Assessing NI FPGA-based platform with MXIe interface for use in ITER hard real-time investment protection applications

D. Karkinsky[1], A. Marqueta[1], E. Barrera[2], **I. García-Siguero**[2], J. Varlec[3], M. Ruiz[2], M. Sekoranja[3], P. Rebollo[2], and T. Kogovsek[3]

[1] ITER Organization, St. Paul-lez-Durance, 13067 France.

[2] Instrumentation and Applied Acoustic Research Group (I2A2), Universidad Politécnica de Madrid, Madrid, Spain.

[3] Cosylab, Gerbičeva ulica 64, 1000 Ljubljana, Slovenia.

iter
china eu india japan korea russia usa

COSYLAB
Advancing humanity. Engineering remarkable.

ACKNOWLEDGEMENTS
This work was supported under grant PID2022-137680OB-C33 funded by MCIN/AEI/10.13039/501100011033

INSTRUMENTATION & APPLIED ACOUSTICS RESEARCH GROUP

MINISTERIO DE CIENCIA E INNOVACIÓN

AGENCIA ESTATAL DE INVESTIGACIÓN

POLITÉCNICA

UNIVERSIDAD POLITÉCNICA DE MADRID

OUTLINE

- Background
- Motivation
- Step 1 – IO Timekeeper
- Step 2 – Improve the kernel module & MARTe2 interface
- Step 3 – MXIe interface management
- Validation
- Results & Conclusions

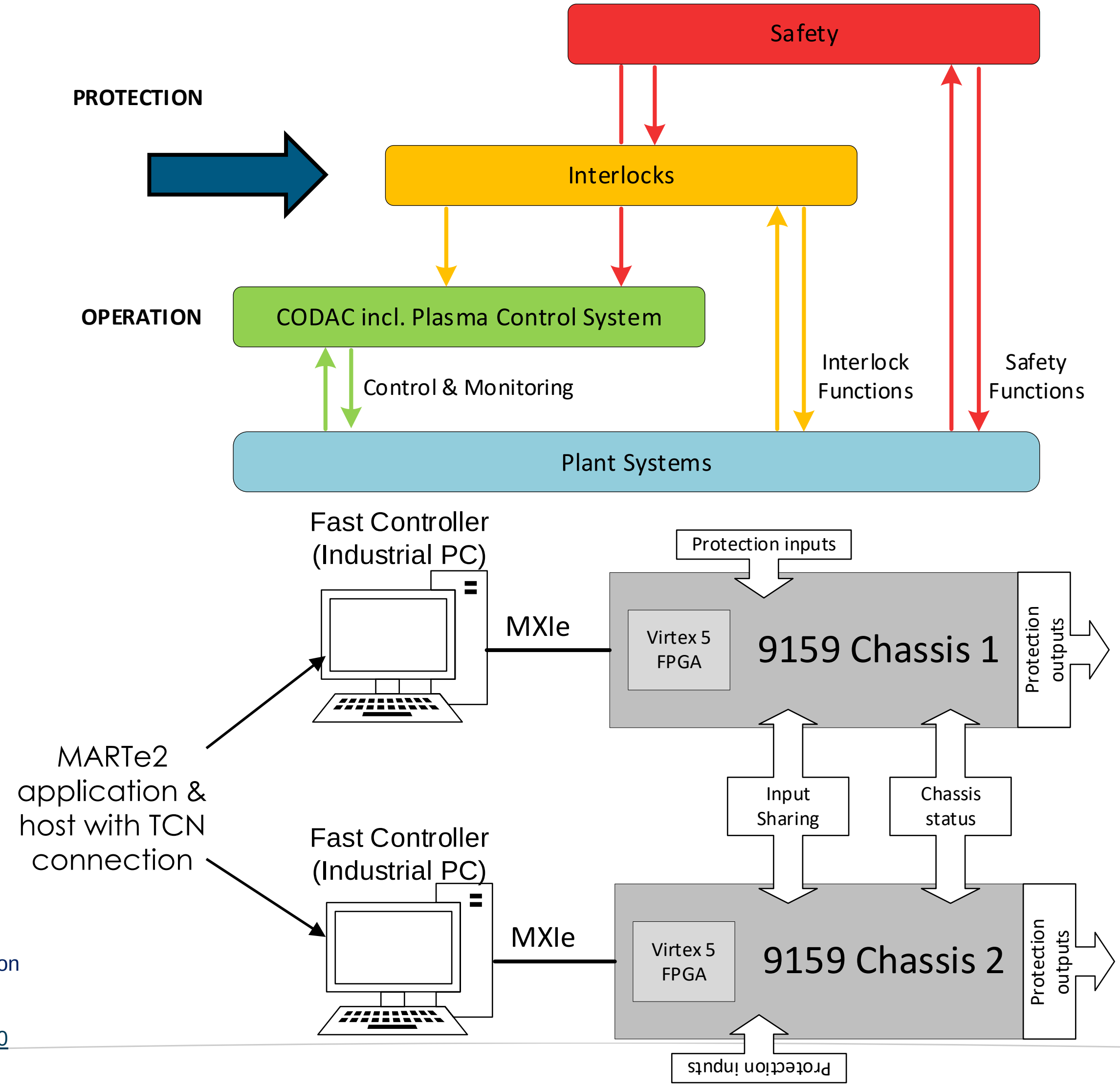
Background I

- The Interlock Control System (ICS) is the ITER system that implements the Investment Protection Functions, and is responsible for protecting the integrity of the machine against any possible failure

- It is divided into different architectures: hardwired, slow (PLC) and **fast (FPGA)**

- The cRIO NI9159 was qualified for use in the design of the ITER ICS Fast Architectures in 2016 using redundant two-chassis architecture in 1oo2D mode
  - As part of the qualification IO commissioned open-source Linux drivers,
  - Performed detailed analysis of the platform's internal architecture[1]
  - Following the IEC 61508 standard guidelines[2]

- Development through the LabVIEW FPGA tool



PROTECTION

OPERATION

Safety

Interlocks

CODAC incl. Plasma Control System

Control & Monitoring

Interlock Functions

Safety Functions

Plant Systems

Fast Controller (Industrial PC)

