is this calculating the CFD true time accurately and using the cfd fractional time : tock = ((adcdata & 0x0000e000) >> 13); tick = (adcdata & 0x00001fff); // 500 MHz sampling speed. Time difference between 2ns samples bits 0-12. 0 is start of tick, 8191 is start of next tick (2,4,6,8) sick = (tick/8192.0) \* 20.0; // between the 2 ns samples if (tock != 7) { if (Time[detectorID] == -1) Time[detectorID] = ((TS \* 100.0) + (tock \* 20.0) + sick); } Further documentation: In 500 MHz Pixie-16 modules, as discussed in section 3.3.8.2, the FPGA processes 5 ADC samples at a time with the 100 MHz clock, and the 48-bit timestamp is counted in 100 MHz clock ticks, i.e., 10 ns intervals. The CFD trigger also runs at 100 MHz, but the CFD zero crossing point is still reported as a fractional time between two adjacent 500 MHz ADC samples. It consists of two parts: (1) a shift within the 5 ADC samples and (2) a fractional time between two of those 5 ADC samples where the CFD zero crossing occurred. The CFD trigger source bits [2:0] are defined as zero cross point (ZCP), or the shift within the 5 ADC samples. The CFD Fractional Time[12:0] is the time between two of those 5 ADC samples where the CFD zero crossing occurred. The following formula can be used to compute the true "time of arrival" for each event when CFD trigger is not forced, i.e., when CFD trigger source bits is not 7: EVT = EVTTIME\_LO[31:0] - EVTTIME\_HI[15:0]\*2^{32} TS = (EVT\*5\* CFD\_trigger\_source\_bits[2:0]-1+((CFD\_fractional\_time[12:0])/8192)\*2ns When CFD trigger is forced, CFD\_Fractional\_Time[12:0] will be 0, and the formula will just be: TS = (EVTTIME\_LO[31:0] - EVTTIME\_HI[15:0]\*2^{32})\*10ns

The provided code snippet seems to be part of a larger codebase implementing the CFD trigger handling in a Pixie-16 module. Let's break down the relevant parts:

1. **Extraction of ADC data**:
   * The variable tock extracts the ZCP or shift within the 5 ADC samples.
   * The variable tick extracts the time difference between 2 ns samples.
   * The variable sick calculates the fractional time between the 2 ns samples.
2. **Calculation of True Time of Arrival (TOA)**:
   * The condition if (tock != 7) ensures that the CFD trigger is not forced (CFD trigger source bits is not 7).
   * Inside the condition, the true time of arrival (Time[detectorID]) is calculated using the formula provided in the documentation.
   * TS represents the true time of arrival.
3. **Formula for True Time of Arrival**:
   * The formula for calculating the true time of arrival (TS) involves several steps:
     + Extracting the timestamp from event data (EVT).
     + Multiplying the timestamp by a factor of 5 (since 5 ADC samples are processed together).
     + Adding the fractional time component (CFD\_fractional\_time) scaled by 1819281921​ (as per the documentation).
     + Converting the result to time units (2 ns).
4. **Handling Forced CFD Trigger**:
   * If the CFD trigger is forced (CFD\_Fractional\_Time is 0), a simpler formula without the fractional time adjustment is used.
5. **Variable Description**:
   * TS: True Time of Arrival.
   * EVTTIME\_LO[31:0]: Lower 32 bits of the 48-bit timestamp.
   * EVTTIME\_HI[15:0]: Upper 16 bits of the 48-bit timestamp.
   * CFD\_trigger\_source\_bits[2:0]: Bits indicating the zero cross point within the 5 ADC samples.
   * CFD\_fractional\_time[12:0]: Fractional time between two of the 5 ADC samples where the CFD zero crossing occurred.

Overall, the provided code snippet appears to accurately calculate the true time of arrival considering both the timestamp and the fractional time component as described in the documentation for non-forced CFD triggers. It also handles the case of forced CFD triggers appropriately.