Lab 7 - Bit Error Rate Measurement

Introduction

You will use a Field Programmable Gate Array (FPGA) to generate data and noise signals and measure Bit Error Rate (BER).

BER is the fraction of bits that are received incorrectly and is a common way to measure the performance of digital communication systems.

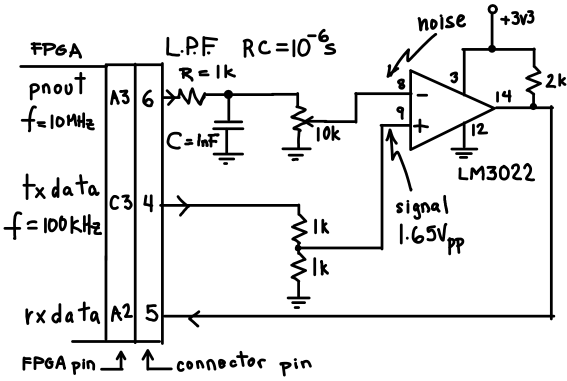
You will measure the BER of an NRZ data signal that has Gaussian noise added to it. This is called an Additive White Gaussian Noise (AWGN) channel. You will then compare the experimental BER results to the results predicted by the theory covered earlier in the course.

FPGAs are ICs containing thousands of simple logic elements that can be configured to perform arbitrary digital logic functions.

You will use an Altera *DE-0 Nano* FPGA board that includes an FPGA, 8 LEDs, two push-buttons and two connectors with 40 I/O pins each.

Circuit Description

The hardware you will put together to measure the BER over the AWGN channel is shown below:

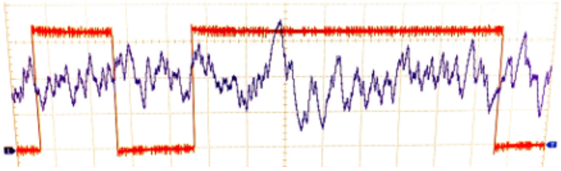


`pnout` is a logic-level (0 to 3.3V) 10 MHz Pseudo-Random Bit Sequence (PRBS) with a period of $2^{31}-1$ bits. It is low-pass filtered by an RC filter with a 1 μs time constant. This produces an analog voltage that varies in a pseudo-random manner according to the bit pattern and has a probability density function that is approximately Gaussian. The DC level (average) and the power (variance) of the noise are both adjusted by a 10k pot acting as a variable voltage divider.

`txdata` is a 100 kbps (kHz) PRBS data signal with a period of $2^{10}-1$ bits.

The data and noise signals are input to an LM3022 comparator. The comparator subtracts the noise signal from the data and outputs a high logic level if the result is greater than zero. Thus the noise causes an error when the data signal is positive and the voltage of the noise signal exceeds the voltage of the data signal.

The following 'scope photo shows the data and noise signals at the input to the comparator. Near the middle of the trace the noise voltage is large enough to exceed the data voltage and would produce a high output from the comparator.



The FPGA compares `txdata` and the output of the comparator (`rxdata`) to detect errors. Bits 0, 4, 8, 12 and 16 of a 17-bit error counter are displayed on the board's LEDs so you can tell when the error count reaches certain values (1, 16, 256, 4096 and 65536 respectively).

A `reset` signal is driven by the pushbutton `KEY0`. It has two functions: it resets the error counter to zero and initializes PRBS generators. Note that **your circuit will not generate noise or data until it has been reset!**

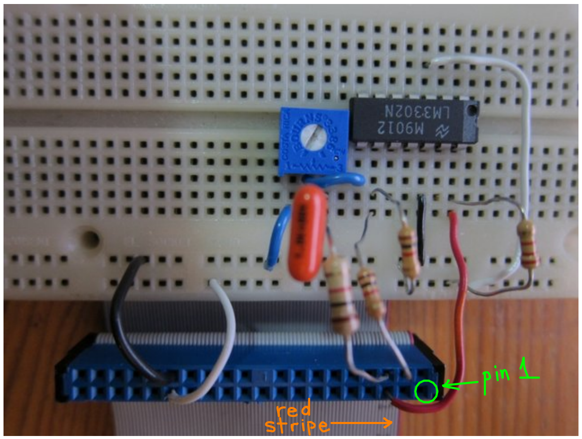
Lab Procedure

Assemble the AWGN Channel

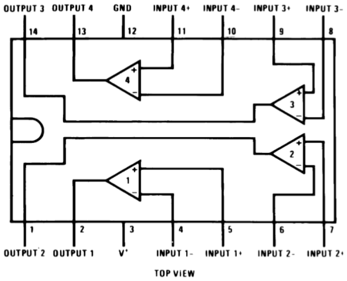
Connect a supplied 40-pin ribbon cable to the GPIO0 connector on the FPGA board (see photo below).

