

ROD Firmware recap

Nico Giangiacomi

Università di Bologna, Dipartimento di Fisica e Astronomia
Istituto Nazionale di Fisica Nucleare - sezione Bologna

nico.giangiacomi@bo.infn.it

January 29, 2019



1 New development

- Firmware clean-up
- Linux Kernel and new developments
- Smart L1ID forwarding

2 Maintenance

Cleaning the firmware

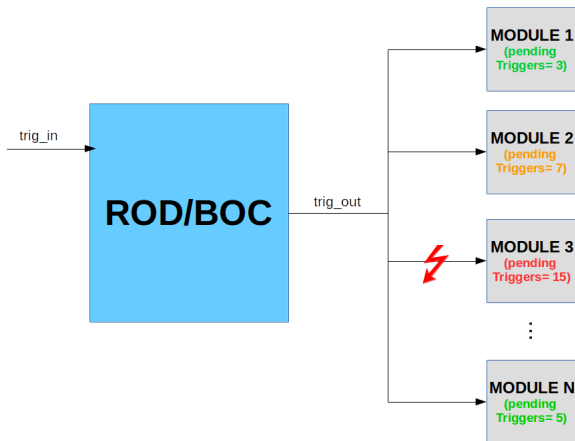
- probably the **highest priority** task right now
 - 1 strongly needed
 - 2 will allow much faster future implementation
 - 3 will guarantee higher stability
- Microblaze **should** be removed from data-taking firmware → **two different firmwares** for calibration and data-taking
 - 1 firmware loading using **Linux kernel** will be **much faster** (~ 1 minute and in parallel over all the boards)
- **work ongoing** → related to new developments e.g. on Linux Kernel
- currently only **Windows** systems are guaranteed for firmware generation

New features

- PRM firmware can be reloaded from PROM **without** touching the JTAG chain (**ICAP**) → very useful in case it gets stuck
- similar mechanism applies to **Slave** FPGAs
- Linux drivers ready for **SPI, Serial Port** and **Slave Registers** → check past presentations for more details
 - 1 https://indico.cern.ch/event/776936/contributions/3231283/attachments/1760027/2855598/Some_news_fromRODside27_11_2018BIS.pdf
 - 2 <https://indico.cern.ch/event/778644/contributions/3240043/attachments/1765256/2865753/DriversForROD2.pdf>

Smart L1ID forwarding (1)

- If a front-end module receives a trigger *before* it finishes to process the previous event, it can store the new event in a **pending trigger** buffer → the buffer can store **MAX 16** events;
- if the number of **pending trigger** is too big, the front-end will **not be able to send information** regarding that trigger;
- to **relieve** the front-end from the pressure, the ROD firmware can decide **not to propagate the trigger** signal.



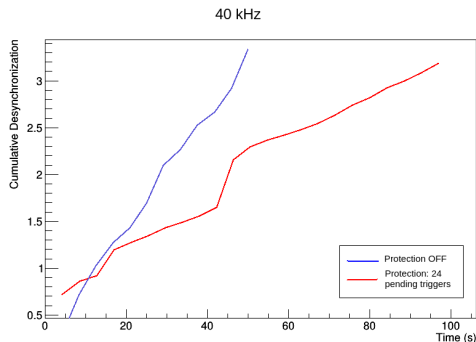
Smart L1ID forwarding (2)

For the smart L1ID forwarding mechanism, the ROD **must**:

- ❶ **Monitor** the amount of pending triggers per each module
 - a pending trigger counter has been added for each module
 - if a trigger arrives, **pending counter + 1**
 - if a module header arrives, **pending counter - 1**
- ❷ **Compare** the amount of pending triggers with a **threshold**
 - **new register** 0x814 to set the threshold \rightarrow 5 bits register
 - **default value:** 0x1F = 31 \rightarrow never activated by construction
 - if number of pending trigger > threshold, trigger is **not forwarded** (**ECR** and **BCR** are forwarded)
- ❸ **Insert an empty event** if the trigger was not propagated to the module
 - if trigger is not forwarded, an empty event is generated for that module.
header: **2xxxbaad**
trailer: **4080xbad**
 - the event must be inserted in the **proper fragment** to keep **synchronization**
- ❹ **Correct** the L1ID counter of the module
 - if the trigger was not propagated to a module, its **internal L1ID counter** will not be in line with the other modules
 - the ROD firmware must **correct** this counter to keep synchronization.

Status

- all steps completed
- **no corruption** showed in **SR1** tests
- **corruption** appeared in **PIT** test → probably related to **firmware stability** problems
- performing **badly** when threshold is set to ~ 16
- performing **good** when threshold is set to $20 \leq \text{Threshold} \leq 32$
- **pending trigger monitoring** algorithm needs improvement



Setup

- SR1 setup very useful for real detector scenario development → desynchronization, smart L1 forwarding...
- simulation system *ready* and *stable*, already proved to be very useful
- local setup in Bologna still has some issues