Tdc_spec002.doc June 8, 00 Bill Rawnsley

EPICS/VME TDC Program Specifications

Time Structure and Data Array

The repetition rate of the beam bunches is 11.6 MHz, see figure 1. The RFQ operates at 35 MHz and the Linac at 105 MHz. One would like to be able to plot at least two bunches so that the distance between them gives a quick check of the time scale as it is known to be 1/11.6 MHz = 86 ns. This requires two full cycles of the 11.6 MHz to be displayed, i.e. 172 ns. A chopper system will be installed in the MEBT. Its usual mode will be to eliminate the small amount of beam in the 35 MHz buckets on either side of the main bunch. In this mode, a TDC full scale of 200 ns should be about right. The second mode of the chopper also eliminates every second main bunch. In this mode, a full scale of 400 ns should be suitable.

The bin size of the TDC is 1/(10 MHz * 2^11) = 48.828125 ps. In 400 ns there are 8192 bins. If each bin could hold up to 2^24 = 16,777,216 hits then three bytes per bin would be needed. So the entire data would be 3 * 8192 = 24,576 bytes, i.e. 24 kB.

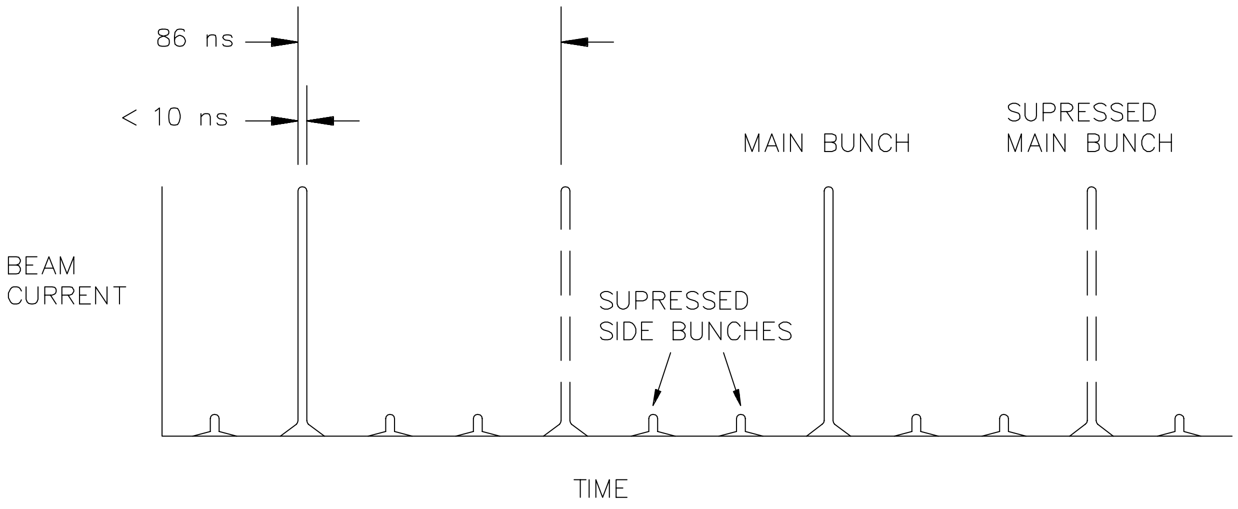


Figure 1. The time structure of the beam.

TDC Timing Modes

Figure 2 shows a block diagram of the setup. The TDC must be operated in its “positive only” mode.. In this mode, triggers on channels 1 to 7 are ignored until after a hit is sensed on channel 8. The particle sensor signal is connected to channel 8 so that once armed, the TDC will wait for a particle event to occur before requiring servicing. The RF stop signal is connected to channel 0 and in the normal chopper mode will receive a hit within 172 ns. The minimum “positive only” time is 4 ns so the measurable span is actually 168 ns. In the second chopper mode, the RF signal will be divided down to give stops spaced 344 ns apart.

The particle hit rate is small compared to the RF repetition rate so in this scheme the TDC will wait for a particle event through many RF cycles without needing servicing and will have very little dead time.

