TITLE: Decoding tail-biting convolutional codes with TCI6482's VCP2

DATE: July 18, 2005

AUTHOR: Eric Biscondi

1.1 Introduction

The IEEE standard 802.16e specifies the air interfaces for fixed and mobile broadband wireless access systems. This technical note describes how the TMS320TCI6482 VCP2's on-chip coprocessor can be used to decode tail biting convolutional codes specified in the IEEE standard 802.16e.

1.2 Description

The OFDMA PHY supports mandatory tail-biting convolutional coding and three optional coding schemes: zero tailing convolutional code, convolutional turbo code, and block turbo code.

The tail biting is implemented by initializing the encoder memory with the last data bits of the FEC block being encoded, and the zero tailing is implemented by appending zero tail bits (K-1 bits, K being the convolutional code constraint length) to the end of each burst.

Zero tailing bits are also used in the concatenated Reed-Solomon-convolutional codes (RS-CC) specified for the OFDM PHY.

Tail-biting convolutional code forces the encoder to begin and to end the treillis in the same state and it provides an advantage in term of bandwidth efficiency over the zero tailing convolutional code as no tail bits have to be transmitted.

1.2.1 Two passes decoding (Method #1)

In this method, a first decoding of the FEC block is realized to determine the final maximum state metric index (given by the last bits of the FEC block, but unknown at the decoder).

By construction of the tail biting codes, the final state metric index is the same as the initial state metric index.

At the end of a decode phase, the VCP2 generates the index of the final maximum state metric into the output register FMAXI. The final maximum state metric (FMAXS) as well as the final minimum state metric (FMINS) are also provided.

For the first pass decode, the VCP2 would be configured with the following parameters:

- IMAXI is arbitrary set to 0 as the initial state is unknown
- IMAXS = 0x400, IMINS = 0
- Trace back mode set to convergent.

At the end of the first pass decode, the CPU reads the register FMAXI (and store its value into a variable: *finalStateIndexOfPreviousPass*)

At this stage, a second decode operation of the same FEC block can be realized considering that the initial state is now known and equal to *finalStateIndexOfPreviousPass*.

Besides the registers IMAXI, IMAXS, and IMINS that respectively configure the initial state metric maximum index, maximum state and minimum state, the VCP2 also offer the possibility to initializes the index for the trace back processing.

For the first pass decode, the VCP2 would be configured with the following parameters:

- IMAXI = *finalStateIndexOfPreviousPass*
- IMAXS = 0x400, IMINS = 0
- Enable the initialization of the traceback starting state (ITBEN = TRUE)
- Set the index of the starting state for the traceback unit (ITBI = *finalStateIndexOfPreviousPass*)
- Trace back mode = convergent

The main disadvantage of this method is the fact that two different VCP2 decode tasks have to be configured by the C64x+ CPU. This significantly impacts the overall processing requirement and decreases the overall decoding bandwidth that the VCP2 can support.

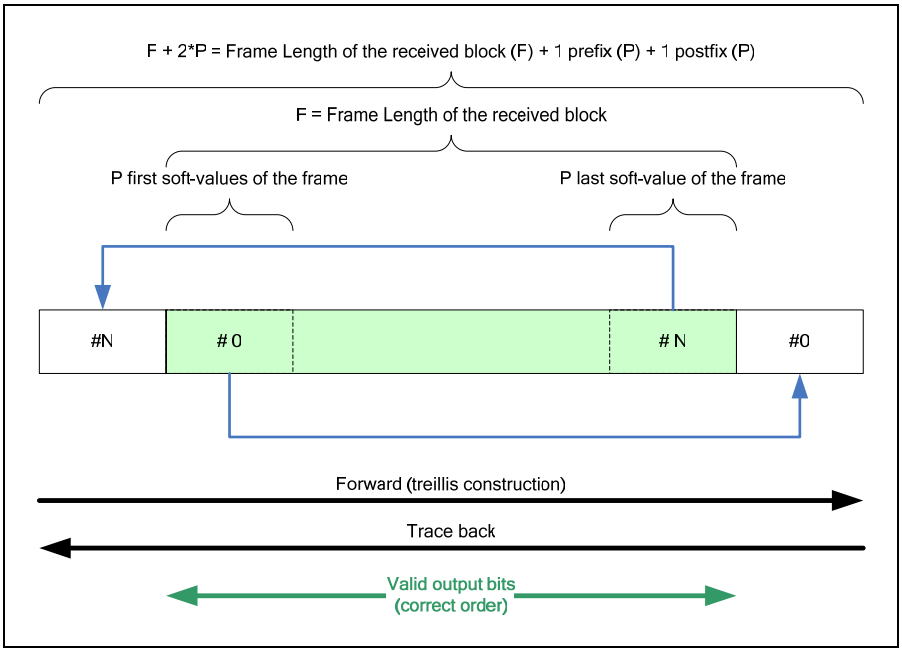
1.2.2 Single pass decoding (Method #2)

