

CESR BPM System Calibration

Joseph Burrell

Mechanical Engineering, WSU, Detroit, MI, 48202

(Dated: August 11, 2006)

The Cornell Electron Storage Ring(CESR) uses beam position monitors (BPM) to determine certain aspects of a particle beam inside of the storage ring. Using this system, it is possible to determine such characteristics of the beam as its position in the pipe and the spacing between each bunch. This project pertains to the calibration and testing of the readout hardware of the BPM. The tests that were run this summer are important to help characterize the noise and linearity of the BPM, thus minimizing the amount of errors received from readouts.

I. INTRODUCTION

CESR provides colliding beams for high energy physics research with the Cornell Large Experimental Operations(CLEO) detector and x-rays for the Cornell High Energy Synchrotron Source(CHESS). The previous BPM system was not suitable for doing high energy physics research on a continuous basis. CESR is preparing to become a testing facility for the ILC damping ring. The current BPM system must be upgraded for future tests. The beam position monitoring system must be able to operate on a much higher level. The BPM must be able to measure the beam position with a high resolution on a turn-by-turn basis.

Requirements listed include individual bunch-by-bunch resolution, true single turn operations and a large dynamic range for measurements. The current system is only able to do single bunch, turn-by-turn measurements. A newer system has been developed to do multiple bunch-by-bunch measurements. The readout for the CESR test facility needs to be fast (on the nanosecond timescale) and with a high resolution (on the micron level) with which noise possibilities and lack of linearity becomes a challenge. The goal of this summer's measurements are to characterize the noise and linearity of the BPM upgrade.

II. HARDWARE OVERVIEW

Most of this summer's work revolved around the use of various codes, and two different programs were used. These two programs were "BPMV1" and "Burrellclock," both written in Fortran. "BPMV1" was a program already established before this summer at Cornell, and is meant as an all-purpose testing program for the BPM systems. "Burrellclock" is an edit from "GETBPMCLOCK," which was meant to determine the "best" clock settings. Using these two programs, various tasks and experiments were conducted, including getting pedestal values from various BPM systems as well as calibrating for the best possible settings. There are many learning tools experiences here at Cornell University.

The digital board has an Analog Devices TigerSHARC TS101 digital signal processor (DSP), a Xilinx Virtex-2 field programmable gate array (FPGA), 2 Mbytes of static RAM, and 512 kbytes of FLASH memory. The FPGA contains the logic for external communication, extraction of data encoded on the CESR timing signal, address decoding and bus sharing, and data acquisition control.[1]

There are three main systems that can connect to the digital board. The fast luminosity monitor (FLM) is used to count the rate of photons emitted from radiative bhabha events at the CLEO interaction point, the beam size monitor (BSM) is meant to measure the size and profile of the beam and the BPM. In Figure 1, one can see a hardware layout for these three different type of monitoring systems.

