EE6322: VLSI Broadband Communication Circuits
Assignment 1

Nagendra Krishnapura (nagendra@iitm.ac.in), Saurabh Saxena

due on 14 Aug. 2016

1. Model the channel as a first order RC filter with 0.5 GHz bandwidth. Determine the delay between the input and output zero crossings for the following cases. All inputs are ideal rectangular waveforms between -1 and $+1$.

   - A $101010\ldots$ pattern at 1 Gb/s
   - A $11001100\ldots$ pattern at 1 Gb/s
   - A 1 GHz clock with 50% duty cycle. For this case, repeat the calculations if the channel bandwidth is 20% higher or lower than the nominal value above.

   Determine the peak-peak value of the clock at the output of the channel for the nominal bandwidth.

2. A 1 Gb/s binary data has levels of ± 100 mV. Ideally the clock is placed in the middle of the bit interval. Determine the following. Both noise and jitter are zero mean Gaussians.

   - σ of the added noise for $\mathrm{BER} = 10^{-6}$, 10^{-8}, 10^{-10}, 10^{-12}, 10^{-15}.
   - RMS jitter σ_τ so that the probability of the sampling instant moving outside the bit interval is 10^{-6}, 10^{-8}, 10^{-10}, 10^{-12}, 10^{-15}.

3. Plot the eye diagrams with for the following channels for -3 dB bandwidths of 0.5 GHz and 0.75 GHz. Assume that the input is random binary data with levels of ± 1 at 1 Gb/s. Run the simulation for 1000 bits and remove the initial transients. For each of the channels, the pole locations for -3 dB bandwidth of 1 rad/s are given below. All zeros are at infinity. In each case, plot the eye diagram over a 2 bit interval and mark the horizontal and vertical eye openings (estimated by looking at the eye diagram).

	Type	Poles (rad/s)
1	First order RC	-1
2	Fourth order Bessel	$-0.9952 \pm j1.2571$ $-1.3701 \pm j0.4102$
3	Fourth order Butterworth	$-0.9239 \pm j0.3827$ $-0.3827 \pm j0.9239$
4	Fourth order Chebyshev	$-0.3200 \pm j0.3869$ $-0.1325 \pm j0.9341$

   To do this problem, first you have to create the channel with appropriate poles and zeros. Brush up your filter frequency scaling and don't confuse Hz and rad/s. In MATLAB, the same filter can be represented with zeros and poles, or numerator and denominator polynomials, or state-space matrices. Familiarize yourselves with `zp2tf, tf2zp, tf2ss, ss2tf` etc. You then have to generate a random bit pattern (see `rand, randn`), pass it through the channel (see `lsim`), and plot the eyediagram (see `eyediagram`). The signals are continuous-time, but for simulation, you have to sample them. In this case, since the simulations are simple and run quickly, you can use a large number of points, say 100, per bit interval. But, in more complicated cases, this may present a trade-off between simulation speed and accuracy and has to be chosen carefully. For tools other than MATLAB, you can find the equivalents for the above.