Victor Lam
April 18, 2020
ECE 231 Lab 3
PWM and Timestamping on BeagleBone Black

Part I: Jitter Distribution Plots

Figure 1: Jitter Distribution Plot for 1Hz and 50% Duty Cycle

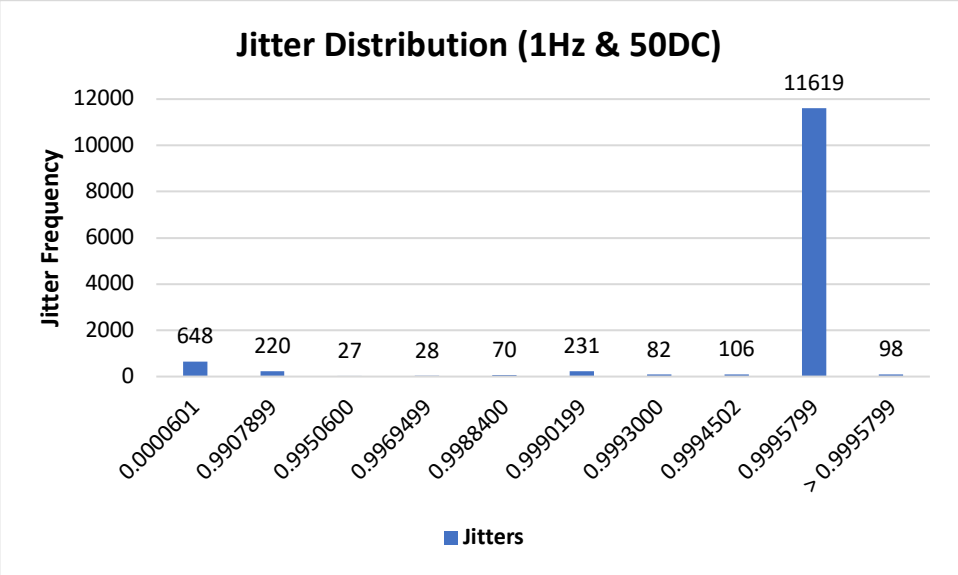


Figure 2: Jitter Distribution Plot for 100Hz and 50% Duty Cycle

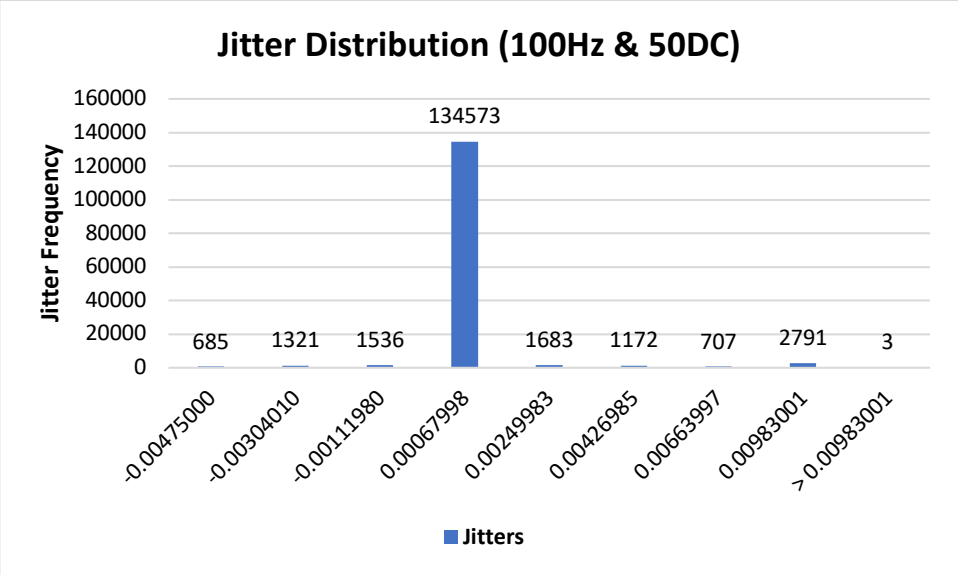
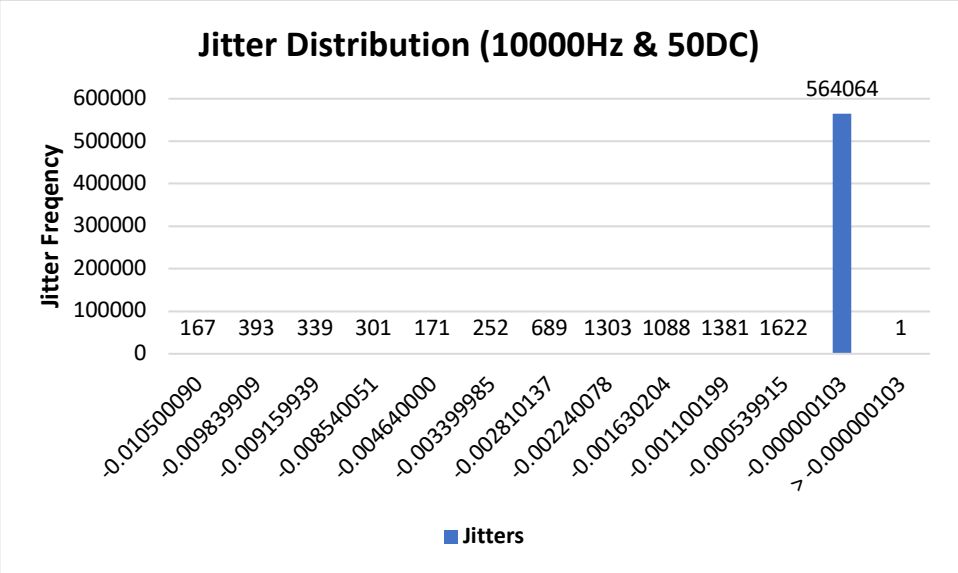


Figure 3: Jitter Distribution Plot for 10000Hz and 50% Duty Cycles



Part II: Means and Standard Deviations

Frequency:	**1Hz**
Mean	0.933750646
Standard Deviation	0.247678329

Frequency:	**100Hz**
Mean	0.000166095
Standard Deviation	0.001488139

Frequency:	**100Hz**
Mean	-0.000120099
Standard Deviation	0.00051409

Part III: Quantitative Analysis

Jitter is the variation in delay of a signal, and in every embedded system, there is going to be jitter. It is unavoidable, but it can be worsened or reduced. In our BeagleBone Blacks (BBB), we have several potential factors that may worsen the jitter like our boards having a poor crystal oscillator. These crystals are used to keep track of time, and if the crystal is unstable, then we may end up with smaller periods. The environment may also affect the jitter as the crystal needs to be in certain conditions to operate at optimal levels. If the next rising edge comes too fast and our code is just timestamping the last edge, we will miss the edge entirely, waiting longer for the next edge. Because we are measuring the period of the PWM using timestamps, we have to consider faults there too. Code efficiency is another factor, as the BBB only timestamps when we tell it to. If another edge comes and the code is just getting around to writing the last timestamp to the external file, then it may miss the edge entirely. There may

also be a problem if there's too many lines of code from "if GPIO.event_detected(pin)" to the timestamp, then there will also be a delay as the BBB has to fulfill the other tasks before getting to timestamping. This bring us to a really obvious factor, processing power. If the Sitara AM335x had a higher clock speed than 1GHz and multiple cores, we may see better timestamps, especially for the 10000Hz frequency. Although it isn't apparent in our data, it is probable that the higher the frequency, the higher the measured jitter as the BBB may not be able to keep up with the rising edges or timestamp fast enough so the jitter could be low but the measuring is off due to system limitations.

Things we could change to improve (decrease) jitter is to get a better crystal and CPU. This will help in not only having a stable and accurate clock so that the PWM is constant and accurate, but so that our measuring of the jitter will be also more accurate. Optimizing code is also important so that the chances of there being a missed rising edge is lower.