TDIG CALIBRATION

JING LIU, ED. MICHAEL DEMARCO

1. Introduction

The STAR Time–of–Flight (TOF) system measures time intervals in order to calculate the velocity of a particle. This requires a very high timing accuracy. The Time-to-Digital-Converter (TDC) turns a time delay relative to the external clock (which runs at 40 MHz) into a number. There is a small interval of time, a bin, in which hits will return the same number. Unfortunately, as discussed later, the bins are often of unequal duration, or width. If uncorrected, these unequal bin widths would degrade the achievable timing resolution of the TOF system. In order to achieve the best possible resolution, this non-linearity requires an offline correction.

For the TOF, we will be acquiring data into all TDC bins asynchronously with respect to the TDC clock. The TDC data system consists of TDIG circuit boards, each of which have 3 High Performance TDCs mounted on them. On STAR, each TDIG relays time information to a buffer card, TCPU. 30 TCPUs the send the data to a THUB which then sends it to the Data Acquisition (DAQ) system. Each HPTDC has eight channels, and we must calibrate each channel.

The HPTDC uses tapped delay line architecture to perform its time-sampling function. The 40MHz input clock is multiplied on chip by a phase locked loop to 320MHz. This 3.125ns clock signal is fed through a 32-tap delay locked loop (DLL). The 97.66ps delay elements of the DLL are CMOS gates with voltage controlled delay. Voltage feedback from the DLL controller maintains a delay through all 32 delay elements that is equal to one 3.125ns clock period.

The DLL tap intervals are further subdivided by a 4-tap resistor-capacitor (RC) delay network. All four RC taps are sampled when an input occurs, and an interpolation determines which tap pair showed a clock transition during the sampling interval. This final subdivision results in bin widths of approximately 24.41ps. The arrival time of data pulses coming into the HPTDC are compared with how far the 320MHz clock has propagated through the DLL and RC networks: this position gives a time of arrival relative to the 320MHz clock, with a nominal resolution of 24.41ps. The 320MHz clock also drives a digital counter, and the combination of the counter value, the DLL value, and the interpolated 24.41ps RC data gives a time of arrival relative to the counter reset. [1]

The acquired data will be distorted by the three types of nonlinearities present in the HPTDC.
1. Unequal bin widths in the RC tapped delay lines.
2. Unequal bin widths in the DLL.
3. Noise coupling from the logic clock network (40MHz) within the chip into the sampling clock network (320MHz)

The RC nonlinearity is periodic over 4 bins, the DLL nonlinearity is periodic over 32bins, and the clock feed through nonlinearity is periodic over 1024 bins. In all cases, these effects are expected to be deterministic and periodic for each chip. As a result, these effects can be measured statistically and corrected for both on and off chip [1]. The cumulative effect if the non-linearity over all the

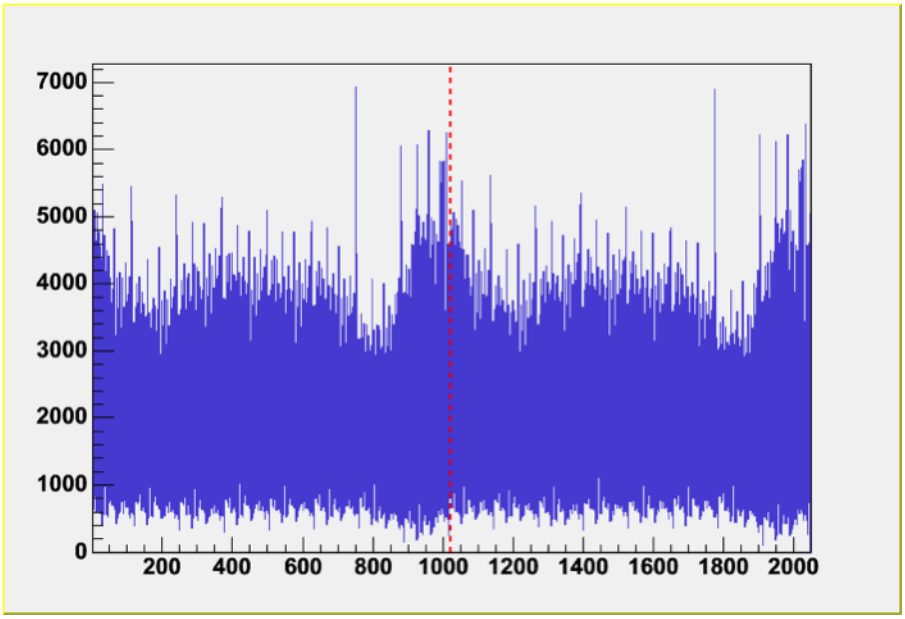


FIGURE 1. This plot shows typical results of a code density test.

bins is called the Integral Non-Linearity.