This implementation of the tail biting decoding method is based on:

- the fact that the convolutional code is circular, and
- the observation that the state at time *n* can be decoded using only the information from the interval *n-L* to *n+L*, *L* being the survivor path length.

A new FEC block is constructed by taking the received soft values and by adding a cyclic prefix and a cyclic postfix according to the scheme described on Figure 1.

Figure 1: Addition of the prefix/postfix



Even if the initial metric state is unknown at the receiver, the VCP2 starts the decoding of the new constructed frame with the initial state index arbitrary set to 0. After a certain distance know as the convergence distance, with a large probability, all the optimal paths from the other states will have converged to a single path. The size of the prefix/postfix must at least be set to the convergence length of the treillis (i.e. 3x to 5x the length of the code constraint length (K)).

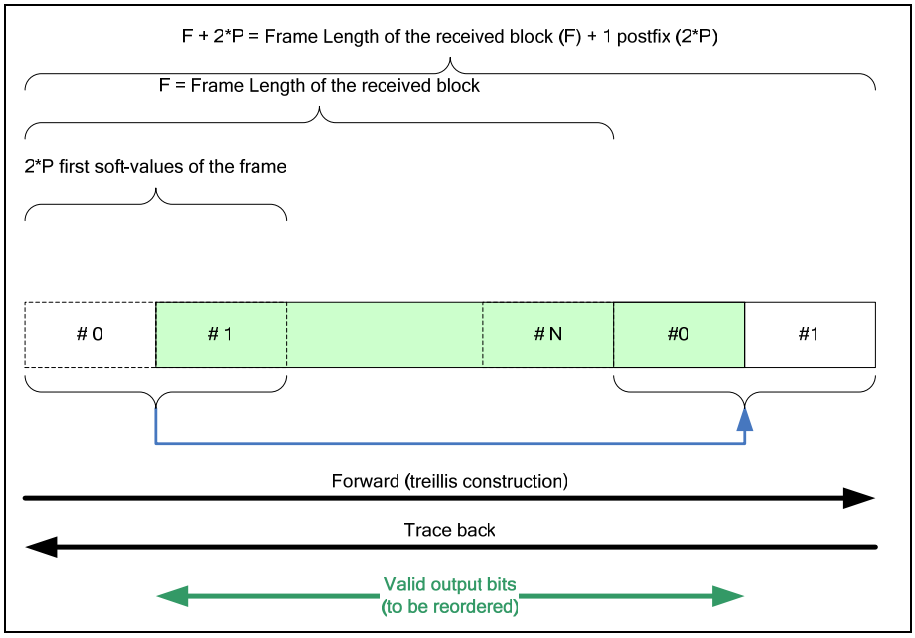
For the single pass decode, the VCP2 would be configured with the following parameters:

- IMAXI is arbitrary set to 0 as the initial state is unknown
- IMAXS = 0x400, IMINS = 0
- Disable the initialization of the trace back starting state (ITBEN = FALSE)
- ITBI = 0
- Trace back mode = convergent
- Frame Length = F + 2*P

This technique has the advantage to require a single VCP2 setting and to provide the data in the correct order. The disadvantage is of this technique is that the received soft input data block has to be rearranged (copying the prefix and postfix) before sending it to the VCP2.

Another slightly different technique is to extend the frame to decode by copying a postfix of size 2*P) from the beginning of the block to the end as shown on Figure 2 . The beginning of the new constructed frame is used as a training sequence to determine a valid initial state during the treillis construction and the end of the new constructed frame is used as a training sequence to let the trace back to converge toward the final state. The disadvantage of this approach is that the valid output bits have to be reordered.

