ECSE 436 **Laboratory Assignment 3 (part 2)** **Winter 2018**

McGill University Due: with part 3 **Prof. Jan Bajcsy**

3. Computing Spectrum of Audio Signals

(a) Determine DFT coefficients of signal x[n] represented by vector [1 -1 0 2] and compare it to the DTFT transform $X(e^{jw})$ of x[n] assuming x[n] = 0 for n<0, n>3.

(b) State and prove the linearity and time delay properties for DFT.

(c) Evaluate circular convolution for x[n] = [1 -1 0 2] and h[n] = [1 0 1 0].

(d) State and prove the circular convolution property for DFT.

(e) If X(k) denotes DFT coefficients for length N signal x[n], show that the inverse DFT of X(0), X(1), ⋯, X(N-1) yields the original signal x[0], x[1], ⋯, x[N-1].

(f) Write your own Matlab function *my_DFT* to implement DFT for a specific input signal in vector **x.** Use the summation expression from the basic DFT definition.

(g) Determine the number of FLOPs to get all DFT coefficients in part **(f).**

4. Fast Fourier Transform Algorithm

(a) Show how through decimation in time, DFT of a length N signal can be determined through calculation of two N/2 DFT's and the use of appropriate 'twiddle' factors. (Hint: N is assumed to be even, then separate the original DFT sum into two sums for odd and even time indexes n.)

(b) If N is a power of 2, repeat the approach from **(a)** to further reduce length N/2 DFT into two length N/4 DFT's, then a length N/4 DFT into two length N/8 DFT's, etc. until only size 2 DFT's and appropriate twiddle factors are needed.

(c) Determine the reduced computational complexity of calculating DFT through the approach in **(b).** Plot the computational complexity as a function of signal length N and compare it to computational complexity for the original approach from Question 3 (g). Use $N=2^M$ for M=1,2,⋯ ,30.

(d) Implement your own Matlab function '*my_FFT*' that uses recursive calls to implement FFT for length $N= 2^M$ signals. Test your routine and compare your transform results to the ones obtained by the built in FFT.

(e) Implement your own '*FFT_16*' routine to calculate length 16 FFT using appropriate butterfly structure. Test your implementation against built-in FFT.

(f) Use the FPGA board to implement and demo the algorithm from(e) in real time.