded FEC block. However, VCP/VCP2 also expects to be presented with branch metrics corresponding to a slightly larger block (Figure 4).

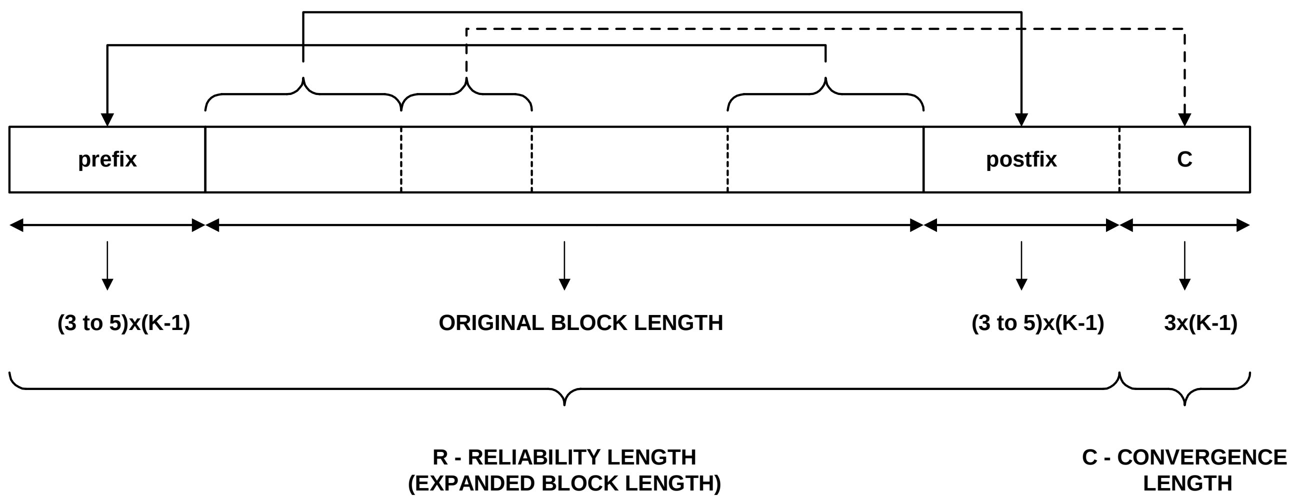

Figure 4. Block presented to VCP/VCP2 for decoding in the tail-biting mode

Terms R (reliability length) and C (convergence length) are taken from the VCP and VCP2 Reference Guides ([3] and [4], respectively). Assuming the maximum prefix/postfix length of 5x6 = 30 and C = 18 in this (R+C) arrangement, the maximum total BM array length is 30+288+30+18 = 366. This is still smaller than $(R+C)_{max}$ for VCP (372) and VCP2 (378) ([3] and [4], respectively). In the case of VCP (Laplace) postfix sometimes needs to be smaller than indicated in the figure. This will be addressed in Section 5.

The rationale behind this approach is as follows. Since there is no known end trellis state, Convergent mode of operation is applied. There is no known starting state either and therefore no state should be favored as an initial state. This is achieved by setting initial path (state) metrics for all states to the same value (IMINS = IMAXS = const; const = 0 in our coding example). It is expected that the accumulated path metrics at the end of the prefix part of the expanded block (beginning of the original block) will, with high probability, favor the true start state. Likewise, it is expected that going backwards from the end of the expanded frame, all the paths will converge to the same state which will, with high probability, be the true final state of the trellis corresponding to the original block length. Overall, this would mean that the bits corresponding to the original block length will be recovered correctly (of course, with the probability which depends on the reliability of the input to the decoder).

Convergence length (C) part may be embedded into the postfix part and prefix and postfix may not be of the same length.

VCP2 coprocessor offers a choice to apply another method (Method B) while decoding the tail-biting convolutional codes. The frame may be duplicated and two decoding passes are performed. During the first pass the start/end state is determined. The second pass starts at the state determined in the first pass. Only the method shown in Figure 3 and Figure 4 (Method A) is implemented and described in this document (this method applies to both, VCP and VCP2). We believe that significant increase in complexity of Method B cannot be justified (no noticeable performance gain is expected).

<u>Zero-tailed mode</u>

In this mode the delay line of the convolutional encoder is initialized with all zeros, so the initial state is the zero state. The final state is also driven to the zero state by appending an all-zero byte to the information block, [1]. Therefore, it is important to favor the zero state at the very beginning. This is done by setting the IMAXI = 0, which identifies that the zero state is the initial state, and also by setting IMAXS to a large (0x400 in our coding example) and IMINS to a small value (0 in our coding example), which correspond to the starting path metrics values for the favored (zero) state and other states, respectively. Traceback operation starts from the zero state. Figure 5 outlines the block structure in the zero-tailed mode.

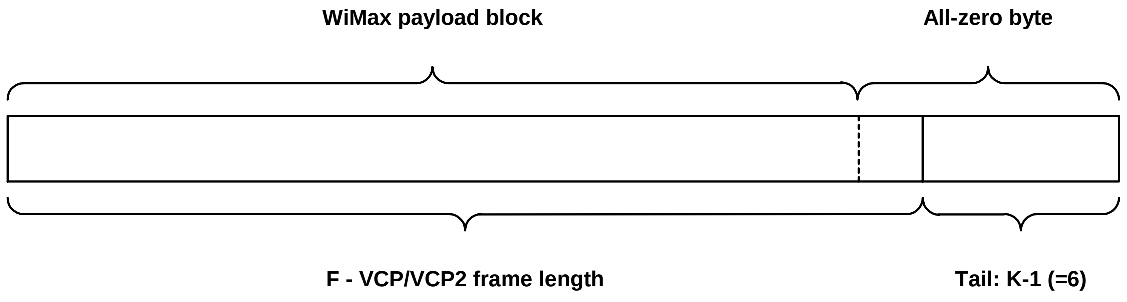

Figure 5. Block presented to VCP/VCP2 for decoding in the zero-tailed mode