INTEGRATING FAST BEAM INTERLOCKS INTO AN EVENT-BASED TIMING SYSTEM

L. Perusko†, A. Debenjak, Z. Oven, T. Zagar, Cosylab, Ljubljana, Slovenia
E. J. Harms, L. Szeto, Nusano, Salt Lake City, USA
J. Ringuette*, TerraPower, Bellevue, USA

Abstract

Accelerators often use separate infrastructures for precise timing and machine protection, which increases cabling, integration complexity and operational overhead. We describe an approach that enhances a standard event-based timing system to provide fast beam interlock functionality on the same platform. Protection logic, flag distribution and rapid reaction paths are implemented directly in the timing firmware, allowing the existing hardware and fiber network used for device triggering to also transport low-latency interlock information. With timing and protection combined, interlock conditions can act immediately on timing outputs while established capabilities such as sequences, delay compensation and timestamping remain fully supported. The system offers flexible configuration of inputs, flags and outputs, and a single operator interface for both timing and protection workflows. Developed by Cosylab in collaboration with Nusano, this solution demonstrates practical architecture for environments where precise timing and fast protection must operate tightly coupled.

EVENT-BASED TIMING SYSTEM

Distributed accelerator control systems require precise orchestration of devices spread across a facility. Timing systems for particle accelerators face particularly demanding requirements: sub-nanosecond synchronization, low jitter and deterministic latency across all endpoints.

An event-based timing system addresses these requirements by distributing a common synchronous clock from a central Event Master (EVM) to a set of Event Receivers (EVRs) over dedicated fiber-optic links. On each clock cycle the EVM may encode an event code into the serial stream; each EVR decodes the stream and reacts to selected event codes according to its individual configuration, generating precisely timed trigger pulses. Despite the differing single-optic cable lengths between the EVM and individual EVRs, triggers issued by different EVRs in response to the same event arrive simultaneously at their respective endpoints as a combined consequence of phase-locking all receivers to the common clock and of the delay compensation subsystem, which corrects for the residual cable-length differences.

MRF timing system

One of the established commercial providers of event-based timing systems is Micro-Research Finland (MRF) [1]. MRF equipment has been deployed at numerous institutes and accelerator facilities over the past two decades. The hardware is available in multiple form factors; the implementation described here uses the mTCA.4 [2] variant, comprising mTCA-EVM-300 event generators and mTCA-EVR-300U event receivers housed in standard mTCA chassis together with system-board computers (SBCs) running the control software.

A key feature of the MRF platform is bidirectional communication: while events flow downstream from the EVM to all EVRs, the data slot of the serial link also supports upstream data transfer from each EVR back towards the EVM. This upstream path, normally used for delay compensation measurements and data buffers, is the foundation on which the interlock extension is built.

For the Nusano linac, the timing system consists of one EVM master and one EVM configured as a fan-out, together serving twelve EVRs distributed across the facility in a two-hop star topology as shown in Fig. 1. The event clock runs at 100 MHz, providing 10 ns resolution on all trigger delays. Timing sequences repeat at up to 400 Hz, with the specific set of triggers within each sequence configured by operations to suit the current machine mode. A GPS synchronized time server provides a PPS signal that synchronizes timestamps across the entire timing system.

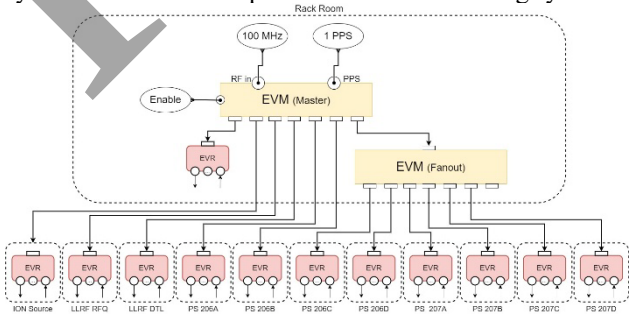

Figure 1: Topology of the Nusano Timing System.

INTERLOCK EXTENSION

Requirements

The Nusano linac requires both a Beam Permit (BP) system, which monitors device readiness and inhibits operation while any subsystem is not ready, and a Fast Beam Interrupt (FBI) system, which responds to sudden faults and latches until explicitly cleared by an operator. Running separate protection hardware would duplicate the fiber infrastructure and create two independent operator interfaces. The chosen approach instead implements BP and FBI

†luka.perusko@cosylab.com
*Current affiliation, was part of Nusano team during the project

functionality directly in the FPGA gateware of the existing MRF timing boards.