Build the AWGN channel circuit on your solderless prototyping board and make the connections to the ribbon cable connector as shown below. Obtain 3.3V from the ribbon cable connector pin 29 and ground from pin 30.

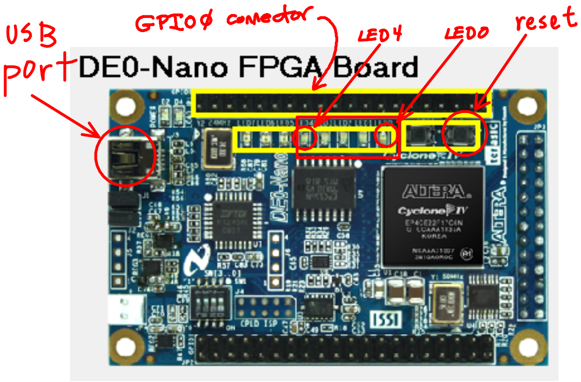
The following photograph shows how connections can be made from the ribbon cable connector to the solderless prototyping board by treating the connector as if it were another prototyping board:



The pinout of the LM3302 is as follows:



The following photograph shows the LEDs and pushbuttons on the board:



You will use the `reset` pushbutton (labelled `KEY0` on the board) to reset the error counter. LEDs labelled LED0 through LED4 turn on when the error counts reach 1, 16, 256, 4096 and 65536 errors respectively.

The following diagram labels the connection points on the ribbon cable connector with pin 1 (red conductor) on top. The letter-number pairs are the row and column respectively on the 256-pin (16×16) BGA (ball-grid array) FPGA package; numbers beginning with zero (0) are signal names used in the the DE-0 documentation (prefixed with `GPIO_`), the other numbers are the 40-pin "header" connector numbers. For example, the signal on FPGA pin A2 has signal name GPIO_02 and is on pin 5 of the connector (third pin from the top on the right side).

D3	00	2	□	□	1	0_IN0	A8
C3	01	4	□	□	3	0_IN1	B8
A3	03	6	□	□	5	02	A2
B4	05	8	□	□	7	04	B3
B5	07	10	□	□	9	06	A4
GND	–	12	□	□	11	–	5V
D5	09	14	□	□	13	08	A5
A6	011	16	□	□	15	010	B6
D6	013	18	□	□	17	012	B7
C6	015	20	□	□	19	014	A7
E6	017	22	□	□	21	016	C8
D8	019	24	□	□	23	018	E7
F8	021	26	□	□	25	020	E8
E9	023	28	□	□	27	022	F9
GND	–	30	□	□	29	–	3.3V
D9	025	32	□	□	31	024	C9
E10	027	34	□	□	33	026	E11
B11	029	36	□	□	35	028	C11
D11	031	38	□	□	37	030	A12
B12	033	40	□	□	39	032	D12

Download the `lab7.sof` file from the course web site. This file contains the configuration ("programming") information that must be loaded on the FPGA to configure it for this lab.

Connect a USB cable from the PC to the USB connector on the FPGA board. The blue power LED should light and the LEDs should show the default "breathing" demo.

Run the Quartus II version 13 programmer (Start -> Altera -> Quartus II Web Edition -> Quartus II Programmer). If not already selected, click on "Hardware Setup" and select "USB Blaster." If not already selected, click on "Add File" and add the `lab7.sof` file. Press "Start" to program the FPGA.

Press the KEY0 button to reset the FPGA. If the FPGA is configured properly, LEDs 5 and 6 should turn on when you press the button.