On chip, all 8 TDC channels use a common 32 tap DLL circuit and a 3-bit register for each tap adjusts the effective tap width. Each TDC channel uses a different 4-tap RC delay line, but there is one set of adjustment registers common to all 8 RC delay lines. Our higher level test (using random data into 2 channels with a constant delay) shows that the clock noise coupling dominates once the DLL and RC adjustments have been made to eliminate gross bin width variation. [2]

2. How to Measure INL

A standard technique for statistical bin width measurement is called a code density test: a pulse generator running asynchronously with respect to the TDC clock produces pulses that arrive with a uniform random distribution with respect to the TDC clock. If all TDC bins were the same width, then the probability of a pulse arriving in any bin would be the same as that for any other bin and a histogram of all bins for many input pulses would be flat However, if the time bins vary in width, the histogram bins for wider time bins will collect relatively more hits than the histogram bins for narrower time bins. Figure 1 is the result from a code density test. On the x-axis are bin numbers, with bin 1024 marked in red. Clearly, not all bins are the same width. The first particularly egregious spike occurs in bin 108, the last bin in the DLL delay tap, a common occurrence. This occurs because bin 108 is the last bin in the DLL delay tap. This effect is observed every 128 bins after bin 108. Also clear is the periodicity of the non-linearity over 1024 bins and, to a lesser extent, the periodicity over 128.

The nominal or expected bin width in the code density test is just the number of hits in each bin, if each bin had the same number of hits (the total number of inputs divided by the number of bins). For N total hits with $H(i)$ hits in the i^{th} bin, the INL correction required for the i^{th} bin is:

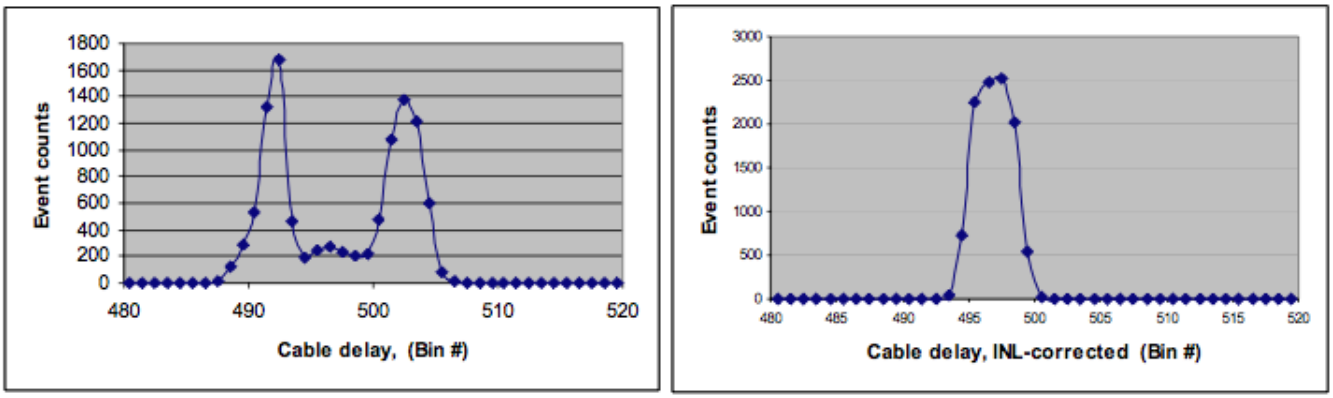


FIGURE 2. Cable Delay results with and without INL correction

- Expected Bin Width:

$$W_{expected} = \frac{1}{N}\sum_{j=1}^{N} H_j$$

- Actual Bin Width:

$$W(i) = \frac{H_i}{W_{expected}}$$

- Direct Non-Linearity:

$$DNL(i) = \frac{W_{expected} - W(i)}{W_{expected}}$$

- Integral Non-Linearity:

$$INL(i) = \sum_{j=1}^{i} DNL(i)$$

3. Cable Delay Test

The cable delay test provides a method of determining the time resolution of a TDIG board. The signal from pulse generator running asychonously with respect to the TDC clock is split and input separately to the channels of the HPTDC with a known time delay. In our setup, this function is performed by TCAL, discussed in the next section. By histogramming the measured time difference between channels for a single pulse, the time resolution, σ, can be approximated. Below are the results from a cable delay test without an INL correction and with an INL correction from J Schambach's paper [1]. Without an INL correction, two average values for the time difference are often measured and the time resolution is too large to be used for TOF; with the INL correction, typical time resolution is less than $40ps$, the requirement for TOF.

