A Low-Dead Time Wave Union FPGA TDC for Optical Beam-Loss Monitor System

Yuchen Yang, Yijun Cai, Wei Peng

Center of free electron laser & high magnetic field, Anhui University, Hefei Anhui, China

Full Article

Abstract — At the Anhui University infrared free-electron laser (FEL) facility, an optical beam-loss monitor (oBLM) diagnostic system is under development for spatially continuous beam-loss monitoring along the accelerator beamline. To meet the stringent position-resolution requirements of compact accelerators and to mitigate event ambiguity under multi-bunch operation, an optical fibre is installed parallel to the beamline and coupled to photomultiplier tubes (PMTs) at both ends. Cherenkov light generated by beam losses propagates along the fibre to the two PMTs. After conditioning with a transimpedance amplifier and a high-speed comparator, the signals are digitized by an FPGA-based time-to-digital converter (FPGA-TDC) to obtain precise timestamps. The differential arrival times measured at both ends enable accurate reconstruction of the bunch ID and the longitudinal beam-loss position. To satisfy the oBLM requirements on timing precision and short inter-event intervals, we propose a low-dead-time Wave Union FPGA-TDC architecture.

Limitation of the Traditional Wave Union TDC Encoder

Conventional WU-TDCs typically employ an **edge-position summation encoder**, in which the positions of all detected transitions in the TDL are identified and summed to generate the final fine-time code. Based on this mechanism, we analyze why the dead time of a WU-TDC is usually two clock cycles.

1.The pulse train associated with **Event1** may be sampled by two consecutive clock edges. If the pulse train is fully sampled in the second clock cycle, the conventional encoder does not produce an invalid code; instead, it encodes the same event again, resulting in a duplicated timestamp. Alternatively, if the pulse train has partially exited the TDL at the next sampling instant, the missing transitions reduce the fine-time code, producing a spurious and smaller timestamp.

2.If **Event1** and **Event2** appear in the TDL simultaneously, the conventional summation encoder merges the transitions from both events, regardless of whether the pulse train of Event1 is complete. This leads to an aliased and incorrect timestamp.

Therefore, after encoding a valid event, a conventional WU-TDC typically blocks event input in the next clock cycle and forces the next-cycle output to be invalid.

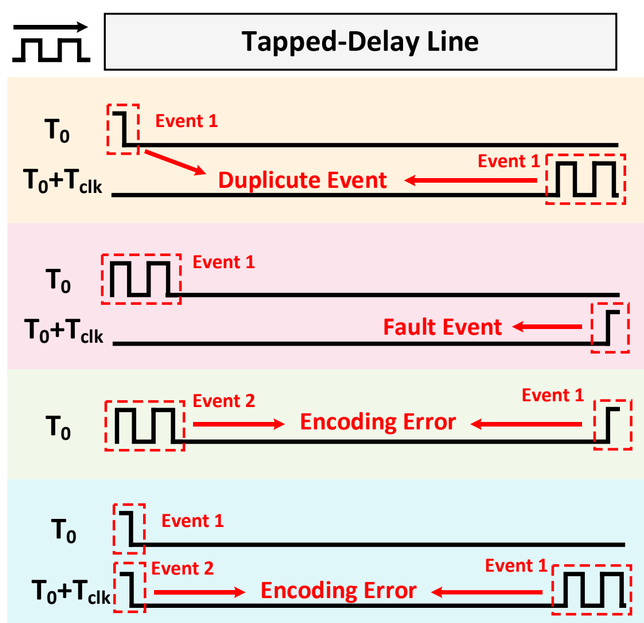

Fig. 1. Error scenarios with traditional encoder.

Purposed TDC Structure

When the minimum event interval is **one system clock period**, at most two events can coexist in the TDL at any time, with a sufficiently large separation. The same applies to each sub-TDL, where the events are referred to as **sub-events**. Based on this property, we propose a region-based aggregation encoding method to distinguish transition edges from different events.

Specifically, each sub-TDL is divided into three overlapping regions: the **front**, **middle**, and **rear** regions. Two properties can be derived:

- When two sub-events coexist, they appear in the front and rear regions, respectively;
- When an event reaches the middle region, only one sub-event is present, and its code can be jointly determined from the front and rear regions.