The key requirements driving the design are:

- End-to-end latency: the propagation time from any physical interlock input to any affected timing output must not exceed 30 µs across the full two-hop topology.
- Fail-safe communication: broken fiber links must be detected automatically and treated as a fault condition without software intervention.
- Flag abstraction: up to 16 independent logical flags aggregate inputs by logical AND, with each flag independently configurable as non-latching (BP) or latching (FBI).
- Flag to IO mapping: inputs-to-flags and flags-to-outputs mappings are fully user-configurable, allowing the same gateware to be adapted to different machine protection topologies.
- Input conditioning: per-input configurable debounce time to reject glitches without introducing unnecessary latency.
- Timestamped logging: every change of a BP/FBI input or output gate is logged with a hardware timestamp synchronized to the MRF event clock, enabling post-mortem analysis.
- Postmortem distribution: a flag transition can trigger a dedicated timing event, synchronously delivered to all EVRs, which drives postmortem data acquisition.

New Functionality

Fail-Safe Communication Protocol A short flag message is added to the MRF serial data slot alongside the existing delay compensation and data buffer traffic. Each message is framed by dedicated start and end markers reserved for this extension and carries the current flag state together with a checksum that protects the payload against transmission errors. The transmitter sends these messages at regular intervals, and the receiver expects a fresh, valid message within a fixed time window; if the message is late, missing, or fails the checksum, the receiver raises a communication error flag (comFlag). As a result, each fiber link continuously verifies its own health and detects faults on its own, without relying on other links or on software. While fail-safe communication is active, ordinary data buffer messages are limited in length so that they cannot delay a flag message beyond its expected arrival window.

Flag Processing Flag handling runs as an independent pipeline that lives alongside the standard MRF event distribution but does not share its delay-compensation logic: timing events are still aligned across the facility as before, while flags follow their own dedicated, bounded-latency path optimized for fast fault response.

On each EVR the physical inputs (typically dry contacts or logic signals from subsystem readiness or fault outputs) are debounced to reject brief electrical glitches. The conditioned inputs are routed through a user-configurable mapping that groups them into up to sixteen logical flags; a flag stays in the OK state only while every input assigned to it reports OK, and an unassigned flag is treated as permanently OK. These sixteen flags, together with the link's own comFlag, are packed into an upstream flag message and sent to the EVM.

The EVM aggregates the flag messages from all connected EVRs into a unified peride view: a global flag remains OK only while every EVR reports it as OK, so a single fault anywhere in the system is enough to drive the corresponding global flag into the fault state.

At the EVM master each global flag then passes through a latch whose behavior is selected per flag: Beam Permit flags follow their inputs continuously and recover automatically when the fault clears, while Fast Beam Interrupt flags hold the fault state once asserted and require an explicit software acknowledgement before they can recover. The resulting flag vector is broadcast back downstream over all fiber ports at the same time, and fan-out EVMs simply pass this message through unchanged.

Each EVR receives the downstream flag message and uses a second user-configurable mapping to decide how the system-wide flags affect its outputs. A timing output configured as an interlock-gated output is held in its idle state whenever any flag assigned to it is in the fault state, blocking the trigger pulse without altering the timing pattern itself. An output can alternatively be configured to mirror a flag directly, so its current state can be exported to external equipment as a plain hardware signal. When the BP/FBI extension is disabled the gating logic is bypassed and outputs behave exactly as in a stock MRF system.

Postmortem The falling edge of any flag selected in the postmortem enable register generates a trigger pulse inside the EVM master logic. This pulse is injected into the standard MRF user-event mechanism, so up to eight timing event codes can be pre-armed to fire together in response to a fault. Because these events travel through the standard MRF event distribution path they arrive synchronously at every EVR in the system, timestamped with event cycle precision, providing a common reference for all postmortem data acquisitions.

Validated Latency In the two-hop topology the worst-case time between a fault appearing at an EVR input and the corresponding output being suppressed on any other EVR has a clear, predictable upper bound: it is set by how long the signal spends travelling over the fiber links, by how often flag messages are sent, and by the fixed processing time inside each device. With flag messages sent at their highest rate, end-to-end measurements from input assertion to output response confirmed that this delay stays well within the required 30 µs.

SOFTWARE INTEGRATION

The EPICS control framework [3] is used throughout the Nusano control system. The timing application extends the open-source mrfioc2 EPICS support module [4], which already provides device support for standard MRF hardware. Extended C++ device-support objects expose each gateware register as an EPICS Process Variable (PV) accessible over Channel Access or PVAccess.

Separate EPICS database templates are provided for the EVM master, EVM fan-out and EVR roles. Stock mrfioc2

databases handle timing sequences, delay compensation and timestamping; new gateware databases cover fail-safe communication status and configuration, flag state readbacks, input-to-flag and flag-to-output mapping as show on Fig. 2, per-input debounce time, and logging. Template instantiation at IOC build time assembles the correct database for each crate from these modular pieces.