4. Rice Setup

All the TDIG boards are calibrated at the Rice University Bonner Lab using the cable delay test setup in Fig. 2.

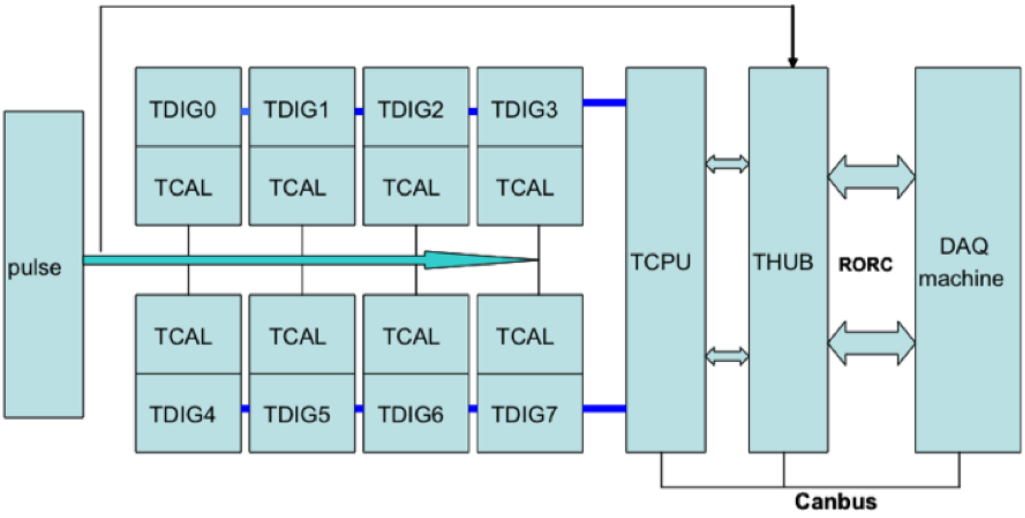
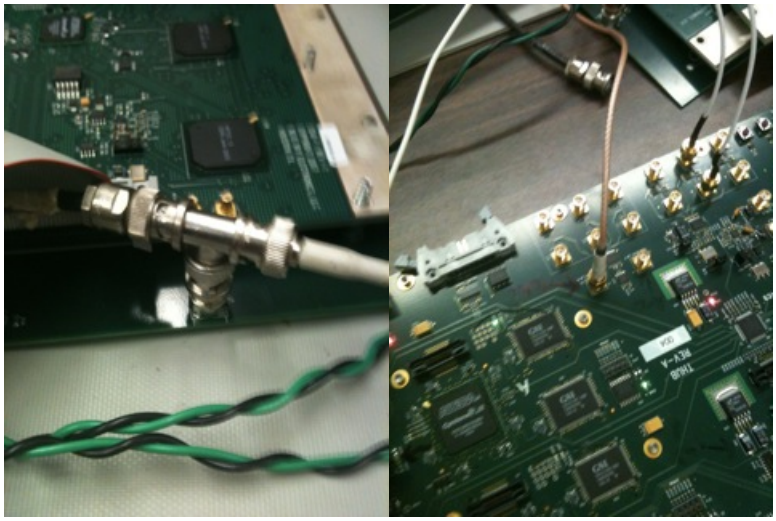


FIGURE 3



(A) The "T" connection to TCAL

(B) The pulse into the THUB

FIGURE 4

4.1. **Pulse Generator.** We use a +3V square wave output pulse with a $400ns$ width, $271\mu s$ period. So long as the period of the input and the bin width do not have a small common multiple, we have a good approximation of a random input signal. There is a problem with a $283\mu s$ period that is currently being investigated. Beyond that, the period is a compromise between the time it takes to collect a sufficient amount of data (6,000,000 pulses) and avoiding buffer overflow. If you do overload the buffers, pcanloop (discussed later) will throw a buffer error (QVRC).

4.2. **TCPU.** Another production board for the TOF system, the TCPU is connected to the TDIG boards via ribbon cables and to the THUB and the DAQ machine, labeled in green in figure 5.