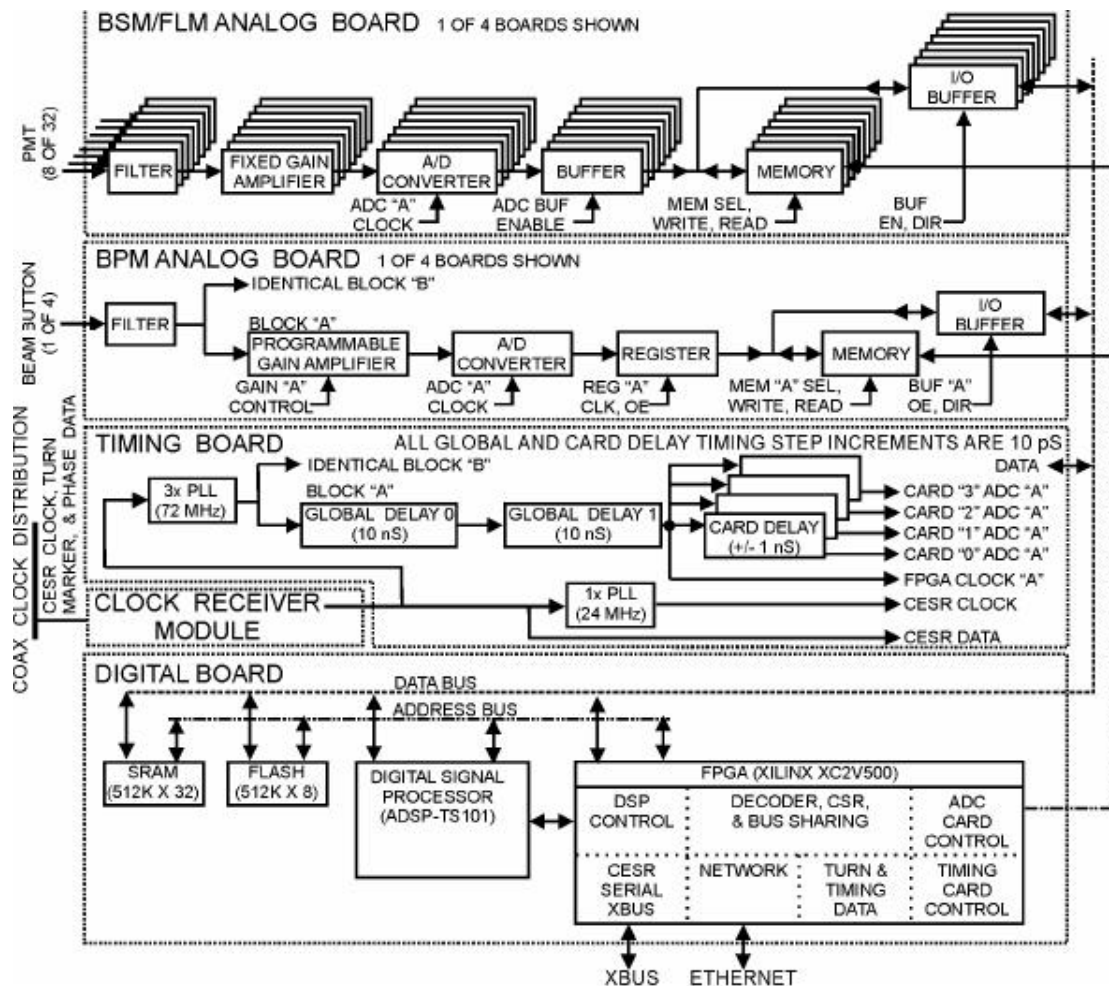


FIG. 1: Hardware layout for the BPM, BSM and FLM Systems.

In specific, the two pieces of electronics that have been primarily tested are the AD8369 (Digital Variable Gain Amplifier)[2] and AD9245 (Analog to Digital Converter)[3]. Currently, there are two test beam position monitor systems that are being tested with (one in room 101 and the other in lab 215). The room 101 BPM is fully equipped, whereas the lab BPM only has 1 card initially installed.

III. THE WORK

Increasing the gain on the DVGA may result with an increase in noise. Gain settings range from 0 to 14 on even intervals. These settings represent a range from -10 dB to 35 dB. For each setting, the input is translated by a factor. Table 1 represents each nominal

TABLE I: This illustrates what a table might look like.

Gain	Input 1	Nominal Factor
0	X	X / 2
2	X	X
4	X	X*2
6	X	X*4
8	X	X*8
10	X	X*16
12	X	X*32
14	X	X*64

factor with which the input data gets translated for each gain setting. Due to noise and a lack of linearity, these values are not what is actually seen through readouts. With this thought, tests were run on both test beam position monitors. A bulk of the tests ran this summer dealt with the effects of the root mean square (RMS) of the pedestal values; which is a measure of the noise in the system. Pedestal values are achieved when the DSP reads values from the BPM when there is no beam present. The RMS values have been tested as a function of gain, number of bunches read, number of channels selected, changes in global timing variants as well as a clock settings. The program "BPMV1" is primarily used in all tests. With this program, the user is able to change any of the variables used in testing.

IV. ACHIEVING RESULTS

Initial Tests

To see the dependency of noise level and linearity on gain settings, tests were conducted on each room 10 as a function of gain. The results show that the RMS values increase as the gain increases, and are different for the various channels.