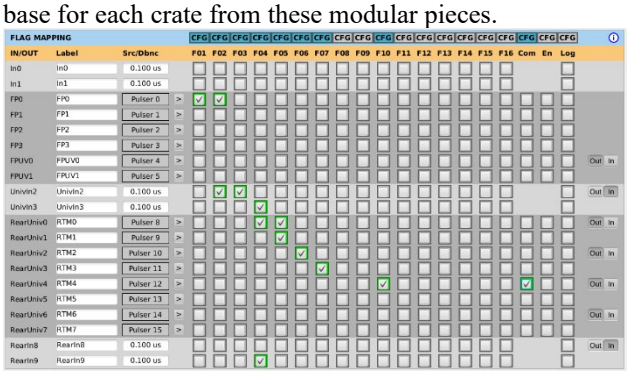

Figure 2: Matrix that provides mapping of input to flags and flags to output.

On each EVR every change of an interlock input or output gate is captured in hardware the moment it happens. Each log entry is stamped with a timestamp generated directly from the event clock, so the recorded times are aligned across the whole timing system and can be correlated between EVRs without post-processing, and is accompanied by a snapshot of the input and output states at that instant as shown in Fig. 3. The IOC reads entries out of the on-board buffer as soon as the gateware signals that new data is available. Should the buffer fill faster than software can drain it, the hardware deliberately preserves the earliest entries (the ones that capture the onset of the fault) by halting further writes and notifying the software, which drains what is held and clears the overflow condition, ensuring that the most diagnostically valuable information is never overwritten. The drained entries are exposed as waveform PVs implementing a circular buffer in the IOC.

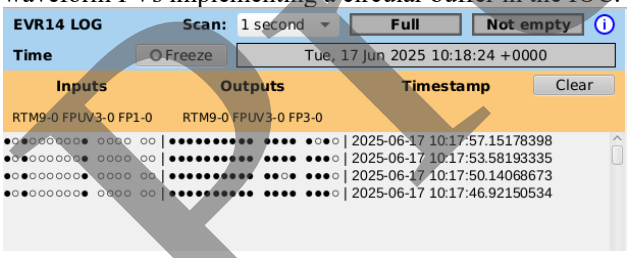

Figure 3: Log of changes on the EVR outputs.

All configurable parameters (mapping matrices, debounce times, flag modes, postmortem enables, communication periods) carry autosave annotations so that the IOC restores the previous configuration automatically on reboot.

Operator interfaces are implemented in CS-Studio Phoebus [5]. A hardware overview screen shows link status and device health for all EVMs and EVRs; a gateware summary screen displays the state of all sixteen flags together with per-EVR connection status; and per-device detail screens expose the full mapping configuration and communication error counters. Because timing and protection share one IOC per crate, operators interact with a single interface rather than switching between separate timing and interlock systems.

STATUS

The system described here is currently being deployed at the Nusano Production facility in West Valley City, Utah. Site acceptance testing has been completed with all distributed timing units installed. End users are currently utilizing both the timing and interlock functions to support RFQ commissioning.

ACKNOWLEDGEMENTS

On-site installation and commissioning of the system was conducted by Nusano staff H. Zhou, Z. Cherrington, B. Eftimov, D. Asad, J. Anderson. Their contributions are valued.

CONCLUSION

Cosylab and Nusano have jointly developed and deployed a novel architecture for the latter's linear accelerator that integrates fast beam interlock functionality into an existing MRF event-based timing system without additional hardware or dedicated interlock cabling. A fail-safe communication protocol layered onto the existing MRF serial links carries BP/FBI flag information with end-to-end latency validated below 30 µs for a two-hop topology. The flag abstraction decouples input conditioning from output gating, provides latching behavior for FBI use cases, and supports synchronized postmortem triggering across all nodes. An EPICS-based software stack extending the mrfioc2 module exposes the complete configuration and monitoring interface through a single operator environment shared with the conventional timing functions. The approach is directly applicable to other facilities using MRF timing hardware where tight coupling between device synchronization and machine protection is required.

REFERENCES

[1] Micro-Research Finland Oy, `http://www.mrf.fi`

[2] T. Walter, F. Ludwig, K. Rehlich, and H. Schlarb, "Novel Crate Standard MTCA.4 for Industry and Research", in *Proc. IPAC'13*, Shanghai, China, May 2013, paper THPWA003, pp. 3633-3635.

[3] L. R. Dalesio *et al.*, "EPICS 7 Provides Major Enhancements to the EPICS Toolkit," in *Proc. ICALEPCS'17*, Barcelona, Spain, 2017, pp. 22–26. doi:10.18429/JACoW-ICALEPCS2017-MOBPL01

[4] mrfioc2 EPICS Support Module, `https://github.com/epics-modules/mrfioc2`

[5] K. Shroff *et al.*, "New Java Frameworks for Building Next Generation EPICS Applications," in *Proc. ICALEPCS'19*, New York, NY, USA, Oct. 2019, pp. 142–145. doi:10.18429/JACoW-ICALEPCS2019-WESH1002