MXIe

Protection inputs

Virtex 5 FPGA

9159 Chassis 1

Protection outputs

Input Sharing

Chassis status

MARTe2 application & host with TCN connection

Fast Controller (Industrial PC)

MXIe

Virtex 5 FPGA

9159 Chassis 2

Protection outputs

Protection inputs

1. E. Barrera et al., "*Implementation of ITER Fast Plant Interlock System Using FPGAs With CompactRIO*," IEEE Transactions on Nuclear Science, vol. 65, no. 2, pp. 796-804, Feb. 2018, doi: 10.1109/TNS.2017.2783243.
2. I. García-Siguero et al., "*Verification and Validation of ITER Interlock System Fast Architecture According to IEC 61508 Standard*" IEEE Transactions on Nuclear Science, vol. 70, no. 6, pp. 1164-1170, June 2023, doi: 10.1109/TNS.2022.3224780

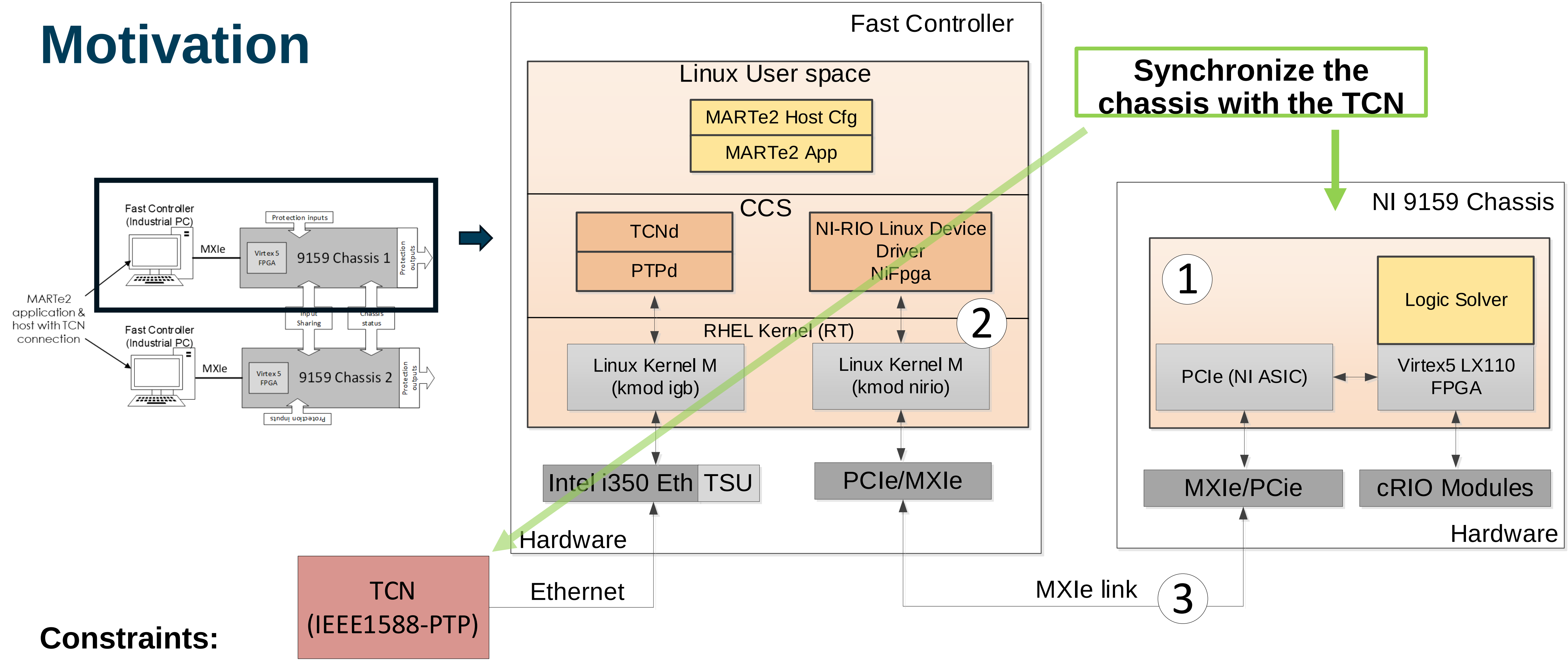
Background II

- The ICS was designed for event-action coordination (pure-logic) in 2016[1]
- In 2021 the ICS adopted the functions of the Advanced Protection Systems (APS) and the evaluation of the priority of machine stop requests.
  - Functions contain elements of dynamic control.
- APS:
  - detects events with the plasma e.g. disruptions,
  - controls shattered pellet injectors to mitigate the disruptions.
- ICS Stops request provide a grace-period to conventional layer to reach a desired final state.
  - Grace-period requires re-evaluation as events stack-up.
  - Expiration causes a request to CIS to escalate to interlock actions.
- All of this led to a change in the paradigm of designed architecture, which caused a number of challenges

Need to synchronize the CompactRIO with the ITER TCN (IEEE1588)
Requirements: **jitter of +/- 1 us and worst-case of +/- 5 us**

[1] J.L. Fernández-Hernando, D. Carrillo, G. Ciusa, Y. Liu, I. Prieto-Díaz, R. Pedica, S. Sayas, J. Soni, A. Vergara, The ITER interlock system, Fusion Engineering and Design, Volume 129, 2018, Pages 104-108, ISSN 0920-3796,

Motivation



Synchronize the chassis with the TCN

Constraints:

