ECSE 436 **McGill University**

Prof. Bajcsy **Winter 2018**

Laboratory Assignment 2 – part 1

Due: with part 2

Reading: Lecture and lab/tutorial notes, handouts on Altera FPGA signal processing boards, relevant sections in reference book by Lin and Costello (available online).

1. FPGA Signal Processing Boards and Convolutional Codes

(a) Consider rate ½ convolutional code used in cellular phones to protect speech and data against noise. The code has generator polynomials in octal form G_1=23 and G_2=33, i.e., in binary form G_1= [1 0 0 1 1] and G_2= [1 1 0 1 1]. Draw finite state machine representation of this encoder.

(b) Draw one full stage of the trellis diagram for this code that has 16 states, carefully labelling states, input and output labels on each trellis edge.

(c) Using Matlab, implement encoder for this convolutional code assuming the message has length 16. (The encoder starts in 0 0 0 0 state and 4 dummy zeros are added at the end of the message block to re-set the encoder.) Submit your Matlab script as well as the result of encoding message **m =** [0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1].

(d) Implement the encoder using the Altera FPGA board, using provided random source script to generate message bits. Demonstrate your encoder to the TAs and submit your implementation script and/or diagrams.

2. Viterbi Decoding Algorithm for the Hamming (8,4,4) Code

Hamming codeword **c =** (c1, c2, c3, c4, c5, c6, c7, c8) has been transmitted over an AWGN channel with noise mean being 0 and a positive variance. The received vector **r = c + noise** is observed to be **r =** (0.54, - 0.12, 1.32, 0.41, 0.63, 1.25, 0.37, -0.02).

(a) Using Matlab, the code book of the (8,4,4) code and material discussed in the lecture/lab, determine the square Euclidean distances between **r** and each of 16 codewords. Determine the closest codeword to **r** and submit your script.

(b) How many floating point operations (additions, multiplications and comparisons) does the approach from part (a) use to decode a codeword?

(c) Design an optimized trellis diagram for this code that has at most 4 states per stage.

(d) Redo part **(a)** by hand using Viterbi decoding algorithm for the 4-state optimized trellis. Determine the appropriate edge metric for individual trellis edges, then use the survivor path approach (ADD-COMPARE-STORE) to determine your answer. Submit your decoding trellis and results of each stage survivors search and their cumulative metrics.

(e) How many floating point operations (additions, multiplications and comparisons) does the approach from part (d) use to decode received **r**?

(f) Reduce the floating point operations in (d) by modifying the edge metric appropriately. How many floating operations you need to decode now?

(g) Use Matlab to implement and test the Viterbi decoder from part (c) for general received **r = c+n**.

(h) Use your Viterbi decoder from part (g) to simulate bit error rate performance of the (8,4,4) Hamming code on the additive Gaussian noise channel. Use 'randn' function to generate Gaussian noise samples and set the noise variance so that the SNR ranges between 0 dB and 10 dB, with a step of 1 dB. Submit your BER vs. SNR plot using 'semilogy' scale as well as the script/s used to obtain it. (**HINTS:** Calculate the errors only for the message bits and simulate each SNR value until you obtain 100 decoding errors or until you run 10,000 packets, whichever is less.)