EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN-EP/86-23
10 February 1986

**THE LTD:
A FASTBUS TIME DIGITIZER FOR LEP DETECTORS**

G. Delavallade and J.P. Vanuxem
CERN, Geneva, Switzerland

ABSTRACT

This paper describes a high-performance time digitizer for application with detectors requiring a fine time-resolution across a wide dynamic range, as well as multiple event buffering at the front-end level. The flexibility of the device, combined with a relatively low cost, should also make it attractive for more conventional applications. It is planned to use modern technology during the development of this unit, e.g. high-speed memories, PALs, and LSI semi-custom ICs (gate arrays).

To appear in the
Proceedings of the Wire Chamber Conference,
Vienna, 25-28 February 1986

1. PRINCIPLE OF THE TIME MEASUREMENT

The LEP time digitizer (LTD) measures the time of arrival of the leading edge of signals (hits) from 48 wires.

The measurement is done with an external clock (normally: $F = 125$ MHz; option 1: $F \leq 125$ MHz), which is started at pretrigger time t_0 and generated for the duration of the Data-taking interval (i.e. the maximum drift-time if the detector is a drift chamber).

The clock increments a coarse-time (CT) counter every 8 ns; the time information contained in this counter is transferred into a 'coarse-time memory' (CTM) when a hit is detected on one (or more) wire(s). The CTM has the same width (12 bits) as the CT counter, and is 256 words deep.

In order to improve the time resolution by a factor of 4, a 2 ns 'time interpolator' circuit is used on each wire. This circuit is based on a high-speed shift register, which is first loaded with the fine-time (FT) information of the hit on the wire through its parallel inputs, and is then shifted in order to transfer this information serially into a 'fine-time memory' (FTM). The FTM is 48 bits wide and 256 words deep. The hit information for a given channel 'i' is placed in the i^{th} bit position and is formatted across five (option 1: 2) consecutive words of FTM. The first word contains a header bit ($= 1$) to indicate the presence of a hit, the three following words (if option 1 is not selected) contain the interpolator value, and the last word contains a trailer bit ($= 0$). This serial storing of the FT is done with a view to optimizing the memory space usage, taking advantage of the fact that the detector and its associated analog chain have an intrinsic non-zero dead-time.

The 'input memory' (INM), consisting of the common CTM plus the FTM of all channels, is the place where the raw event data are stored during data-taking, and therefore constitutes the first level of buffering. Figure 1 shows the internal formatting of data in INM, and it stresses the fact that hits from different channels are also permitted to come simultaneously or to overlap.

The FT information itself is obtained by means of three delay elements placed on the input signal path: these delay elements are merely gates of the ECL 10K series, which have been factory-selected so as to guarantee a propagation delay $t_{pd} = 2 \pm 0.2$ ns on the negative-going transition of the input signal. This gate solution has the advantage of being much cheaper than the delay-line solution, but at the expense of more power consumption. The interpolator and delay elements are represented in fig. 2, which also shows the rest of the front-end circuitry for one channel if it were implemented with the usual ECL 10K and 10KH series of SSI and MSI ICs. The interpolator normally operates in Parallel-Entry mode, i.e. the CONTROL FF is cleared as long as there is no signal on the wire. Whenever a signal is sensed on the wire during a low to high transition of the clock, Q_3 becomes high. This sets the CONTROL FF, placing the interpolator in the Shift mode. At the same time, the monostable generates a signal to trigger the internal pulser. This pulser is responsible for the generation of the appropriate number of Write cycles to the FTM, which is five if the FT information is stored in the FTM (normal case where the interpolator is used), and two if the interpolator is not used (option 1; this option is selected by setting CSR # 1[0] to 1). In order to ensure proper timing, an extra FF is inserted to serve as an 8 ns delay element on the path of the data transmitted serially to the input of the FTM.

Figure 2 also shows, on the input side, a differential receiver, an OR gate to allow for a common TEST input signal (this OR gate is in fact a multiplexer in the actual implementation), and an AND gate which is used to inhibit the input signal when an individual MASK bit has been set by software. A HIT FF (one per channel) is set whenever a signal has been detected on the

wire during the Data-taking phase: this FF is part of the hit-pattern register (CSR #13,14). A SELECT FF (one per channel) controls the selection of the FTM memory corresponding to the same channel. At Data-taking time, all SELECT FFs in the LTD are cleared, so that all FTM memories are ready to accept data from the interpolators in parallel. At Conversion time, the SELECT FFs will all be set, except the one corresponding to the FTM memory being interrogated.

In order to achieve the highest degree of integration (a gain of 30 is expected), to reduce the power consumption (by a factor of 3) and the cost (by a factor of 2), and to considerably improve the reliability, it is planned to implement four channels of the front-end circuitry within a gate array; a preliminary circuit diagram for this semi-custom LSI can be seen in fig. 3.