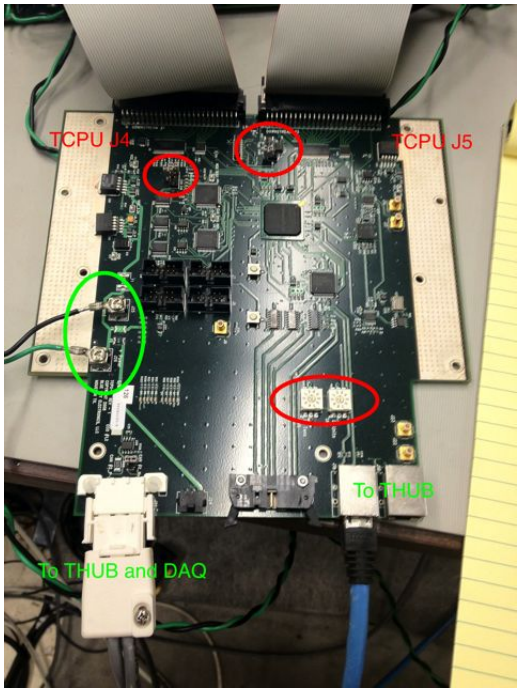


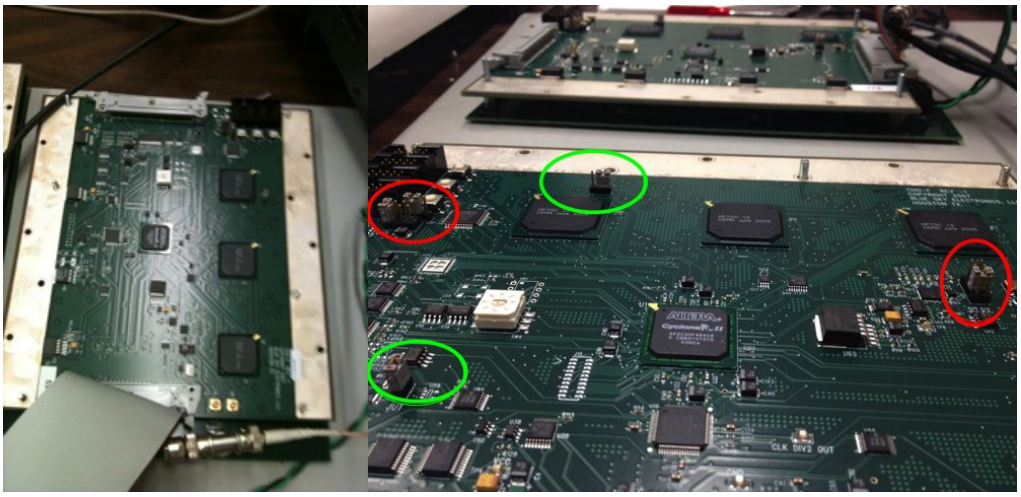
FIGURE 5. TCPU with appropriate connections

Also shown are the power connections (in green), the jumpers (circled in red), and two rotary dials (circled in red). The rotary dials denote the TCPU tray number on TOF and should both be set to zero for calibration. At the top of the photo are the two ribbon cable connections to the TDIG chains, corresponding to TCPU J4 and J5.[3]

4.3. **TCAL.** The TCAL was designed by the Bonner Lab at Rice University. It receives a pulse signal from the pulse generator and outputs 3 groups of 8 signals which are identical to the input pulse. In each group of 8 output pulses, each signal is delayed by about $63ns$ with respect to the previous signal, resulting in a $504ns$ delay in the 8th channel. This signal is then input to the 8 channels of the TDC on the HPTDC. With this 63ns delay, the timing crosstalk within the HPTDC TDC is no longer significant. The TCAL is powered by a LV power cord, and the TDIG is directly mounted onto it.

4.4. **TDIG.** These are the production boards for the TOF system. The setup can install and calibrate up to 8 TDIG boards in one run, though we typically only run 4 boards–2 on TCPU J4 and 2 on TCPU J5. When only one TDIG is on the test stand, TCPU J4 should be used.

The TDIG is connected with upstream and downstream ribbon cable to the TCPU and requires jumpers on JU1, JU2, JU3, JU10, J17, SW4[4]. The jumpers for the TDIG farthest from TCPU are shown in figure 6b. For the other TDIGs on each arm, the jumpers on JU1 should be removed (lower left green) and the Jumper on J17 should be moved to the other pair (upper left green). These two are circled in green on figure 6. The TDIGs attached to the TCPU J4 are TDIGs 0-3 (0 is always the TDIG farthest from TCPU, even if there is only one, and the rest are sequential), while those attached to J5 are numbered 4-7. On each TDIG is a rotary dial that should be set