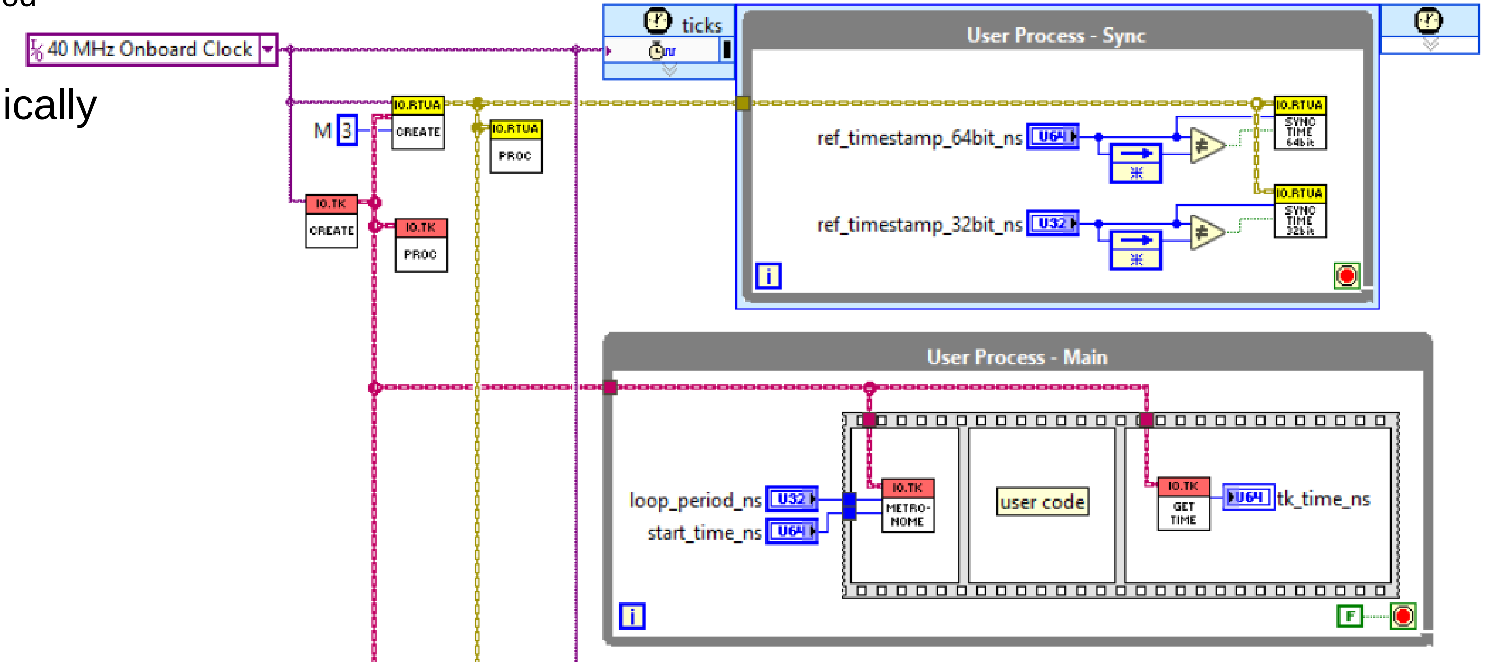
- **1**: NI 9159 has no PTP support but the ITER fast controller does.
- **2**: The NI RIO Linux driver has been designed for data-throughput and not for real time.
- **3**: The interface between the chassis and host is a MXIe link

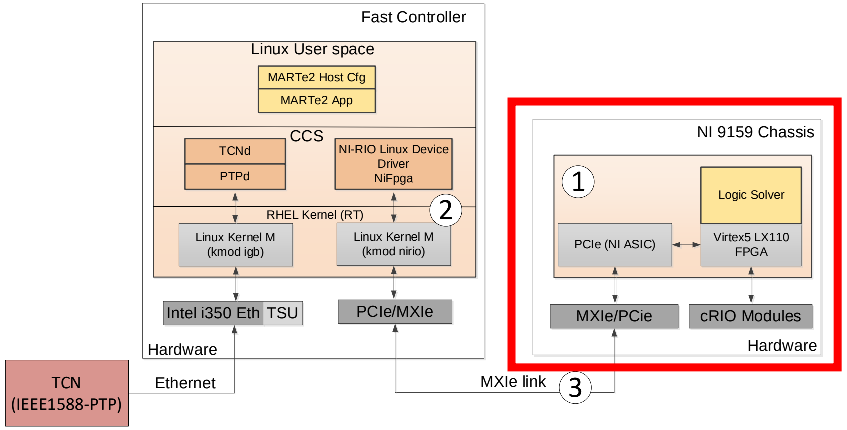
Step 1 – IO Timekeeper

Step 1: Development of IO FPGA Timekeeper Library.