After a Data-taking phase has been completed, the LTD is usually asked to perform a Conversion, during which the raw data are taken from INM, reformed, and finally placed into one of the four output event buffers. The circuit block diagram of the complete LTD unit (fig. 4) shows the gate arrays, the INM, the logic associated with the Conversion, the four output buffers, as well as the FASTBUS NTA and CSR registers. The whole card utilizes ECL 10K and 10KH technologies throughout, and the Slave coupler designed in the CERN EP Electronics Research and Development Group [1] will be implemented during the prototype phase.

2. THE THREE MAIN TASKS OF THE LTD

When the LTD is inactive, it is in the IDLE state (see fig. 5).

For a given (accepted) event, the LTD executes the following three tasks in the same order as they are described below.

2.1 Data-taking

This task concerns the capturing of an event from the channel inputs into the INM.

It is initiated at time t_0 by an external clock generator (ECG) which, after the front-end has been initialized by means of an INIT signal (FASTBUS command, or auxiliary connector signal), generates a series of clock pulses which are distributed to every LTD in the system via its front-panel CLOCK input.

If an event currently being captured becomes invalidated at time t_1 of the level-1 trigger decision (e.g. $t_1 = t_0 + 2.2 \mu s$)*), or after the maximum drift-time has been reached, whichever comes earlier, the ECG stops generating clock pulses: this normally terminates the Data-taking. If the CT counter overflows, a time-out is generated, and the Data-taking is aborted; further clock cycles are ignored until an INIT signal is received. This time-out is hardware-selectable to six different values, which are 32, 16, 8, 4, 2, or $1 \mu s$, in the case where the frequency of the clock is 125 MHz.

After level-2 trigger decision time t_2 (e.g. $t_2 = t_0 + 35 \mu s$ *) and depending on whether the current event is accepted or rejected, either the Conversion of that event or the Data-taking of the next event may be started (overwriting in this latter case the previous contents of INM, which are lost).

*) The figures quoted for t_1 and t_2 apply to the DELPHI DAQ system.

2.2 Conversion

The Conversion of an event may follow its Data-taking: this normally happens if the event has been validated successively by level-1 and level-2 triggers.

The number of the buffer assigned for data conversion in the LTD multi-event output buffer—which from now on is called the ‘front-end buffer’ (FEB) for sake of conformity with a more or less accepted terminology [2, 3]—is indicated by the two LSBs of CSR # 1. This register is normally loaded, before a Conversion is started by a Start Conversion FASTBUS command or a CONVERT signal (auxiliary connector). The LTD then starts converting the event data from INM and transfers the results of the conversion into the assigned FEB buffer (one of four).

The Conversion of an event is normally terminated after all the valid data previously contained in INM have been transferred into the assigned FEB buffer. It may also be prematurely terminated by an ABORT signal (auxiliary connector), or by a Stop Conversion FASTBUS command. Note that a maximum conversion time of 250 μ s occurs when the INM has been filled up (very improbable case!) by hit information during the previous Data-taking.

2.3 Readout

This third and last task concerns the fast readout of the converted event data via FASTBUS.

The LTD is first selected in data space by a primary address cycle. A secondary address cycle then loads the number of the FEB buffer to be read out into a 2-bit NTA data-space register; at the same time this automatically sets the internal FEB address pointer to the starting address of the appropriate FEB buffer. The event data (up to 512 words) are then read out in block transfer mode. An SS = 2 response signals to the reading Master when no more valid data are available for that event. An SS = 6 response is, however, generated (instead of SS = 2) if a memory overflow occurred in INM during the Data-taking of this event, or if a memory overflow occurred in the FEB buffer during its Conversion: note that in either case, bit 22 of the Marker Word is set, indicating a ‘hit overflow’.

The reading Master is responsible for the termination of the Readout, and may re-initiate, if required, a complete Readout of the same event in the way just described. After the data for a complete event have been successfully read out, the Master usually releases the corresponding FEB buffer by asserting the appropriate Release bit R_i in CSR #0, so that this FEB buffer becomes free for the conversion of the next event.

2.4 Concurrent tasks

The three successive tasks related to the same event are executed in the order in which they have previously been described; they are not allowed to run in parallel, which means that a task may only be started after the previous one has been completed. Between two successive tasks, the LTD is in the IDLE state.

The two ‘extreme’ tasks (Data-taking and Readout) are, however, permitted to run concurrently if they deal with two different events, i.e. the Readout of one event may occur in parallel with the Data-taking of a later event. This is shown by the dark arrows in fig. 6; in this example, the Readout of an event from FEB buffer 3 is seen to take place concurrently with the Data-taking of a new event, which is subsequently converted into FEB buffer 1. The multi-event capability of the LTD therefore amounts to a total of five events (four in FEB, plus one in INM).

No concurrence is allowed between two ‘neighbouring’ tasks: trying to run the Conversion of one event concurrently with the Data-taking of another event always results in the Conversion having priority over the Data-taking; the external CLOCK signal is ignored during the

Conversion, and the SR flag is set. Trying to run the Conversion of an event concurrently with the Readout of a previous one is not permitted either; the running task has priority over the newer one, which is ignored. This situation is treated by the generation of an $SS = 1$ response during the appropriate FASTBUS cycle.