(A) The TDIG mounted on TCAL

(B) Jumper Locations on the TDIG

FIGURE 6

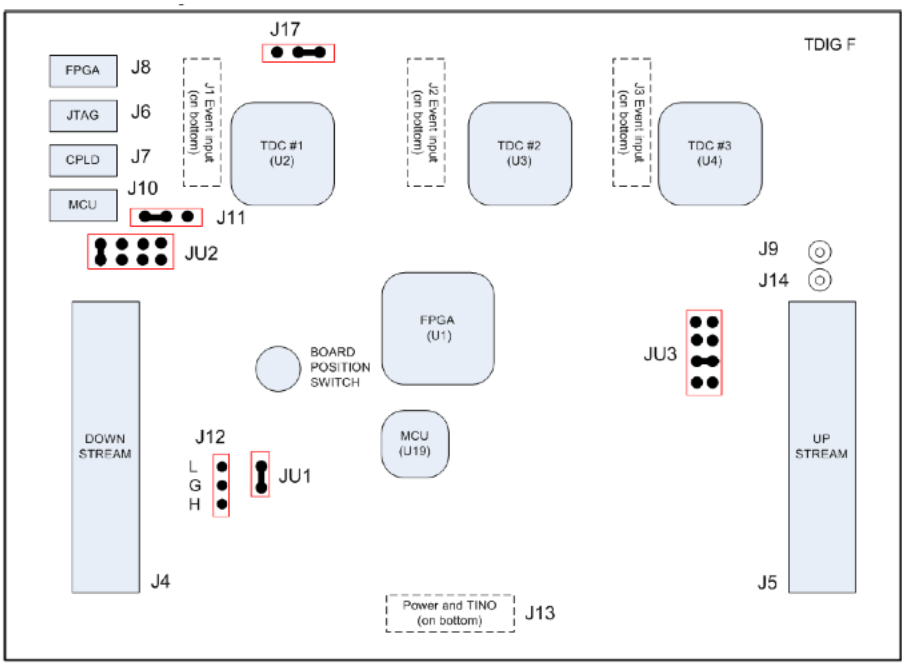
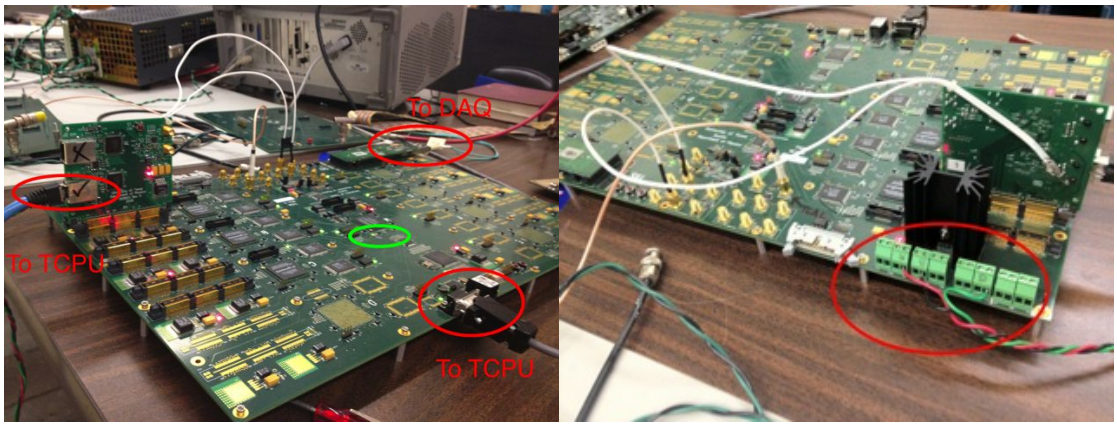


FIGURE 7. TDIG Schematic [4]

to the TDIG position number. TDIGs should be evenly distributed between the two arms of the TCPU. [4]

4.5. **THUB.** Our THUB is a prototype THUB board for the TOF system. Four THUBs of a later version are currently in use on STAR. Only an older version of the firmware can be used on



(A) THUB with connections. In the upper right are the green cords to the DAQ machine, while in the lower portion of the photo are the connections to TCPU

(B) Power inputs to THUB

FIGURE 8

this board, and the same version of firmware should be used for the TCPU. ***I want to double check that THUB firmware can be changed before I remove this statement. See the firmware guide associated with this manual for information regarding the firmware to use and installation instructions. THUB is connected to both TCPU via Canbus and to the DAQ machine via RORC, as shown in figure 8a. Once everything is turned on and flashing, a diagnostic light, circled in green in figure 8a should be flashing steadily.