Figure 2: Addition of the postfix



1.3 Performance

1.3.1 Cycles

1.3.1.1.1 VCP2 Cycles

The number of cycles required to decode a coded block depends on several parameters:

- the frame size (F)
- the number of sliding window (Nsw)
- the sliding window reliability size (R)
- the sliding window convergence size: (C)

The VCP2 User's guide will provide a generic description of all the equations used to determine the number VCP2 cycles required to decode a frame.

Considering the convolutional code specified in the IEEE standard 802.16e (Rate = ½ and K=7), the following formula determine the number of sliding window and the reliability length.

$$Nsw = ceil(\frac{F}{(378-C)}) \text{ and } R = 6 \times ceil(\frac{F}{\frac{Nsw}{6}})$$

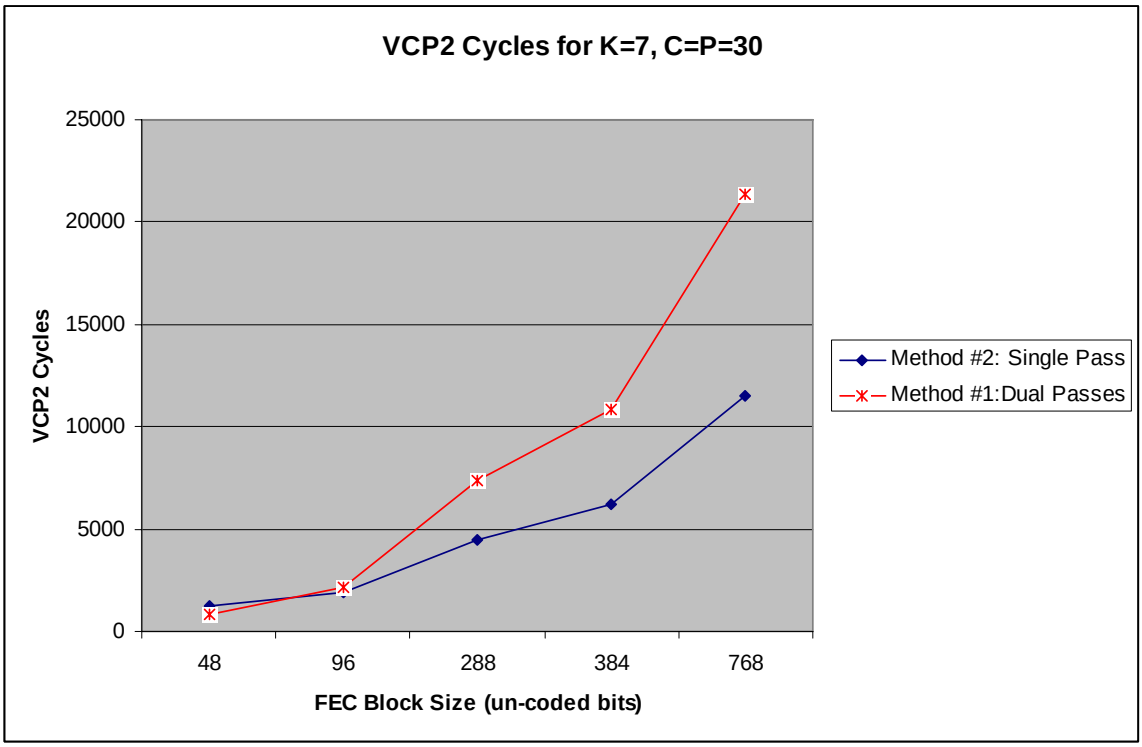
If we consider a FEC block of 288 un-coded bits with $C = 5 \times (K-1) = 30$ and if we assume the $P = C$ for the size of the prefix/postfix, the total number of VCP cycles that are required to

decode a FEC block of 288 un-coded bits is 3685 VCP2 cycles per decode pass. Therefore a total of 7370 VCP2 cycles are required to realize the decoding of frame of 288 bits.

For the second approach (single pass decode), the frame is increased by 2*P. The total number of VCP2 cycles to decode the frame is now 4505 VCP2 cycles.

Figure 3 describes the comparison of the number of VCP2 cycles for both approaches (with K=7 and P=C=30).

Figure 3: VCP2 Cycles vs. frame length for Method#1 (dual passes) and Method#2 (single pass)



1.3.2 Bit Error Rate and Frame Error Rate

Figure 4 and Figure 5 respectively show the FER and the BER for our example.