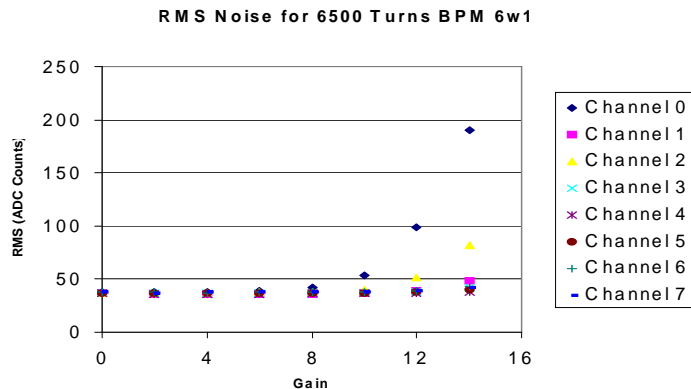


FIG. 2: Initial results found during the RMS versus Gain tests.

Clock Tests

After many tests, the next consideration is that the clock settings may be affecting the RMS readouts. The clock settings are a part of the timing card. The timing card generates two ADC clocks which set timing delays for when to "clock in" the data during a test. The clock settings are read in binary. The clock settings range from 0 to 127, which in binary reads from 0000000 to 1111111. Each binary registers different functions for the DSP. Bits 0 and 1 represent timing delays for the first channel in the card, bits 2 and 3 represent the second channel, bits 4 and 5 represent a memory write timing delay and bit 6 represents the master clock selected for the ADC clock [4]. Typically, a clock setting of 79 had been used as the "reference clock," however this clock may not be the best setting. Bad settings occur for various reasons. Primarily, the different delays of clocking in the data are used to adjust whether a values is being read at its peak. If the timing is off, then bad values are achieved, so the correct clocking times must be used.

In order to test which were the best clock settings for the BPM systems, the program "GETBPMCLOCK" was used. "GETBPMCLOCK" is a program that attempts to find the best clock settings by examining pedestal values produced over an amount of turns. When examining these pedestal values, a threshold set by the user is used to determine whether clock settings produced good values or bad values. Clocks that maintained all pedestal values within the threshold are kept, whereas the rest are disregarded. With the current program, tests had to be initialized after every run, which is very inefficient when trying to find the best of 128 different settings. At this point, the program was modified to automatically loop until the best clock settings are found. With this new "Burrellclock" program all parameters are read from an input file. After the initial settings are entered, the program enters an infinite loop, deleting "bad" clock settings after every loop. "Bad" clock settings were determined by setting a specific threshold whereby pedestal values read beyond this scale automatically meant bad clock settings. With a decrease in the amount of possible clock settings left, the number of turns increased as well (for a more precise scan).

With this new program, the DSP appeared to not find any good values for quite a large threshold (typical pedestal values ranged from +/- 300, however a threshold of +/- 1000 still showed no good clock settings). After further analysis of the data it was noticed that sometimes the initial value read would be very off, thus throwing out the clock setting. This problem was similar to one encountered before where the last value read was always very off. The previous solution was to have the DSP take one extra turn of data, but exclude the data on readback. So now 2 extra turns of data are taken, leaving out both the data of the first and last turns on readback. After this modification, it is found that there can be many "good" clock settings.

Initial Bunch Tests

The next set of tests were run to examine the difference of noise found when analyzing many bunches on a small amount of turns versus analyzing a few bunches on many turns. These tests showed that there are no significant differences in RMS values. The next set of tests were to see the effects of the global timing variables. Earlier, initial tests showed no significant changes when changing globTA and globTB, which are additional timing settings. In terms of figure 1, globTA and globTB are a combination of Global Delay 0 and Global Delay 1. A represents the timing delay for the first channel, and B represents the timing delay for the second channel; the delay settings are on an increment of 10 ps. A repeat of these tests confirmed the results.

More Bunch Testing

At this point, testing the RMS values as a function of the number of bunches selected was being considered. The setup for the tests were to check the RMS values on Gain 14, channel 0, with globTA = globTB = 800 testing 183 bunches, 175 bunches, 170 bunches, 140 bunches, etc. Here, rather abnormal RMS values were found. In some tests, the RMS values would be perfectly normal, however on others there would be strange dips. The way bunches were selected for each run had been random so far. First, an entire "train" (a series of 20 bunches) had been unselected only to find that there were bad RMS values.