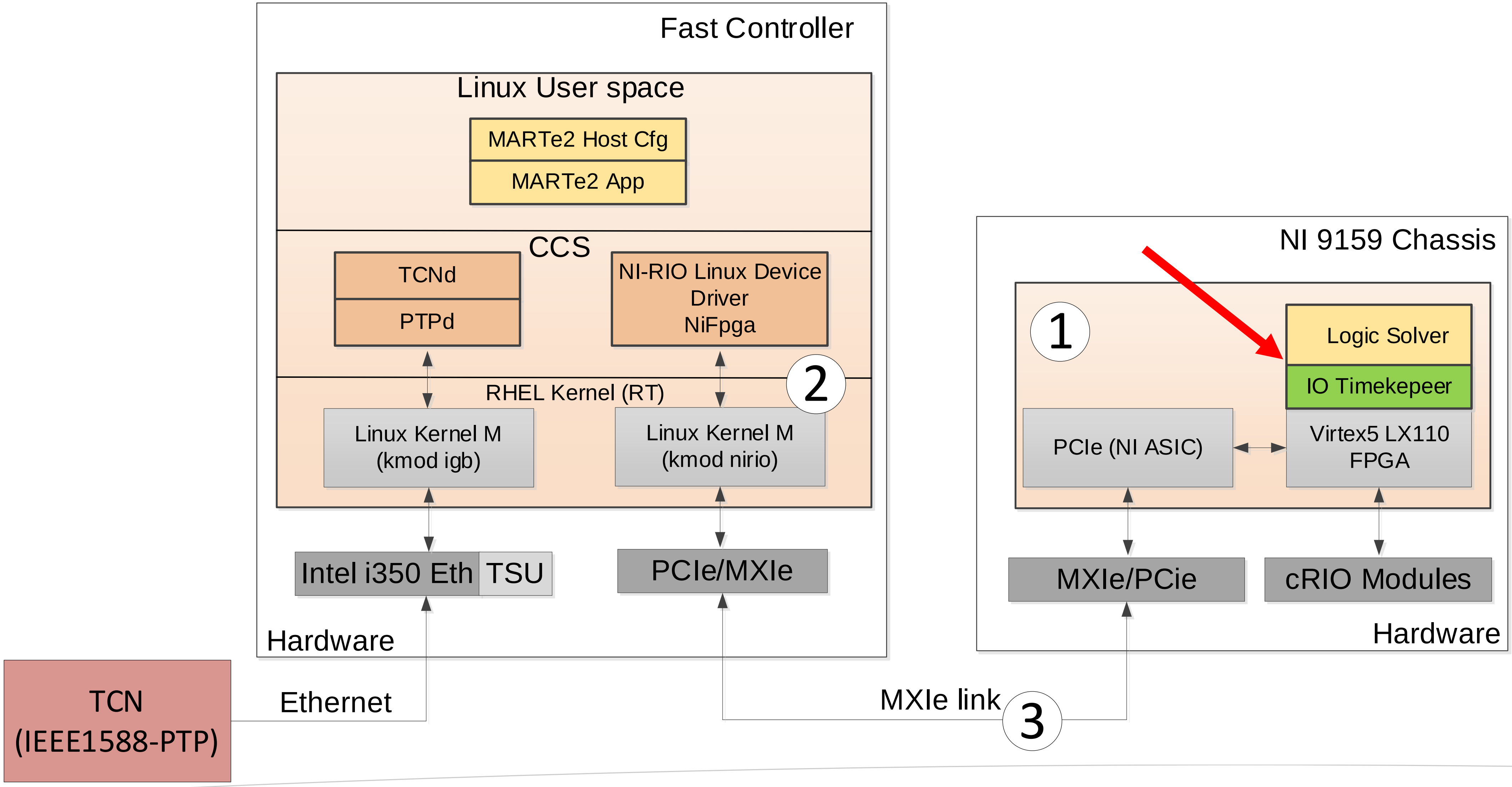
- Implements an internal ns counter that is synchronized with PTP time
- Provides functions to “wait_until”, “metronome_loop”…
- Provides a configurable Proportional Integral (PI) controller. Allows tuning to each particular system.
  - Configurable gains, filtering, error rejection and update period

- The time value reference must be periodically updated in a LabVIEW register.





IO Timekeeper



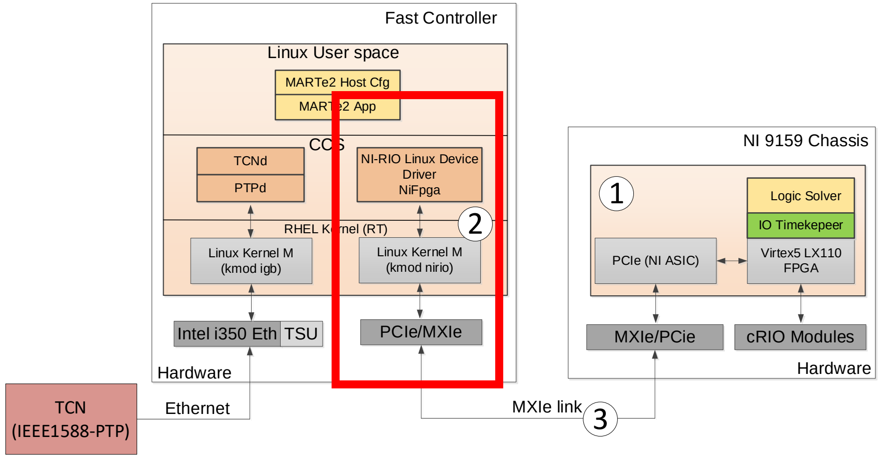
Step 2 – Improve the kernel module & MARTe2 interface

Step 2a: Developed a new NI9159 Linux device driver redesigning the current driver to improve real time performance (reduce latency jitter)

- MMIO operations (R/W FPGA Registers)
  - Removing unnecessary code (e.g. GPU buffering support)
  - Removing unnecessary support for Booleans, uint8 and uint16
- DMA Operations (Host to Target and Target to Host)
  - Moving from *streaming interface* to *packet-based interface*.
  - Providing a zero-timeout polling method to check when a packet is available to be read
  - Converting operations to Atomics so that they are not interrupted
  - Ported DMA buffers to cache-coherent memory limited to 16 kB (4 Linux pages)
- Security
  - Blocking critical operations on the FPGA for users who do not have privileges (e.g. to download a new bitfile to the FPGA)
  - Separate firmware resource query operations

