

## **Project 4: Creating an Arbitrary (ARB) Waveform for Various Digital Baseband Modulation Formats (Line Codes) Using MATLAB and the HP/Keysight 33120A**

### **Notes to Students:**

Each student must turn in a separate report. The lab section can be done in groups of TWO. You must specify the name of your Lab partner on your individual report. Please try to reduce the size of plots and tables to avoid wasting papers. Double-sided reports are welcomed!

### **Time of Completion:**

Estimated time to complete the measurements is about 2-3 hours. Please note that you may have to go back again to redo some of the measurements. DONOT wait until the last day!

### **Important:**

- All the connectors are provided in the lab. In some cases you have to search for the right connector for the setup.
- Please make sure all laboratory components are returned back to the testing table.
- Please keep the lab clean! Make sure you sign the signup sheet!
- Please be Gentle with the connectors!
- All FIGURES must have Figure Number and Caption! Figures without these will not be acceptable!

### **HW & SW required for the lab:**

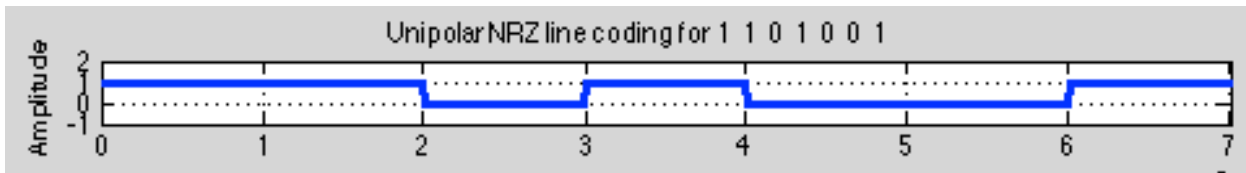
- Mini-Circuit Mixer
- Matlab software + Test & Measurement toolbox
- Function Generator (33120A)
- RF source
- Scope
- Spectrum Analyzer (100KHz to 1GHz)
- USB to GPIB Converter
- USB cable
- UHF N Type Cables [2]

### **Objectives:**

The purpose of this lab is to understand.....

**Pre-lab:**

**A-** Create a random unipolar NRZ bit sequence generator using MATLAB. HINT: Make sure to save the bit sequence as an array type. Use the rand and round functions. Your result should look something like this:



**B-** Find the PDF for each data bit (one and zero). In other words, demonstrate the likelihood of generating a one or a zero, assuming 1000 bits are generated. Show your **histogram**.

**C-** Use the [Matlab Code](#) for Line Coding provided on the class web site. Show the **Manchester** line code for bit stream = [101011110].

**LABORATORY ACTIVITY :**

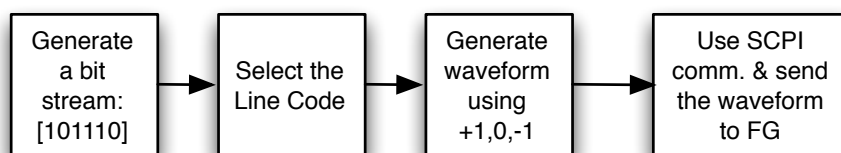
In this section you should use the digital input bit sequence array you created in part A of the Pre-Lab (using Matlab) to create an arbitrary waveform using the function generator (FG) 33120A. Configure your setup as shown below. Note that you can connect the function generator to *either* scope or the spectrum analyzer. (SA).



The basic idea in this lab is to write a Matlab program such that the user can input any arbitrary bit stream, say [101110]. Then, depending on the line coding scheme coded in Matlab the user should generate the waveform using +1,0, -1. Here is an example: assuming we would like to apply bipolar NRZ line coding scheme to the bit stream[101110] the following process must happen:

[101110].—>[11111111 -1-1-1-1-1-1-1-1 11111111 11111111 -1-1-1-1-1-1-1-1]

Note that in this case each bit is mapped into 8 signal levels. The following figure summarizes the basic functionality of the Matlab code.



The generated line code (e.g., [11111111 -1-1-1-1-1-1-1-1 11111111 11111111 -1-1-1-1-1-1-1-1] ) will be converted real voltage level using the DAC of the functional generator: [+V -V +V +V]. The bit period and value of V depends on the setting on the FG.

In this process the Matlab code must be able to sent the generated line coded signal to the FG to convert the +1/0/-1 to an actual waveform. Matlab can communicated with the function generator using Standard Commands for Programmable Instruments ([SCPI](#)) (often pronounced “skippy”). We note that in order to run SCPI commands we need to have Test and Measurement Toolbox in Matlab.

The following Matlab code example shows a simple case where we communicate with the function generator (33120A) using GPIB interface. Note that in this case we are only changing the ARB pattern using FUNC:USER VOLTAGE command and thus generating a waveform. We use the DATA VOLATILE command to download floating-point values between -1 and +1. These are amplitude values interpreted by the DAC in the function generator. Remember in our case use a minimum of 8 amplitude values per bit of the digital input sequence.

```
interface = gpib('agilent', 7, 30); %use GPIB address to connect to
% instrument and assign the connection to a variable named interface

fopen(interface) %"open" the connection
fprintf(interface, '*idn?') %ask the identity of 33120A
idn=fscanf(interface) %read response from 33120A and display it as
fprintf(interface, 'FUNC:USER VOLATILE')
arbCSV = regexprep(num2str(arb,17), '\s+', ',');
fprintf(interface, 'DATA VOLATILE,%s',arbCSV)
fprintf(interface, 'FUNC:USER VOLATILE')

fclose(interface)
```

**PART 1:** Follow the steps below:

Step 1: Write a single Matlab code to generate appropriate waveforms using any given bit pattern and the following line coding scheme: Unipolar NRZ, Polar RZ, and Manchester.

