

The F1 TDC Chip in Synchronous Mode

In synchronous mode, the F1 chip must be programmed to internally reset itself before the 16-bit counter rolls over; otherwise the time measurements get corrupted. So the roll-over count (i.e. maximum count on any channel) is always $< 65,536$. (All chips / boards in the system are synchronized with the 'sync_reset' signal. The common system clock assures that resets in each F1 chip will always occur at the same time.) The reset period (in the documentation) is called the 'tframe' value, and the program we wrote to compute the F1 chip configuration parameters tries to make 'tframe' as large as possible. Because the internals of the F1 use integer multiplies & divides (for the PLL), not all values for 'tframe' are realizable. Below is an example of calculating the roll-over point.

Suppose the user wants a bin size of 0.112 ns. The program I am using comes up with the following F1 parameters (assuming a 40 MHz system clock):

refcnt = 290
refclkdiv = 128
hsdiv = 188

Then: $tframe = 25 * (refcnt + 2) = 7300$ ns,

$binsize = (25/152) * (refclkdiv / hsdiv) = 0.111982083$ ns (actual bin size).

The actual full range is $65536 * binsize = 7338.857783$ ns ($> tframe$).

The roll-over point (for these parameters) should be: $tframe / binsize$ (actual) = 65189 (i.e. $toffset = 65536 * (tframe / fullrange) = 65189$).

You can inspect the cfg file you are using to extract the above parameters. The 16 F1 chip registers (0-15) are listed in a single line (1 line per chip). The registers are described in the User Manual, or the F1 chip data sheet. Line 9 of the configuration file contains the roll-over count (toffset) corresponding to the chip parameters. D. Abbott has a cfg file display/decode routine in the library he created.

Note that this reset after tframe also affects the roll-over point for the **9-bit** trigger time reported in the header words. By the same calculation, the trigger should roll over at 509 rather than 511. The trigger time is used to be sure that data assembled from multiple chips (and boards) are from the same trigger. Differences of 1 count are acceptable (and expected). In the above case, chips reporting trigger times of 509 and 0 would be consistent. The trigger time represents the beginning of the data window.

Determining the Time Order of Hits

Consider two F1 TDC hits with recorded times (counts) t_1 and t_2 that are associated with a trigger. These may occur in the same channel, or different channels, or in different modules. It is clear from the above discussion that if a reset occurs between the arrival of the hits, the earlier hit can have a larger count than the later one. We can use a restriction on the F1 configuration values to resolve this problem. The F1 chip requires that the programmed trigger window width have a maximum value:

$$\text{window width} < 40\% * tframe.$$

Translating to counts,

$$\frac{\text{window width}}{\text{binsize}} < 40\% * \frac{\text{tframe}}{\text{binsize}} = 40\% * \text{toffset}.$$

Since offset is always close to 65,536 (by our maximization of tframe), we can use

$$\text{max count difference} \approx 26,500.$$

Any hits associated with the same trigger must have an actual difference of less than 26,500 counts. If $|t_1 - t_2| > 26,500$ a reset occurred between the hits and the values must be corrected.

Let Δt_{12} be the actual (signed) time difference between TDC hits 1 and 2 in counts.

$$\text{If } |t_1 - t_2| < 26500: \quad \Delta t_{12} = t_1 - t_2$$

$$\begin{aligned} \text{otherwise:} \quad \Delta t_{12} &= (t_1 - \text{toffset}) - t_2, \quad \text{for } t_1 > t_2 \\ &= t_1 - (t_2 - \text{toffset}), \quad \text{for } t_2 > t_1 \end{aligned}$$