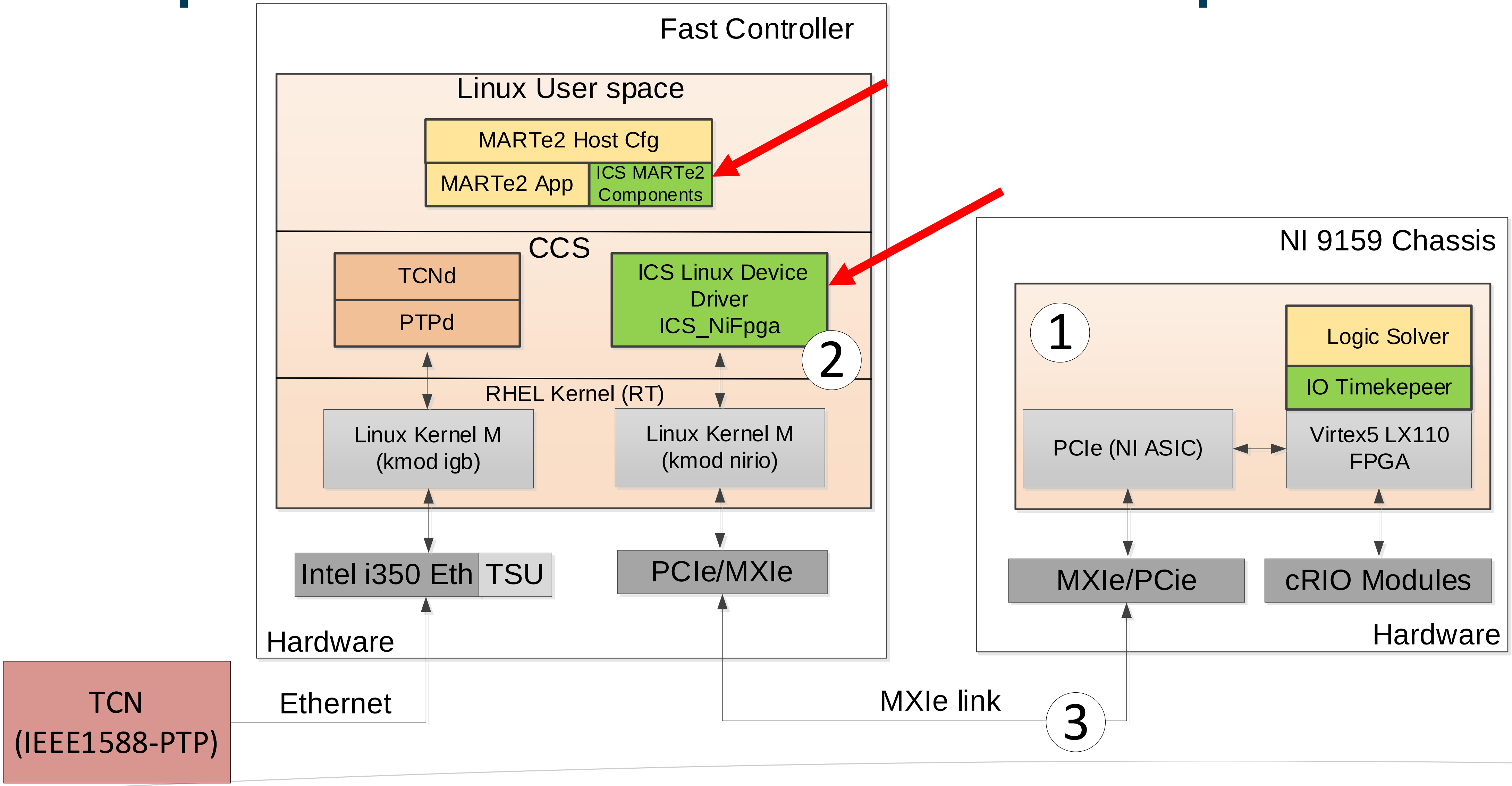
	Old (Streaming-based)	New (Packet-based)
Read 32-bit FIFO	Std dev = 34 us worst-case = 800 us*	Std dev = 0.2 us worst-case = 4.6 us
Write 32-bit FIFO	- *	Std dev = 1.56 us worst-case = 15.4 us

*National Instruments information



Step 2b: Providing the MARTe2 components for the use of new functions (Interfaces and data sources)