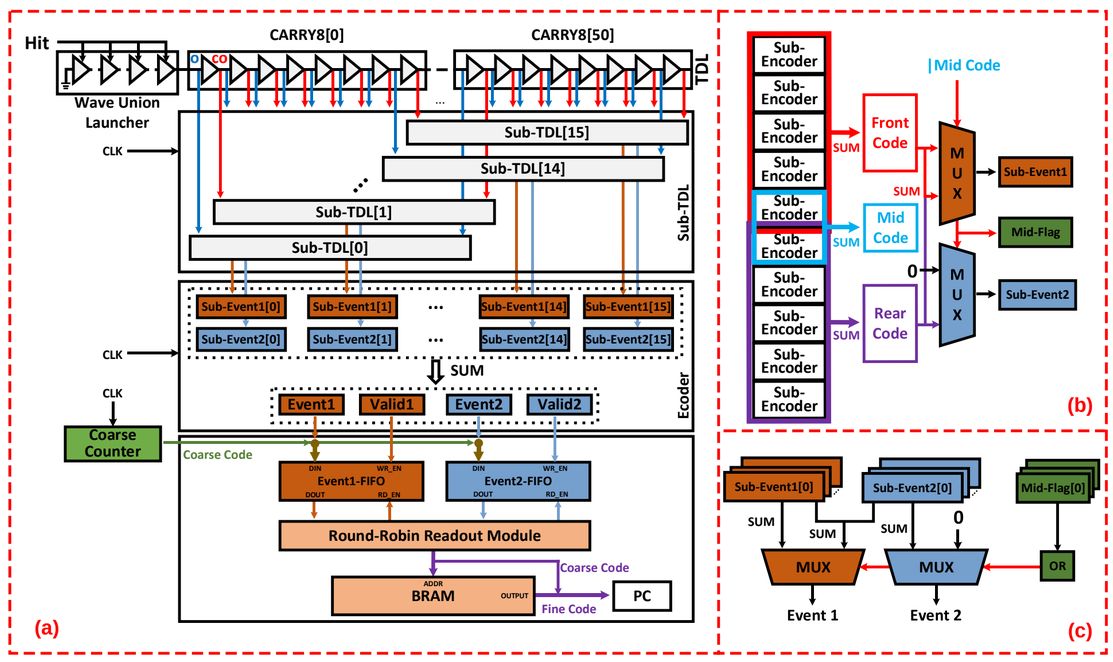

Fig. 2. (a) Block diagram of the proposed TDC. (b) Sub-event aggregation encoder. (c) Sub-TDLs encoder.

Test Result

The proposed 4-edge WU TDC is implemented on the AMD Zynq UltraScale+ ZU3EG FPGA and operates at a 750 MHz system clock. Random hits for CDT were generated by a waveform generator (33600A, Keysight). In the time-interval test (TIT), two hit signals with a fixed TI are generated by a waveform generator and fed into the two TDC channels. For each TI setting, fifty thousand samples are collected while the TI is swept from 0 ns to 5 ns in 139 ps increments. In dead time test (DTT), a CARRY8-based test launcher was implemented inside the FPGA with an architecture identical to the WU launcher. Upon arrival of the hit signal, it generates two pulses separated by 1.33 ns (one clock period) and injects them into the TDC. Key signals from the TDC system are then packaged and transmitted to a PC, where subsequent analysis is performed in MATLAB to verify proper system operation.

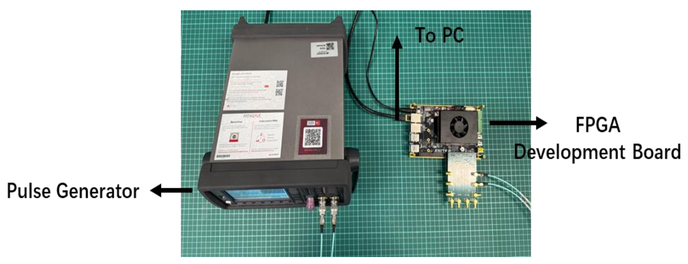

Fig. 3. Experimental setup.

	Coarse	Event1 Valid	Event1 Fine	Event2 Valid	Event2 Fine	
Case 1	47	0	0	1	2593	Pulser1
	48	0	0	1	2596	Pulser2
	...					
Case 2	2	1	88	0	0	
	3	1	88	0	2887	
	4	0	0	0	2889	
	...					
Case 3	40	1	494	0	0	
	41	1	498	0	766	Duplicute
	42	0	0	0	766	
	...					
Case 4	25	1	50	1	2811	
	26	0	0	0	2811	

Fig. 4. Encoding results.

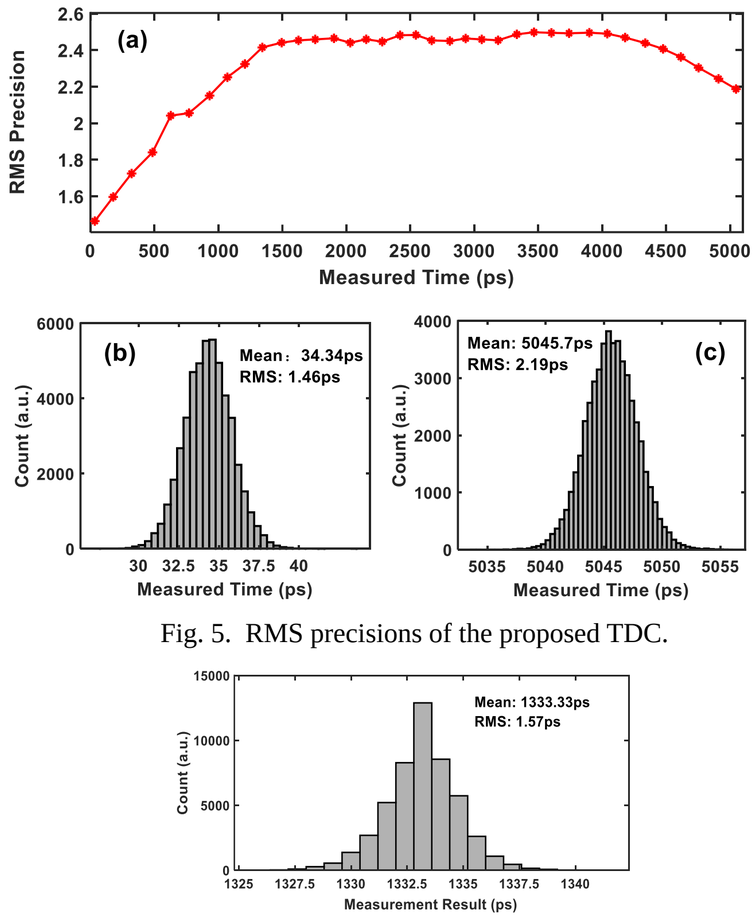

Fig. 5. RMS precisions of the proposed TDC.

Fig. 6. Histogram of the measured 1.33ns time interval.