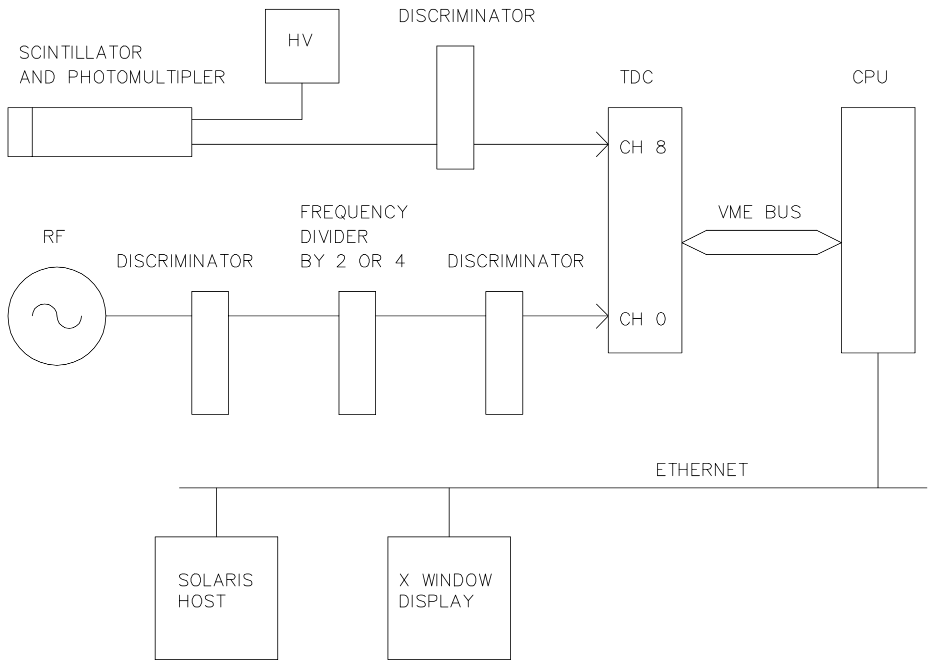


Figure 2. A block diagram of the setup.

Unfortunately this scheme prevents channels 1 to 7 to be used so multiplexing is not possible. This is the same problem that we have with the LeCroy 2228A Camac TDC's used on beamline 1 and 4. It seems that the TDC's are designed for time of flight experiments where one has a single start and many stops but this is not the case here. The Highland TDC does have programmable microengine in it for processing events so it is possible that this could be rectified but I have no plans to do this right now.

The TDC has two timing modes called standard and timestamp mode. In timestamp mode, the time reported for channel 0 will be the positive time difference between a hit on channel 8 and the following hit on channel 0. If this time difference is used as the X axis of a display, time will be displayed "backwards", i.e. events on the left side of the display will have occurred after events on the right side. This is typical of using TDC's this way and the operators are used to it, however, it should probably be reversed in the display program.

Histogram

When data collection is enabled, the histogram display should be continuously updated. An update rate of few times per second, if possible, should suffice since it is expected to take a couple of minutes to collect a clean histogram. It has been found very useful to have the date, time and device stamped on the plot

Data collection need not be stopped while the data is being read back by the host system. When the stop button is pushed, however, the data collection should be stopped and then the full array data should be read back by the host and the histogram updated. This will put the displayed histogram data in sync with the CPU array data after data collection has been stopped.

If the user pushes the SAVE or PRINT button while data collection is in progress, data collection will not be paused and the operation will proceed using the data currently available in the host's memory. It will be up to the user to stop data collection before using these commands if that is what is desired.

Rate Display

Two rates should be displayed; the single event and double event rates. They do not need to be very accurate as they are meant as guide to allow the operator to tune the accelerator to obtain a useful count rate from the beam monitor. The update rate need not be very fast, either. One update per second should be sufficient. The expected single count rate will be from 0 to about 100 kHz. The double event rate will only be significant at very high count rates. The rates will be for “live” time only, not real time. Live time is the time that the TDC is actually waiting for a particle event and does not include dead time when the TDC is being serviced.

About once per second, as determined from a real time clock in the Solaris host, the rates could be read and displayed. On startup, the VME CPU would clear the rate counters, start a timer and begin data collection (if collection is enabled). Whenever a particle hit occurs it would increment the counters appropriately and record the timer. When the Solaris host wants to display the rates, it would stop the data collection, read the counters and the time recorded for the last hit. The rates would then be calculated by dividing the counters and by the recorded (elapsed) time. After a read by the host, the VME CPU would clear the counters, clear and restart the timer and restart the data collection (if collection is enabled).

VME Interrupt Action

The VME CPU should write the TDC interrupt mask to enable an interrupt to be generated by a hit on channel 0. Using an interrupt will prevent wasted VME bus cycles which would occur if polling were used instead. The interrupt routine should increment the rate counters appropriately and the recorded time. It should read the TDC time registers for channel 0 and then increment the appropriate bin count in the histogram array. Any counts beyond the last bin should be counted in the last bin rather than be discarded. The TDC must then be re-armed.

There CPU should wait some minimum amount of time before servicing an interrupt to prevent hogging the VME bus if a very high hit rate occurs.

Horizontal Scaling

The horizontal axis of the histogram should be displayed in ns with time in the normal direction (later events to the right). The default scale should be 0 to 400 ns. The operator should be able to set the minimum and maximum values, however, so that the fine structure of a bunch can be expanded and examined in detail. Parsing should be done to insure that the user inputs values between 0 and 400 ns and that Xmin < Xmax.