4.6. **Power Supplies.** The TCAL and TCPU are both attached to a $4V$, $6A$ power supply. THUB, on the other hand, as shown in the photo, is attached to a $+5V$ $5A$(Red); a $+2.7V$, $.5A$ (Green); and a $-2.5V$, $5A$ (Black).

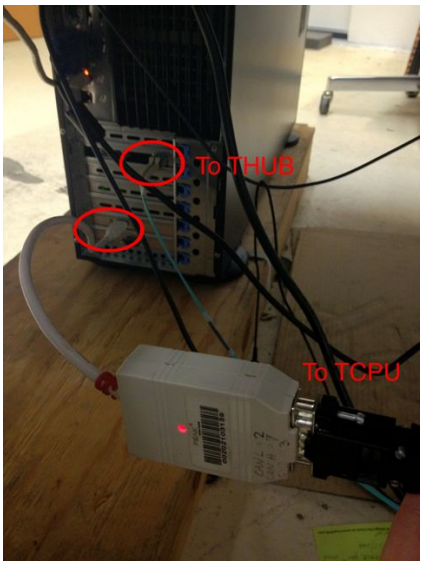
4.7. **DAQ Machine.** The DAQ machine is a Linux box, accessible at *toftest.rice.edu*. It must run Scientific Linux 4 or higher. Pcan is used to control the setup, and the driver must be installed on the computer. The linux box in the Bonner Lab currently has pcan installed, but if necessary, it can be downloaded from:

`http://www.peak-system.com/fileadmin/media/linux/index.htm`

The current account is the root account, though the calibration work will be moved to "tofr" or "teststand" in the future. The DAQ machine is connected to THUB via a fiber connection through the SIU card for the data stream, and connected to THUB and TCPU via a Can-Bus cable for control.

5. TAKING DATA

Before power-up, opening several x-terminals on the DAQ machine is recommended. We open 4 terminals, hereafter referred to as x_1, x_2, x_3, x_4. After logging into the root account, *root@toftest.rice.edu*, enter the following in x_1:



(A) Connections into the DAQ Machine

FIGURE 9

Entry	Response
cd pcan ./pcanloop	*Trying to open PCAN devices with BTR0BTR1=0x0014* *Device at /dev/pcan32: Hardware ID = 0xff* *pcanloop: driver version = Release_20080220_n*

This means the can-bus device is connected correctly. After set-up is complete, power needs to be turned on in the following order: THUB, TCPU, TDIG, after which the following will appear in x_1:

pcanloop: message received : 255 m s 0x00000407 4 0xff 0x00 0x00 0x00
pcanloop: message received : 255 m s 0x00000207 4 0xff 0x00 0x00 0x00
pcanloop: message received : 255 m e 0x041c0020 4 0xff 0x00 0x00 0x00

The first line means THUB is seen by Can-Bus, the second line means TCPU is seen by Can-Bus, the third line means 1 TDIG is seen by Can-Bus. Concerning the third line, 041c represents a TDIG number, while 0020 is the tray ID. Connecting more than one TDIG results in more lines like the third one, but with different TDIG numbers. See the Canbus manual for more information on commands and responses [5]

Then you can go to terminal x_2, and type the following commands line by line (statements after arrow are only comments):

./pc "m s 0x404 1 0x91"
./pc "m s 0x202 3 0xe 0x2 0xf" → connect TCPU to serdes
./pc "m s 0x402 2 0x91 0x81" → connect THUB to serdes
./pc "m s 0x404 1 0x91" → check the serdes connection
./pc "m s 0x402 2 0x80 0x3" → turn on trigger

You should expect to see the following in terminal x_1:

```
pcanloop: message assembled: 255 m s 0x00000404 1 0x91
pcanloop: message received : 255 m s 0x00000405 1 0x80
pcanloop: message assembled: 255 m s 0x00000202 3 0x0e 0x02 0x0f
pcanloop: message received : 255 m s 0x00000203 2 0x0e 0x00
pcanloop: message assembled: 255 m s 0x00000402 2 0x91 0x81
pcanloop: message received : 255 m s 0x00000403 2 0x91 0x00
pcanloop: message assembled: 255 m s 0x00000404 1 0x91
pcanloop: message received : 255 m s 0x00000405 1 0x81
pcanloop: message assembled: 255 m s 0x00000402 2 0x80 0x03
pcanloop: message received : 255 m s 0x00000403 2 0x80 0x00
```

