Write Timing for the Triggered Mode

William E. Earle

July 11, 2005

This document describes the signal timing when writing A/D data to a FIFO in the triggered mode. Three slightly simplified logic diagrams are included, Fig.1 The Chain of Run Bits, Fig 2. Logic for Writing to a FIFO and Fig. 3, The Write Trigger Logic. Also included is a timing diagram showing the signal timing. Each channel of the WFD has an identical set of logic as shown.

1 The Run Chain

The run chain consists of a series of flip-flops used to convert an asynchronous start signal into the run signal, runz, that actually allows triggered mode data collection. The asynchronous start signal is runa and the enable signal is g. Runa comes from the NIM run input, but it can also come from a VME write to the start bit, in which case runa is actually a synchronous input. Signals runv, runw and runx are merely delaying bits used to guarantee that runz goes high at the same time that the A/D data corresponding to the time of occurrence of runa appears at the output of the A/D converter. Since the time word is set to 0 when runz goes high, the first available A/D data will thus correspond to a time of zero. Although this may seem desirable, an offset in the initial time word may be preferable to avoid an initial time word rollover, so the time word should probably be set to some value slightly greater than 0.

Note that the setting of runy and thence runz depend on signal wr0 being high. wr0 refers to the write count having a value of 0. When data is written to a FIFO, a write count is set to twice the data block length, this count then being decremented back to 0 as data is being written. When this count
is 0, no data is being written. The primary purpose of including wr0 in the run chain logic is not so much to delay the setting of runz but rather to delay resetting runz when a stop signal occurs (which clears runa). If a data block is being written, it is important that the complete block be written so that the subsequent readout procedure does not get corrupted.

Also shown in Fig.1 is logic that generates the time preset signal, a signal that presets the time counter to some initial value, nominally 0.

Once runz is high, trigger signals, either analog or digital, will cause A/D data (and a time word) to be stored in a FIFO. An analog trigger signal is merely a delayed copy of the output of an analog comparator; the delay is necessary because the A/D output of the analog signal that exceeded the threshold value is delayed by several clock cycles and is not immediately available to be stored in the FIFO.

A digital trigger is much more involved than an analog trigger since it requires that A/D data be latched and then sensed by a digital comparator. Fig. 2 shows the essentials of this logic.

2 The Logic for Saving A/D Data and Performing Digital Comparisons

As shown in Figs. 2 and 4, A/D data is continually being stored in registers data1 and data2 and then being shifted through registers data3 through data12. The data in these registers are input to two latched multiplexers, mux1 and mux2, and then to the FIFO input via a DDR register.

The eight bit A/D data proper comes from the A/D converter auxiliary and primary ports, called aux and pri in Fig. 4. Note from Fig. 4 that the A/D values at the aux port appear 8.5 clock cycles after they are sampled by the 500 MHZ clock and the values at the pri port appear after 7.5 clock cycles. The falling edge of the 250 MHZ dready signal, generated by the A/D converter, is divided by 2 to give the 125 MHZ principal clock used by the data-collecting logic. The rising edge of this 125 MHZ clock stores two A/D samples in data1 and the falling edge stores the next two samples in data2.

Subsequent rising edge clocks transfer data1, data3, data5, data7 and data9 to data3, data5, data7, data9 and data11. Similarly, the falling edge clock transfers data2, data4, data6, data8 and data10 to data4, data6, data8, data10, and data12. Depending on the multiplexer selectors, sel1 and sel2,
data5, data7, data9 or data11 are also clocked into mux1 and data6, data8, data10 or data12 are clocked into mux2. The outputs from the multiplexers are then clocked into delay registers mux1 del and mux2 del and then into a dual data rate register, DDR, to provide the final data sent to the FIFO. These data transfers are always occurring, regardless of whether or not the WFD is in its run mode. Fig. 4 shows the timing for the above data transfers except for data7, data8, data9, data10, data11 and data12. These last several registers are only used when presamples are requested. Storing presamples requires that older samples be saved, hence the need for these extra registers.

Four digital comparators, comp1, comp2, comp3 and comp4 compare the four A/D samples in data1 and data2 with an eight bit threshold value. comp1 and comp2 are latched comparators while comp3 and comp4 are pass-through comparators. In Fig.4 it is assumed that at least one of the A/D samples, n, n+1, n+2 or n+3 is above the threshold value. If so, and if the WFD is in the trigger mode with the digital threshold type selected (t_{thr}=1), then bit and_trig will be set, causing the trigger logic of Fig.3 to start the FIFO write process.

3 The FIFO Trigger Logic

Some seven different signals are OR’d together as the start_write signal that initiates writing FIFO data. One of these signals, trig_intrinsic, comes from the and_trig signal shown in Fig.2, but only when the write count is 0, 1 or 2, signified by signal wr012 being high, and also only when both runx and runz are high with triggering enabled, en_a =1. When start_write is high, signal latch_write_count is clocked high for one clock cycle, which causes the write_count to be initialized, normally to twice the data block length. In Fig.4 the write_count is set to 4, corresponding to a value of 2 for the data block length.

When signal latch_write_count falls, the write enable signal to the FIFO, wen_b, is pulled low, allowing data from the DDR register to be written to the FIFO. wen_b is pulled high, terminating writing to the FIFO, when the write_count goes to 0.
4 Other Information

Note in Fig.2 that the time word is added to A/D samples as four-bit nibbles, so data to the FIFO consists of two eight-bit A/D values plus a four-bit time nibble, making the FIFO input 20 bits wide.

Not shown in the logic is a continue bit that can extend writing to a FIFO by an additional data block if the analog input signal is still above the threshold value when the last data from the present block is being written.

In Fig.3 is shown a trig_out signal consisting of the AND of signals an_d_c_trig, runx and runz but not including en_a or wr012. This is a trigger signal to the other three channels of the WFD so that a trigger from this channel can also trigger other channels to store FIFO data, if so-enabled. The trig_out signal is stretched to 1.5 clock cycles to ensure that the other channels have time to recognize the trigger.
Fig. 1  The chain of run bits
Fig. 2 Logic for writing to a FIFO
Fig. 3 Write trigger logic
Fig. 4  Signal timing showing correct lineup of sample n with a time value of 0.