3. SHORT-FORM SPECIFICATIONS OF THE LTD

TIME RESOLUTION: 2 ns (option 1: ≥ 8 ns)

DYNAMIC RANGE: 32 μs (option 1: ≥ 32 μs)

NUMBER OF CHANNELS: 48

CONVERSION TECHNIQUE: Digital; uses one 2 ns interpolator per channel
(option 1, software selected: the interpolators are not used)

MULTI-HIT CAPABILITY: 51 (option 1: 128) hits minimum

MULTI-EVENT BUFFERING: 5 events maximum

EVENT SIZE: 512 words maximum, consisting of

Header: 3 words for the hit pattern
Body: 508 words maximum, 1 word per hit
Trailer: 1 Marker word

INPUT SIGNALS:

Level: ECL differential, sensitivity = ± 150 mV, CMR ≥ 1.5 V
Width: 40 ns (option 1: 2/F) minimum
Dead-time between pulses: 8 ns (option 1: 1/F) minimum
Individual masking possibility for all channels

CLOCK: ECL differential input; $F = 125$ MHz (option 1: $F \leq 125$ MHz)

CONVERSION TIME: 250 μs maximum

READOUT SPEED (intrinsic): ≤ 50 ns per word in block transfer

ZERO SKIPPING: It is implemented at three levels:

Hit: Only valid data is stored in INM during Data-taking
Channel: Only hit channels are interrogated during a Conversion
Module: Sparse Data Scan is implemented for the Readout

HIT PATTERN AVAILABILITY:

Current event: Auxiliary connector, or FASTBUS

Event being read out: Three-word header

FASTBUS PROTOCOL:

Fully complies with the FASTBUS specifications [4]

Addressing : Geographic + five different broadcasts

Data cycles (R/W): Secondary Address, Random, Block Transfer

NTA registers: 2 (1 for data, and 1 for CSR)

CSR registers: 7 in total: CSR # 0, 1, 10, 11, 12, 13, 14

AUXILIARY CONNECTOR: It is used to convey:

Common Control and Status signals

Hit pattern information of the last event

MECHANICS: FASTBUS module, 1 U wide

4. RESULTS

In order to check the validity of the time digitization method, six modules have been built in CAMAC, each one implementing eight channels. Tests on these modules have been made independently by different groups and institutes interested in the project, and have clearly demonstrated both the validity of the time digitization method and its performance.

A complete specification of the LTD in its FASTBUS implementation has been worked out in close collaboration with the DELPHI Channel Standardization Group, and is now available on request [5].

Work on prototypes has started, including the design of the gate array chip, and a fully fledged FASTBUS prototype is planned for summer 1986. After a small series production by the end of 1986, a batch of 30,000 channels, representing a total of about 600 LTD modules, will be made available.

REFERENCES

- [1] G. Fremont and E. Sanchis, A general-purpose, PAL-based Master and Slave FASTBUS coupler, CERN EP Electronics Note, March 1985.
- [2] Ph. Gavillet, Trends in new collider experiment data-acquisition systems, Nucl. Instrum. Methods A235 (1985) 363.
- [3] Ph. Charpentier et al., Proposal for the data-acquisition system of the TPC and perspectives for the TPC tests, CERN DELPHI Note 85-73 DAS-20 (1985).
- [4] FASTBUS: Modular high-speed data-acquisition and control system for high-energy physics and other applications (US NIM Committee, December 1983).
- [5] G. Delavallade and J.P. Vanuxem, The LTD: A FASTBUS time digitizer for LEP detectors (Preliminary specifications), CERN EP Electronics Note 85-06 (1985), and CERN DELPHI Note 85-109 ELEC-15 (1985).

FIGURE CAPTIONS

Fig. 1: Data structure of input memory.

Fig. 2: LTD front-end circuitry.

Fig. 3: Gate array circuit diagram.

Fig. 4: LTD block diagram.

Fig. 5: The three main tasks of the LTD.

Fig. 6: Data-flow through the LTD.

In this example:

- Hits 1, 2 and 3 are disjoint
- Hits 4 and 5 are simultaneous
- Hits 3, 4, 5 and 6 are overlapping
- Hits 6 and 7 are disjoint

Fig. 1

AND ARE FACTORY SELECTIONS OF 10103

Fig. 3

Fig. 4

(Light bidirectional lines indicate the start/completion of a concurrent task)

<u>Data Taking</u>	[	1	Start receiving Clock pulses
		2	Stop receiving Clock pulses, or Time-Out
<u>Conversion</u>	[	3	Start Conversion command, or CONVERT signal
		4	Natural completion of the conversion, or Abort Conversion command, or ABORT signal
<u>Read-Out</u>	[	5	Load NTA-Data
		6	End of Block Transfer (SS=2, or SS=6), or Release command

Fig. 5

Note: The dark arrows indicate that the corresponding tasks may be run concurrently if they deal with different events.

Fig. 6