Timekeeper + Kernel Module + MARTe2 Components



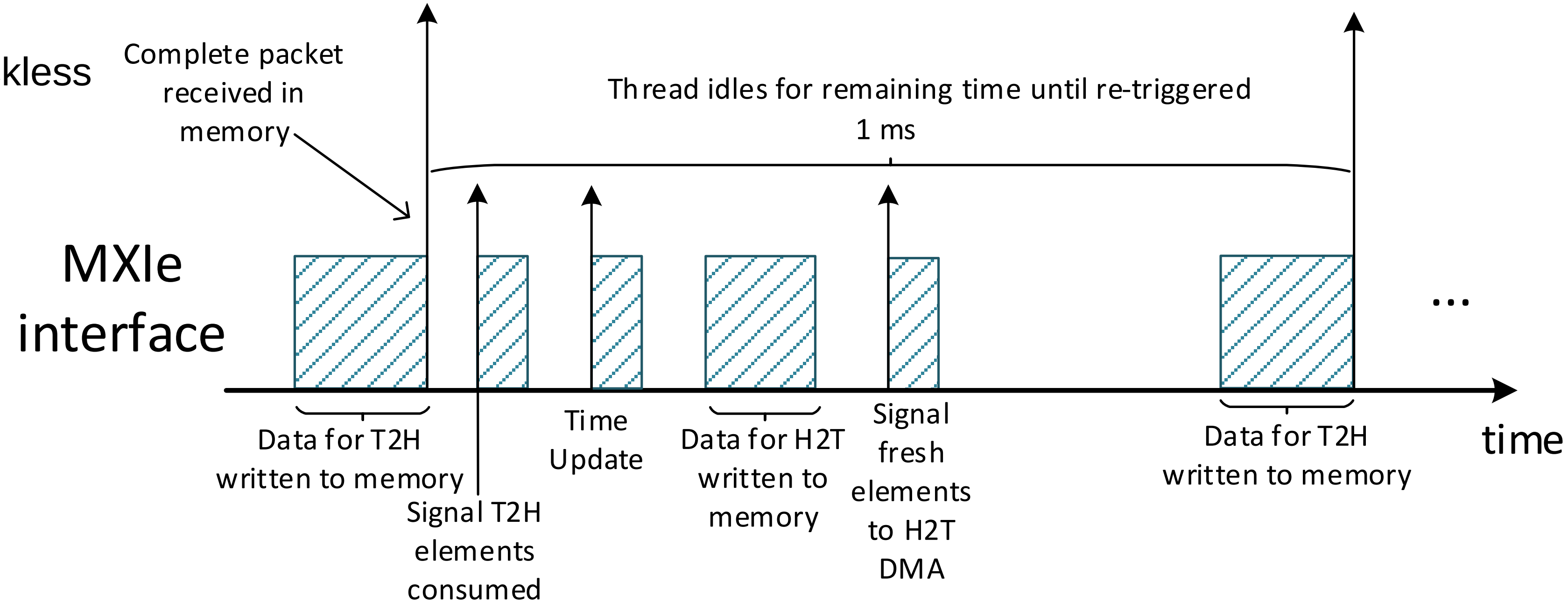
Step 3 – MXIe interface management

Step 3a: MARTe2 application employs MXIe temporal segregation

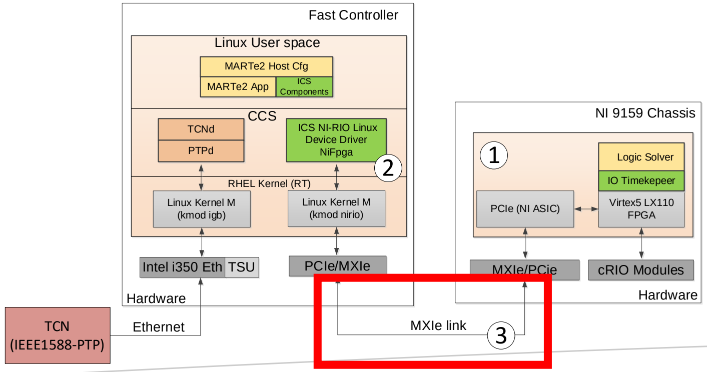
- There is no possibility of physical segregation in the MXIe, use temporary segregation.
- Perform operations separated in time in a controlled manner minimizing jitter.

Step 3b: Create a custom Real Time Red Hat tuned profile

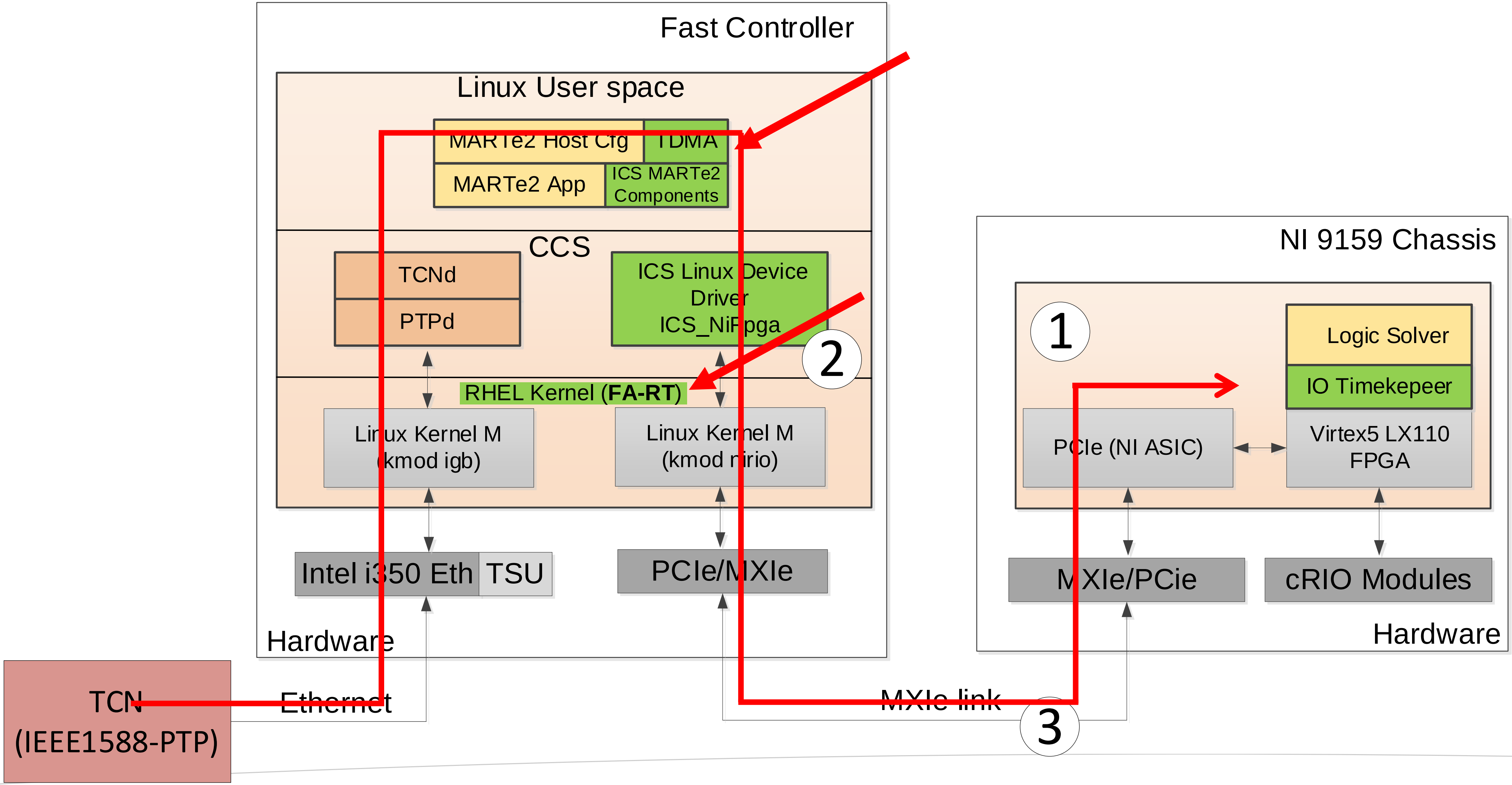
- Execute the host applications on a tickless kernel and isolations CPUs



Complete packet received in memory
Thread idles for remaining time until re-triggered
1 ms
MXIe interface
...
Data for T2H written to memory
Signal T2H elements consumed
Time Update
Data for H2T written to memory
Signal fresh elements to H2T DMA
Data for T2H written to memory
time