Rarely, the TCPU will need to be reset. If the responses to the TDIG commands are grossly different from the above, type:

```
./pc "m s 0x202 5 0xe 0xe 0x10 0xe 0x0"
```

and repeat the previous command sequence. Now the whole setup is ready to take data for calibration. Go to x_3 and go to the directory /root/ddl/tofrorc/ __vers.5.2.1/Linux and type the following command:

```
.rorc_receive -r 3 -x 0 -h rawdatatdig-n0-n1-n4-n5.dat -e 6000000
```

where $n(i)$ is the serial number of the tdig in position i. We use 'xxx' to replace unused TDIG locations. Data will begin and the data will be saved at /data/rawdata/. After finishing 6,000,000 pulses (normally about half an hour if you run pulse of 283us period), the data taking will stop automatically. To check if the data is reasonable during or after the data taken, you can go to terminal x_4, and type: *od -t x4 tdig-n1-n2-n3-n4.dat* to check. You should see five columns of hex numbers being typed out on the screen; hit cntrl–c to quit. The data is saved in binary and you can see all the separators and timestamps, as well as geographical words; trigger words depend on the configuration.

6. Data Analysis

After the data run is complete, we use an analysis code to produce the INL data and some QA plots. The source codes are saved in /data/analysis. There is a shell script to run the various codes in the analysis directory. Just type:

```
./analysis.sh tdig-n0-n1-n4-n5.dat
```

Th script will run an analysis of the INL and a cable delay test. The INL analysis proceeds as described in section 2, while the cable delay test measures the delay of the signal across several channels. The results are stored in INL files and three kinds of plots in the *data/result* directory on the DAQ machine. As of May 2012, the results are stored in *data/result/mdm*, though this should be changed back to just *data/result* in the future.

The INL plot is simply the INL curves for all channels of a HPTDC. The script itself produces a single graphic with the three plots for each HPTDC on the TDIG.

The *checkchannel* (Fig. 10 B) plot provides a way to check the function of TCAL. The numbers show the number of data words per channel. The numbers should be nearly identical. If a channel is missing, or worse, has extra words, then the output of particular channels, and not that of the whole TDIG, is being distorted, which indicates a TCAL malfunction. Over the course of calibrating the boards, four TCALs have failed, like due to the mechanical stress from mounting and unmounting hundreds of TDIGs.

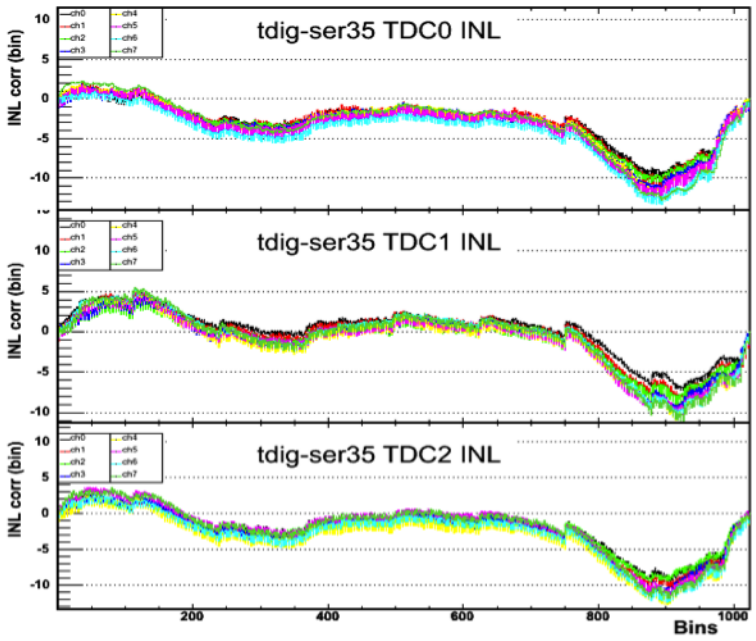


FIGURE 10. The INL Plot.