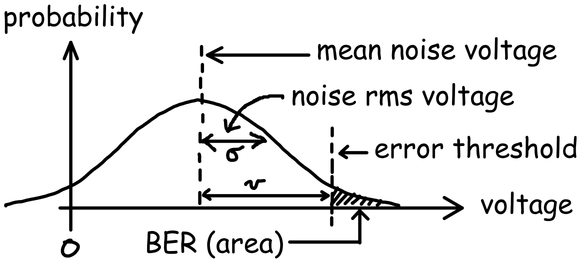
Connect one 'scope channel to the non-inverting (data) input of the comparator and one to the inverting (noise) input. Enable the 'scope's average and RMS voltage measurements for the noise channel and enable measurement statistics (see Measurement Hints section below). You should see waveforms similar to those shown above.

BER Measurements

Adjust the noise power until you get *approximately* one change every two seconds on the least-significant LED (`LED[0]`). Press the reset button and measure, using a clock, the time it takes for the next most-significant LED to turn on (about 30 seconds). This is the time it took to count 16 bit errors. Multiply this time by the data rate (50,000 bits/second since only positive data bits are checked) to get the number of bits checked.

Measure the high-level signal voltage (the error threshold voltage) using the measurement cursor. Measure the mean (DC) noise voltage and the noise variance (see Measurement Hints below).

The following diagram show the probability distribution of the noise, the error threshold (the voltage at which the error counter is incremented) and the area corresponding to the BER:



Predict the BER using the equation:

$$P_e = \frac{1}{2}\text{erfc}(\frac{v}{\sigma\sqrt{2}})$$

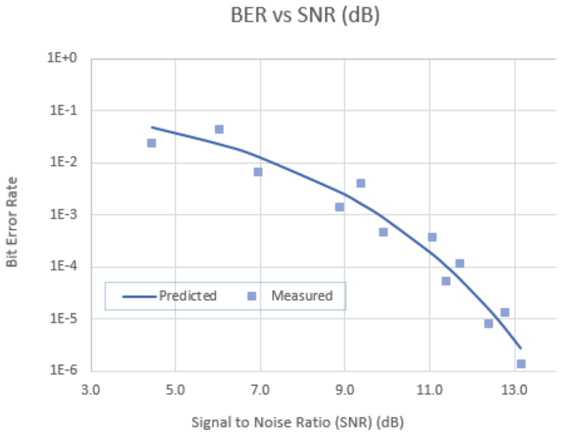
where v is the positive signal voltage (approximately 1.65) minus the average noise voltage, and σ is the standard deviation of the noise (this is the noise RMS voltage if the signal is AC coupled; see below). Compare your measured and predicted BER values.

Enter your measurements into the spreadsheet[1]. It should include columns for the error count, elapsed time, measured BER, signal voltage, average noise voltage, noise variance and predicted BER.

Repeat for the next most significant LED (256 errors): increase the noise power so the blink rate on `LED[1]` is about one every two seconds and repeat the BER and SNR measurement and the BER prediction calculations.

Repeat for the remaining LEDs so that you have four measured and four predicted BER values.

Insert a graph (chart) in your spreadsheet comparing the measured and predicted BERs. The BER should be plotted on a logarithmic scale and the SNR (ratio of signal voltage to square root of the variance) should be in dB as shown below (your values may be different):



Measurement Hints

You will need to measure the signal and noise powers carefully to get useful results because the BER changes quickly over a narrow range of SNRs. Use the following guidelines when making measurements:

- set the voltage offset and gain to get the largest waveform display that does not clip the signal
- turn on bandwidth limiting (20 MHz) and use a sampling rate that avoids aliasing (auto sample rate with about 5μs/division sweep rate for this lab)

[1] It is good practice to make a permanent record of your measurements in a lab notebook.

- use the 'scope's measurement feature along with measurement statistics. Average at least 20 measurements and continue until the average stops changing.
- measure the variance of the noise voltage by measuring the RMS voltage of the AC-coupled input. AC coupling removes the DC (average) component so that the RMS value is equal to the square root of the variance[2]. The average voltage must, of course, be measured with DC coupling or it will be zero.

Report

Submit a report in PDF format containing:

- the identification asked for in Lab 1
- the spreadsheet showing your predicted and measured values
- the graph showing the predicted and measured BER versus SNR

[2] The variance is the second central moment: $\overline{x^2} - \overline{x}^2$.