Figure 4: FER for R=1/2, K=7 Conv. Code (F=288 un-coded bits)

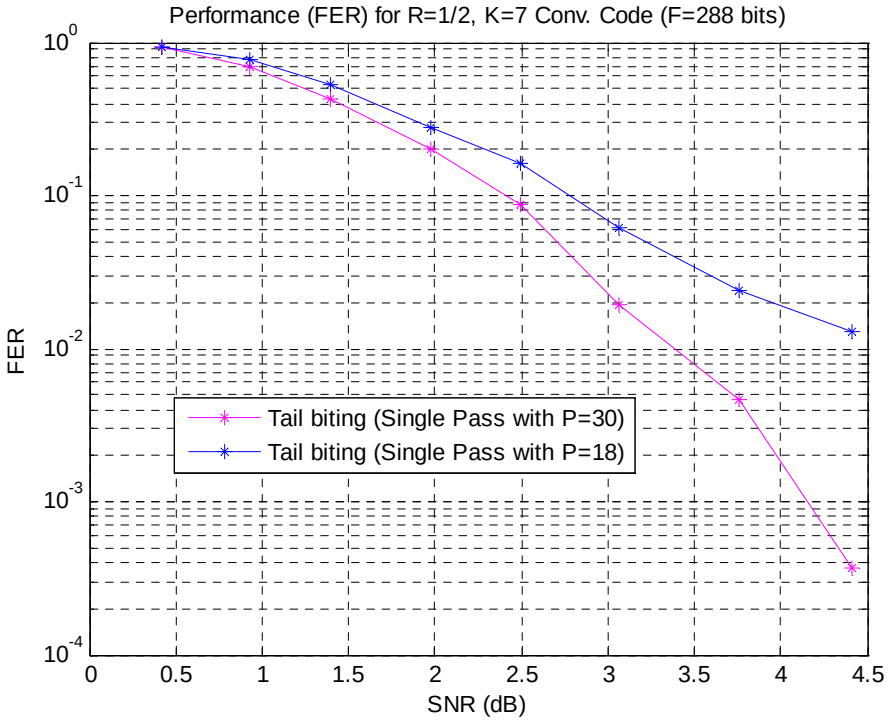
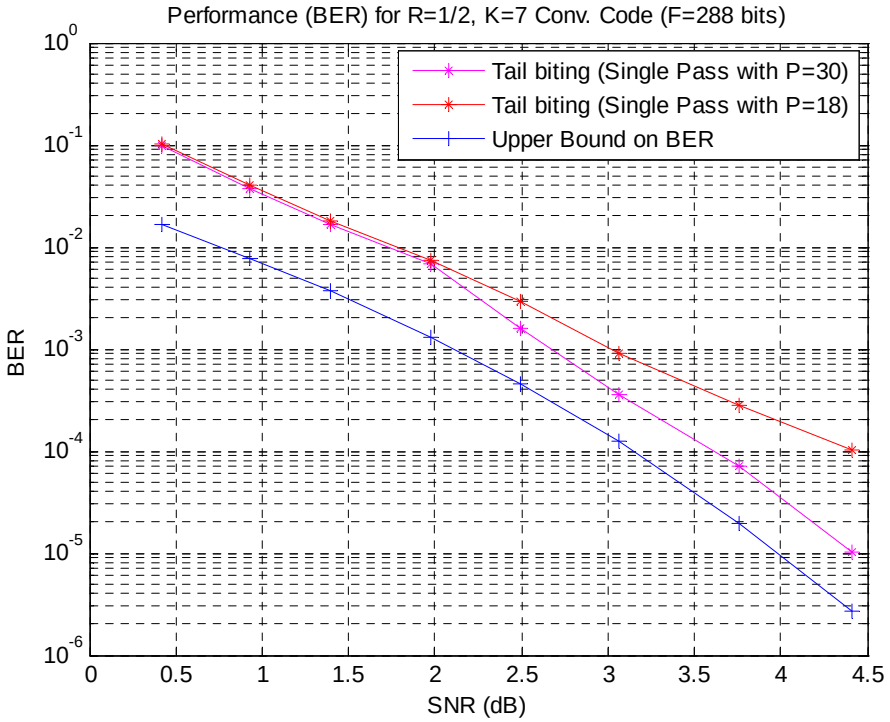


Figure 5: FER for R=1/2, K=7 Conv. Code (F=288 un-coded bits)



1.4 Revision

Date	Version	Revision
23 May 2005	DRAFT	Initial release
22 July 2005	0.0.1	Added some BER/FER as well as cycle performance figures