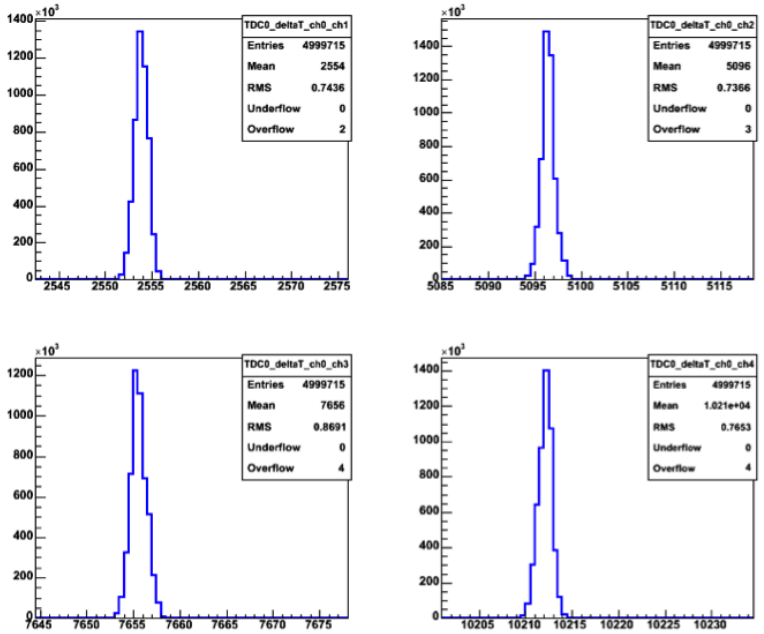
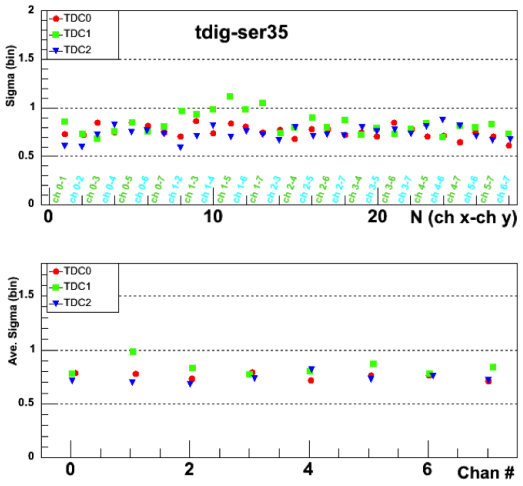
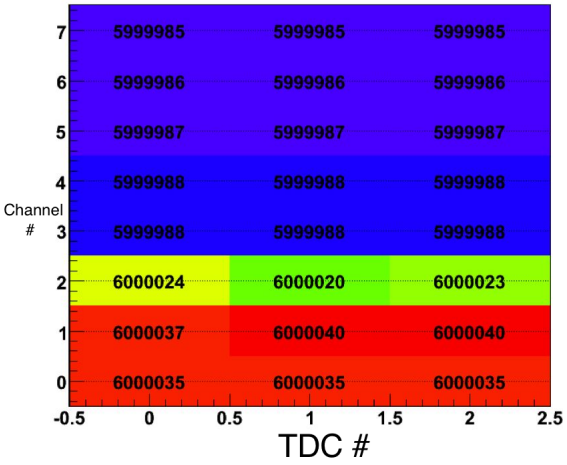


FIGURE 11. These are four of the 28 time resolution plots produced from the cable delay test.



(A) Here is the summary of the cable delay test



(B) This plot shows the number of words per TDC channel

FIGURE 12

The summary of the cable delay test is very important, as this is the benchmark as to whether or not a TDIG will be used. We require a resolution of less than $40ps$ for all the channels in order to meet the $100ps$ total resolution that TOF requires in order to meet its physics goals. To be on the safe side, we usually reject boards with resolution over $35ps$. Without an INL correction, the resolution is over $100ps$, which is why we calibrate every board. If the TCAL is malfunctioning, the resolution will go up about $4-5ps$. Also, sometimes the time resolution plots will have two peaks instead of one. This will usually add $5-10ps$ to the resolution, though we do not reject twin-peaked boards if the resolution is less than $35ps$.

REFERENCES

[1] J Schambach for STAR TOF Group. *Proposed STAR Time of Flight Readout Electronics and DAQ.* Computing in High Energy and Nuclear Physics, LaJolla, California, 2003.

[2] J Chistiansen. *HPTDC Manual.*`Ineedaplacetoputthissinceitsnotonline`

[3] Lloyd Bridges, William Burton, Jo Schambach. *TCPU Manual.*`Ineedaplacetoputthissinceitsnotonline`

[4] Lloyd Bridges, William Burton, Jo Schambach. *TDIG Rev F Engineering and User's Manual Vers. 4.0a.* `http://www.star.bnl.gov/public/tof/electronics/TDIG_Engineering_Manual_rev_4_0a.pdf`

[5] Kennington. *TDIGCanbus High Level Protocol Vers. 1.2.* `http://www.star.bnl.gov/public/tof/electronics/TCPU_CANbus_HLP_v1_2.pdf`

[6] *TOF Data Fiber Format* `http://www.star.bnl.gov/public/tof/electronics/TOF_Fiber_Data_Format_v2_4.pdf`