Next, since there are two different lengths in trains, one of the 21-bunch trains were then unselected instead. Here, it was observed that there were no bad RMS values. Starting from the end of the bunches selected by the DSP, the last train of 20 bunches was systematically unselected. After doing this, every two bunches were then selected for each consecutive run. This test gave RMS charts for 21 bunches, 20 bunches, 19 bunches, 17 bunches all the way down to one and zero bunches selected. On every odd number of bunches not selected, there were no bad values, so then a next set of tests were run for every even number of bunches not selected. Here is where the problem is on a consistent basis; every even number of bunches not selected (from 20 to 0) showed a "dip" in the data followed by high RMS values. Also, on the odd number of bunches not selected, the RMS values were lower by a factor of 75 percent. For 0 bunches, the last 20 DSP bunches were

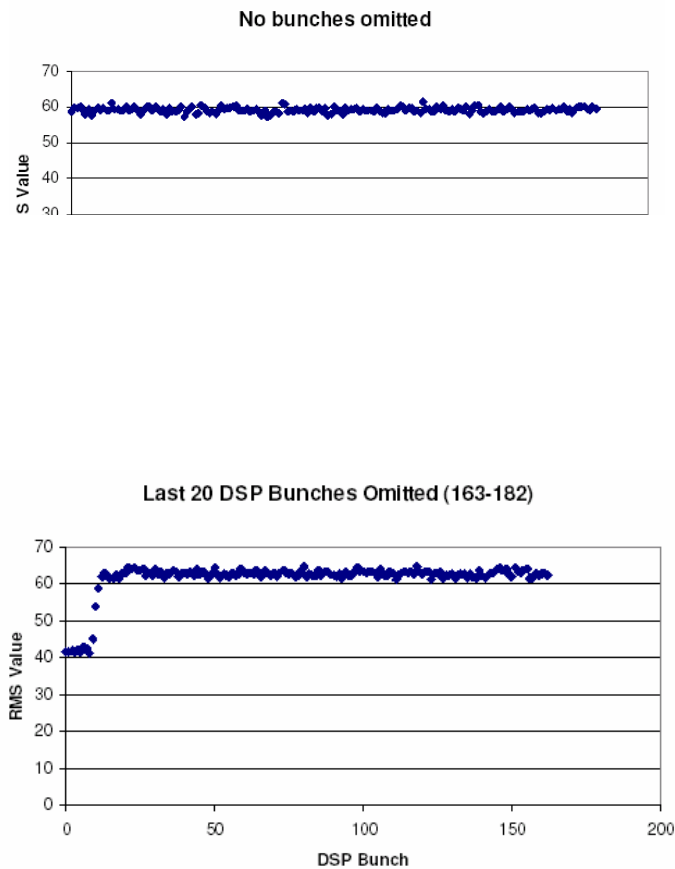


FIG. 4: Average RMS Values for every Bunch. (20 not selected)

Another variable that could affect the noise of our data are the cards themselves. In

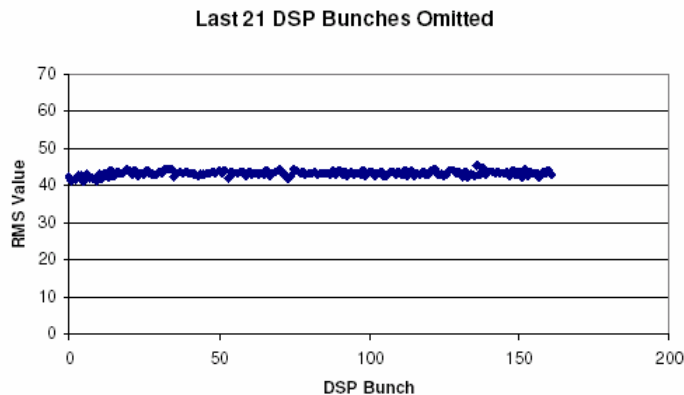


FIG. 5: Average RMS Values for every Bunch. (21 not selected)

the hardware layout, shown in figure 1, it shows that there are four different BPM boards connected to the BPM system, which is also connected to the timing board and digital board. Several experiments suggest that these six boards have the ability to interfere with each other, and after several tests we found this to be true. Initially, the room 101 BPM was fully equipped with functioning cards, however the BPM would still produce slightly higher RMS values than it should on the even channels. Previously, the Lab 215 card had been deemed a "good" card, so the first theory tested was that there was one bad card in