—>On a single plot show the generated line code for 101011110 for each of the above line coding schemes. **Important note:** you may have to press the ARB button on the FG to activate the code.

Step 2: Make sure your step up has been configured according to the figure shown above. Using the provided Matlab code for SCPI interface, send your waveform to the Function Generator. Observe the waveform on the scope. Set the frequency to 10KHz. Answer the following questions:

1. Set the frequency on the FG to 10KHz. Assume Manchester line code. Measure the bit rate (Rb) and symbol period (Ts).
2. In the above case, what happen to Rb and Ts if you change f on the FG to 100KHz?
3. Take a snapshot of the signal on the scope using Manchester line coding. Record the frequency, bit rate, amplitude (+/-), Ts for the given bit pattern.

Step 3: Connect the output of the FG to the spectrum analyzer. Change your bit pattern to [10 00 00 00 00] - that is we are sending 10 bits, all being zero except one. Set the

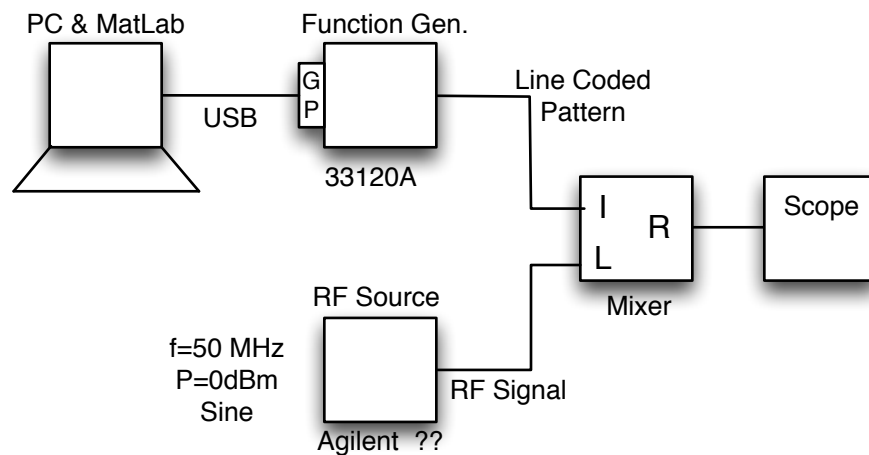
frequency of the FG to 100KHz. Capture the spectrum of the signal when unipolar RZ is used. Answer the following questions.

1. What is the pulse width of the signal as you observe it on the Scope?
2. Take a snapshot of the generated signal as shown on the scope.
3. Take a snapshot of the generated signal, as shown on the SA. You should change the span such that you see a complete SINC function with several zero crossings.
4. Change the frequency of the FG. What exactly happens to the shape of the SINC function as you change the frequency on the FG? Clearly explain.
5. What exactly is the relationship between the the pulse width and the frequency difference between zero crossing points on the SINC function?
6. Change the line coding from unipolar RZ to Manchester. Take a snapshot of the generated signal, as shown on the SA. What is the difference? Explain.

**PART 2:** In this section using different line coding schemes we create digitally modulated signals. We only focus on BASK and BPSK.

Step A1: Using your Matlab program generate the waveform for 101011110 bit stream, assuming we are using **unipolar NRZ line coding**. Check the scope and make sure the generated waveform is correct.

Step A2: Configure your setup as shown below. Note that the output of the mixer must be connected to the scope. Note that the Power level in the RF source is set to 0dBm with Frequency of 50 MHz.



Step A3: Take a snapshot of the signal on the scope. What type of modulation is this (remember you are using unipolar NRZ). On your snapshot identify where 1's and 0's are. Explain how a one changes to a zero. What exactly happens?

Step B1: Using your Matlab program generate the waveform for 101011110 bit stream, assuming we are using **Bipolar NRZ line coding**. Check the scope and make sure the generated waveform is correct.

Step B2: Configure your setup as shown below. Note that the output of the mixer must be connected to the scope. Note that the Power level in the RF source is set to 0dBm with Frequency of 50 MHz.

Step B3: Take a snapshot of the signal on the scope. What type of modulation is this (remember you are using **bipolar NRZ**). On your snapshot identify where 1's and 0's are. Explain how a one changes to a zero. What exactly happens?

### **Acknowledgment:**

We would like to thank Mr. Books Hanley for his assistance and contributions in developing this laboratory activity, Matlab code, and testing the setup.

### **References:**

[1] Reference: 33120A User Guide mirrored at the University of Illinois: <http://mntl.illinois.edu/equipment/docs/agilent33120auserguide.pdf>

[2] Connector types and their performances: <http://highfields-arc.co.uk/constructors/info/connchart.htm>