Vertical Scaling

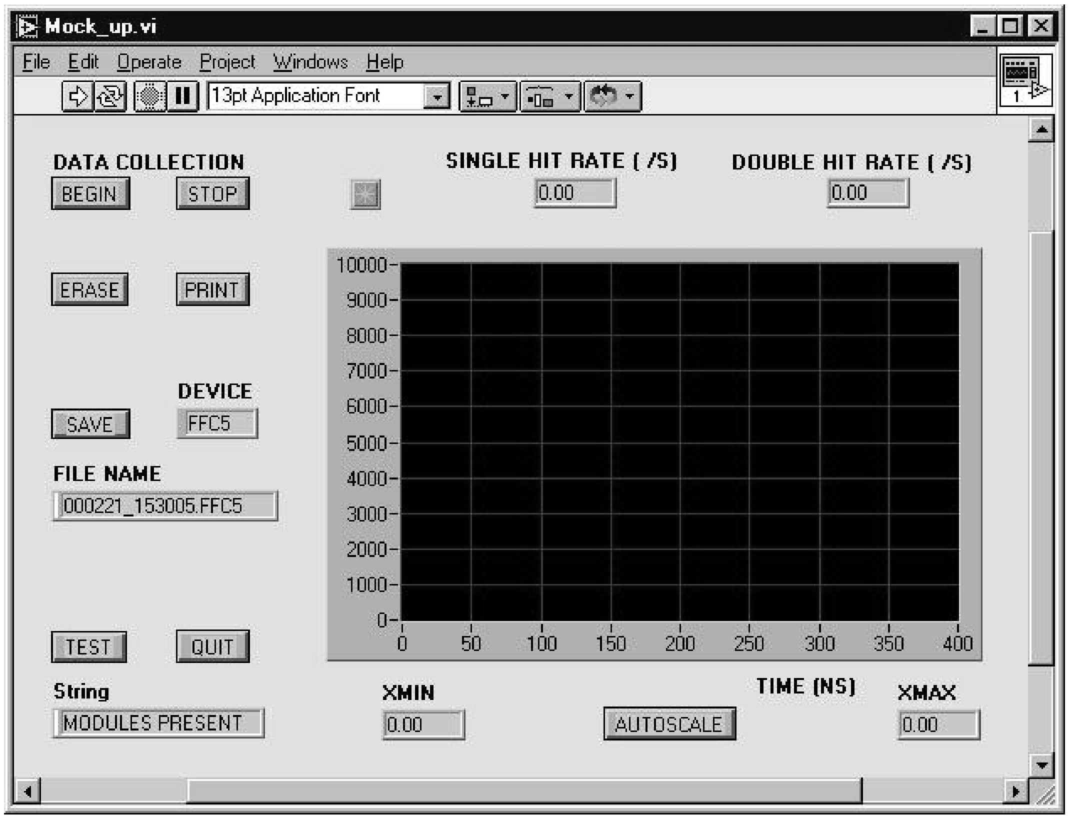
The vertical scale should be autoscaling, i.e. the peak count should be about 80% of full scale. The scale should be readjusted every time the autoscale button is clicked. TDC’s often put a lot of extraneous counts into bin zero so it, and the last bin, would have to be excluded from the autoscaling routine

User Interface

Figure 3 shows a mock up (done in Labview) of what the user interface might look like.

Commands

• **Test**	self-test the CPU and TDC modules and show the results
• **Source**	(future possibility only) select TDC channel 0 to 7 (default 0)
• **Erase**	clear the histogram data kept in the CPU module
• **Begin**	start data collection
• **Stop**	stop data collection (default is stopped)
• **Autoscale**	autoscales the vertical axis
• **Xmin**	set the minimum x value for the display graph (default 0 ns, floating point)
• **Xmax**	set the maximum x value for the display graph (default 400 ns, floating point)
• **Save**	save the data to a file with the date and time as the name and the device as the extension
• **Device**	one to four letters to describe the monitor in use
• **Print**	print the plot with date and time stamp
• **Quit**	stop data collection, end the program



SINGLE HIT RATE (/S) 0.00
DOUBLE HIT RATE (/S) 0.00
DEVICE FFC5
FILE NAME 000221_153005.FFC5
String MODULES PRESENT
XMIN 0.00
XMAX 0.00

Figure 3. A mock up of the user interface.

Command Details

Test
This function should simply report that the CPU and TDC modules are present. It will be enough to read back the module ID's. The modules do have very extensive self test routines built in, but they are not required to be run for this function.

Source
(future possibility only). This function should select which of the eight inputs will be used.

Erase
This function should simply set the data array to all zeroes.

Begin
This function allows the data collection to proceed.

Stop
This stops the data collection.

Autoscale
This function autoscales the vertical axis of the histogram.

Xmin
Sets Xmin.

Xmax
Sets Xmax.

Device
A text box to allow the user to enter one to four letters to describe the monitor that is in use, such as ffc5 for Fast Faraday Cup 5. This should be parsed to prevent illegal characters as it will be a file name extension.

Save
Saves the last displayed histogram data to a file in a folder on the Solaris host. Typically a folder such as isacdata is set up for this purpose. The file name should be the date and time as YYMMDD_HHMMSS. A one to four letter extension is required and it is input by the user by a text box. An example file name is 000209_182030.ffc5.

Print
This prints out the last displayed histogram. This may not be needed if this can be done using the system pull down menu functions. It has been found very useful to have the date and time stamped on the printout and possibly the device as entered by the user.

Quit
This function should gracefully end the program. It should stop data collection and place the CPU and TDC into standby states where they use as few VME bus cycles as possible so as to allow efficient use in a shared crate.