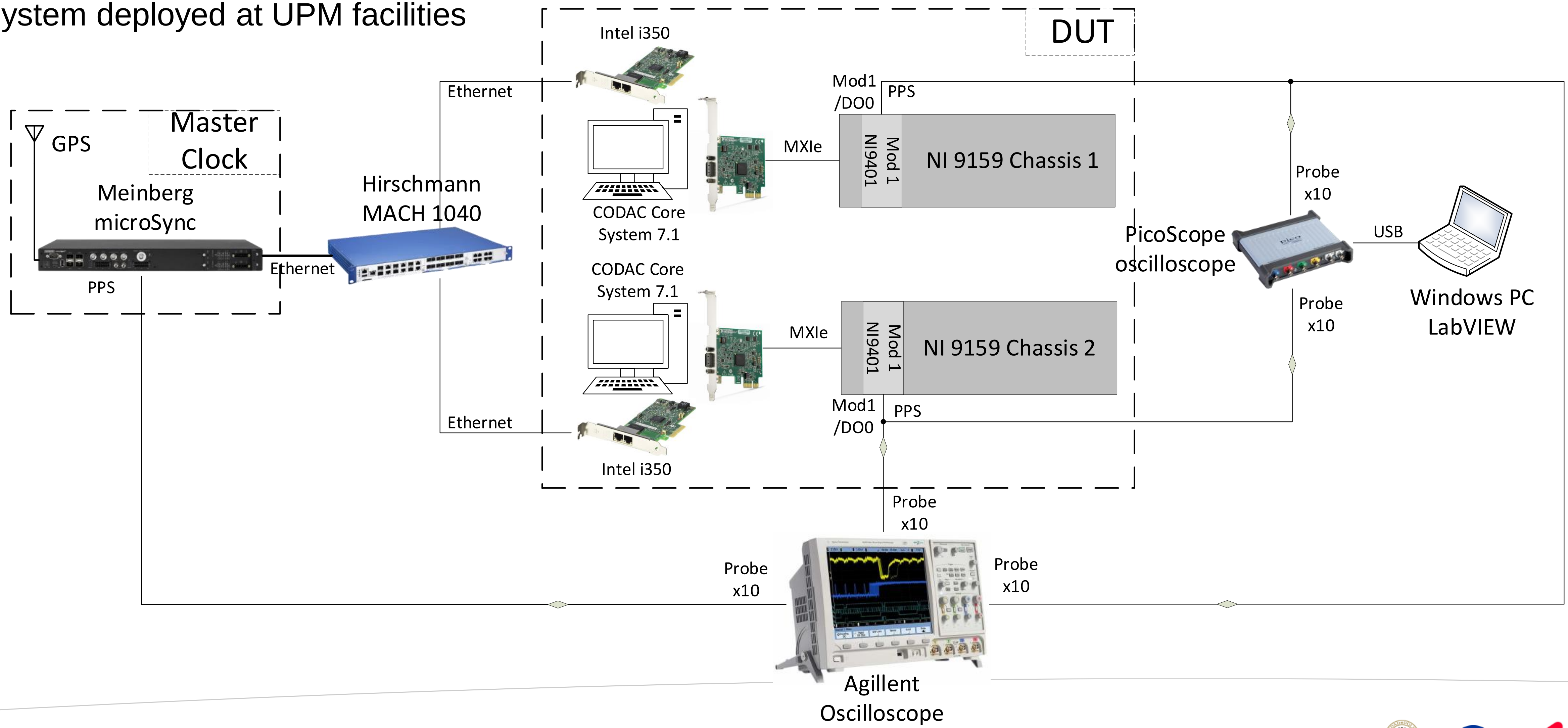


Final Implementation



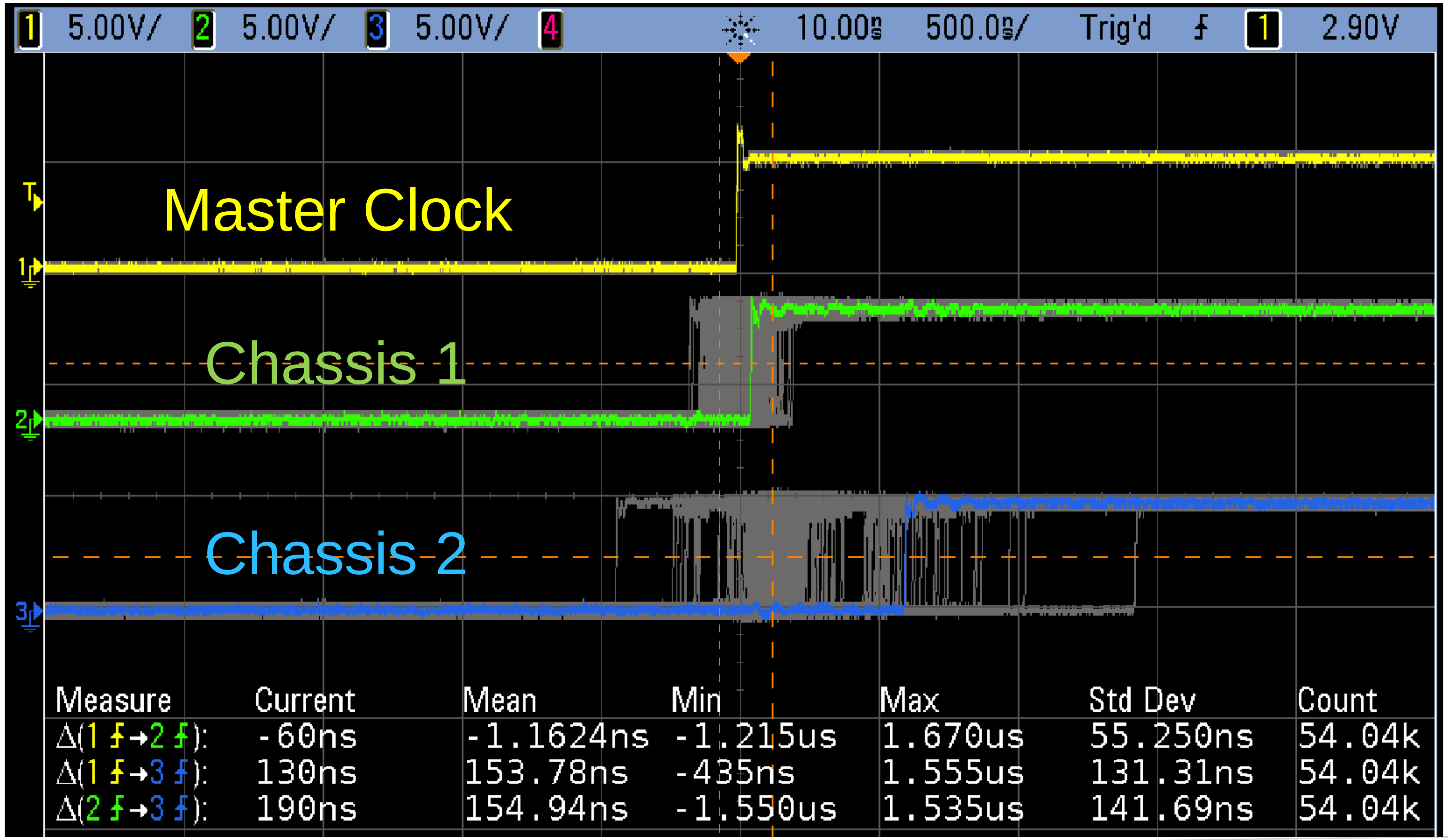
Validation

- System deployed at UPM facilities



Results & Conclusions

- Test over the full system (with the 3 steps integrated) running 15h.
  - System using Time Update and DMAs at 1kHz


1 5.00V/ 2 5.00V/ 3 5.00V/ 4
10.00ns 500.0ns/ Trig'd ƒ 1 2.90V
Master Clock
Chassis 1
Chassis 2
T
1
2
3

```
Measure       Current   Mean        Min        Max       Std Dev    Count
Δ(1 ƒ→2 ƒ):   -60ns     -1.1624ns   -1.215us   1.670us   55.250ns   54.04k
Δ(1 ƒ→3 ƒ):   130ns     153.78ns    -435ns     1.555us   131.31ns   54.04k
Δ(2 ƒ→3 ƒ):   190ns     154.94ns    -1.550us   1.535us   141.69ns   54.04k
```

	Std. Dev	Worst-case
Master Clock – Chassis 1	55.25 ns	1698 ns
Master Clock – Chassis 2	131.31 ns	1424 ns
Chassis 1 – Chassis 2	141.69 ns	1705 ns

Requirements achieved!

Thank you!

Assessing NI FPGA-based platform with MXIe interface for use in ITER hard real-time investment protection applications

D. Karkinsky[1], A. Marqueta[1], E. Barrera[2], **I. García-Siguero**[2], J. Varlec[3], M. Ruiz[2], M. Sekoranja[3], P. Rebollo[2], and T. Kogovsek[3]