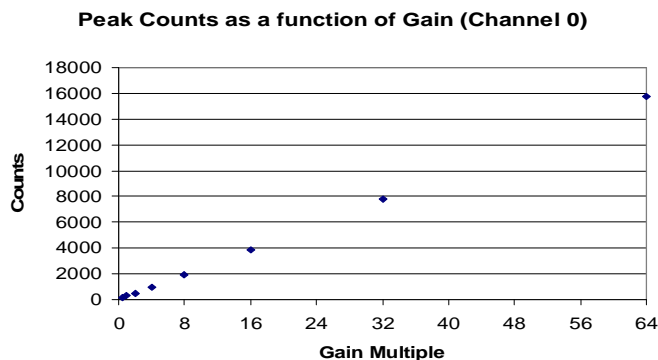


FIG. 6: Counts as a function of gain.

Figure 6 shows the results of the test as a function of gain. This test shows that the system is linear in respects to gain settings.

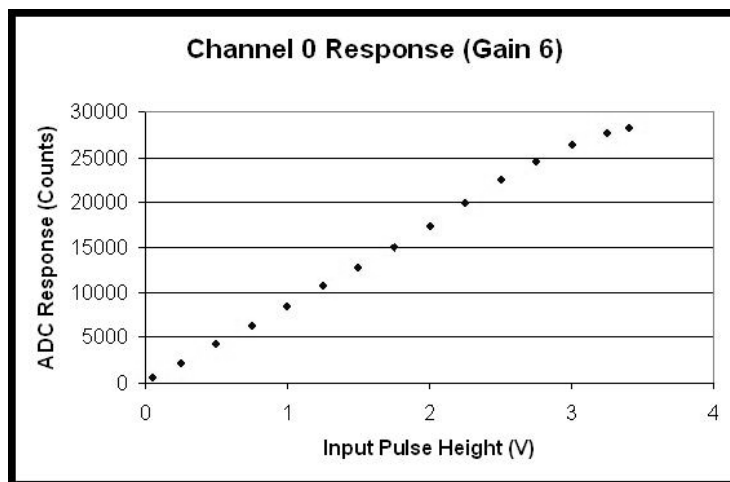


FIG. 7: Counts as a function of input pulse.

Figure 7 shows the results of the test as a function of increasing input pulse. This test shows that the readout is rather linear until the higher end due to saturation. Although the final few inputs aren't very linear, operation at this point isn't recommended due to digital errors encountered at such a high readout pulse. In short, the DVGA-ADC is able to digitize values from +32k to -32k, therefore if you increase the pulse any more at this point, the values become "saturated." When a value is too high saturated, the readout reads the value as very large, and only +/-32k can be shown.

V. SUMMARY, RESULTS AND CONCLUSION

Tests run this summer on the new BPM system have shown that there are both hardware bugs as well as digital bugs within the system. A complete characterization of the BPM system is not yet complete and further tests must still be run. Work done this summer has furthered the analysis of noise within the system and the linearity of testing. All results and those not found here can be found on the LEPP wiki webpage [5].

VI. ACKNOWLEDGMENTS

First and foremost, I would like to thank G. Bonvicini for giving me this opportunity to spend the summer of 2006 at Cornell, as well as all of the mentors and teachers that helped aid to my knowledge before coming. Also, I would like to thank the NSF for grant PHY-0353994, which allowed me to work here at Cornell University. I would like to thank Mark Palmer and Eugene Tanke for being such great mentors and friends during my stay. Last but not least, I would like to thank Rich Galik for his guidance and support throughout the entire summer. Without each one of these people, my work could not have been possible.

[1] M. Palmer et al. in the article "A BUNCH-BY-BUNCH AND TURN-BY-TURN INSTRUMENTATION HARDWARE UPGRADE FOR CESR-c. (In the Proceedings of the 2005 Par-

ticle Accelerator Conference).”

- [2] AD8369 Analog Devices Reference Manual (Digital Variable Gain Amplifier)
- [3] AD9245 Analog Devices Reference Manual (Analog to Digital Converter)
- [4] CESR BPM/BSM/FLM SYSTEM DIGITAL PROCESSOR BOARD
z:\crs\Cesr_BPM_BSM\Docs\DSP_Board_Programming_V5.doc
- [5] https://wiki.lepp.cornell.edu/lepp/bin/view/CESR/Bunch/BPM6W1Log_2006