[1]ITER Organization, St. Paul-lez-Durance, 13067 France.

[2]Instrumentation and Applied Acoustic Research Group (I2A2), Universidad Politécnica de Madrid, Madrid, Spain.

[3]Cosylab, Gerbičeva ulica 64, 1000 Ljubljana, Slovenia.

Design and implement the final system and prepare it for commissioning.

Responsible: Damien.karkinsky@iter.org

Presenter: i.gsiguero@i2a2.upm.es

china eu india japan korea russia usa

ACKNOWLEDGEMENTS

This work was supported under grant PID2022-137680OB-C33 funded by MCIN/AEI/10.13039/501100011033

COSYLAB

Advancing humanity. Engineering remarkable.

INSTRUMENTATION & APPLIED ACOUSTICS RESEARCH GROUP

MINISTERIO DE CIENCIA E INNOVACIÓN

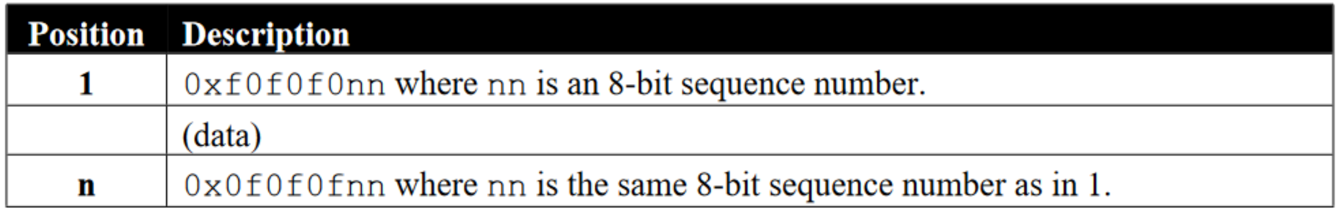
AGENCIA ESTATAL DE INVESTIGACIÓN

POLITÉCNICA

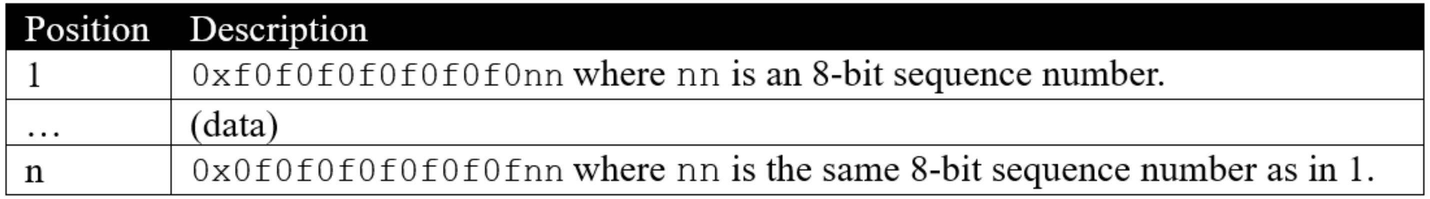
UNIVERSIDAD POLITÉCNICA DE MADRID

BACKUP SLIDES

Packet-based DMAs



Position	Description
1	`0xf0f0f0nn` where `nn` is an 8-bit sequence number.
	(data)
n	`0x0f0f0fnn` where `nn` is the same 8-bit sequence number as in 1.



Position	Description
1	`0xf0f0f0f0f0f0f0nn` where `nn` is an 8-bit sequence number.
…	(data)
n	`0x0f0f0f0f0f0f0fnn` where `nn` is the same 8-bit sequence number as in 1.