

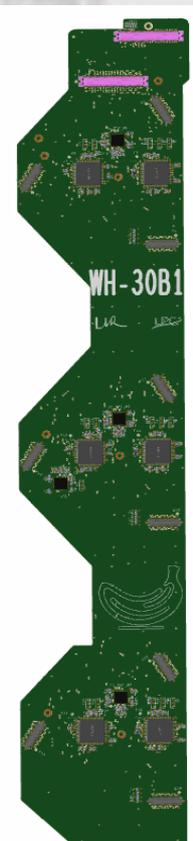
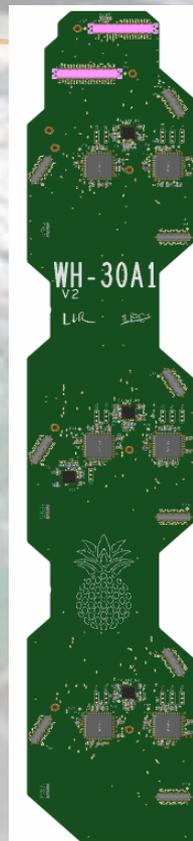
Electronique CMS HGCAL

Florence Beaujean Delaune, Rémi Guillaumat, Olivier Le Dortz

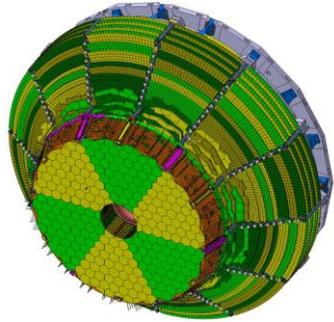
Journée des groupes techniques du LLR, 3 juillet 2025

Electronique Front-End: Wagons HD

Rémi G., Olivier L.D.
Marie-Lise M. (LPCA)

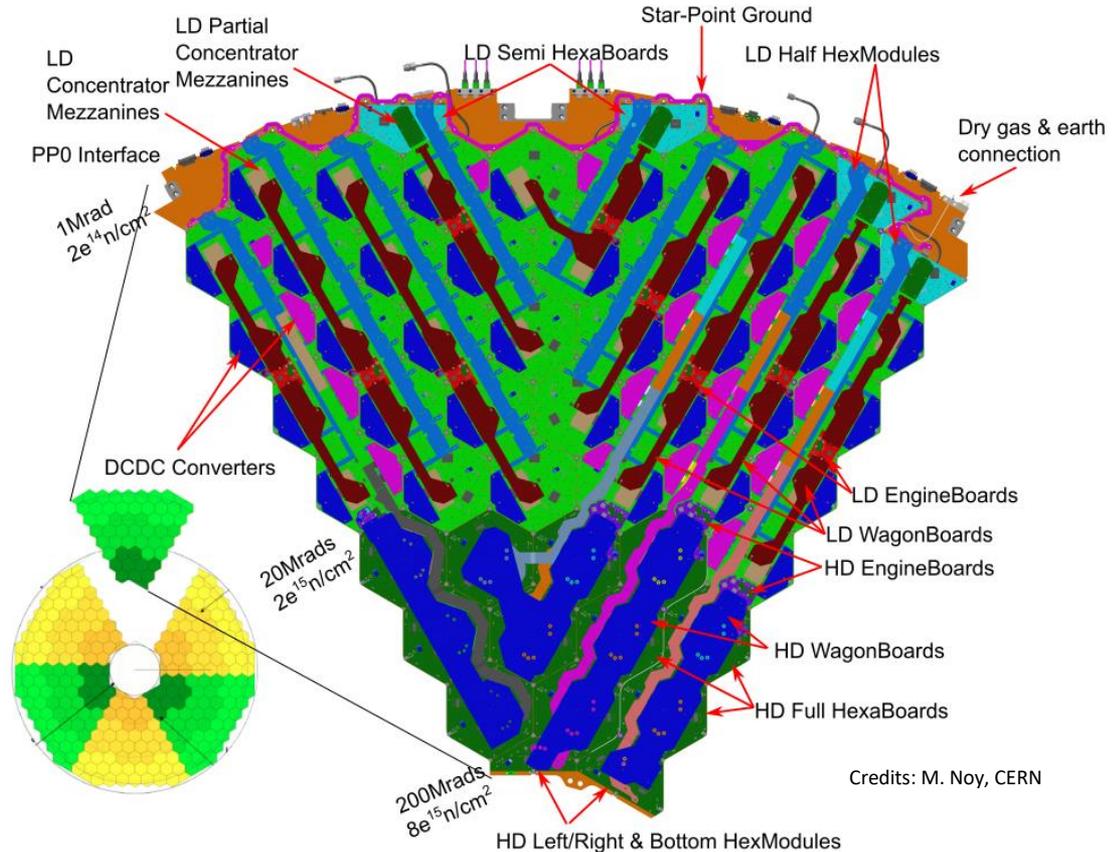


HGCAL Electronics System Overview (SI Region)



Vue d'une cassette HGCal
CE-E (Double Sided)

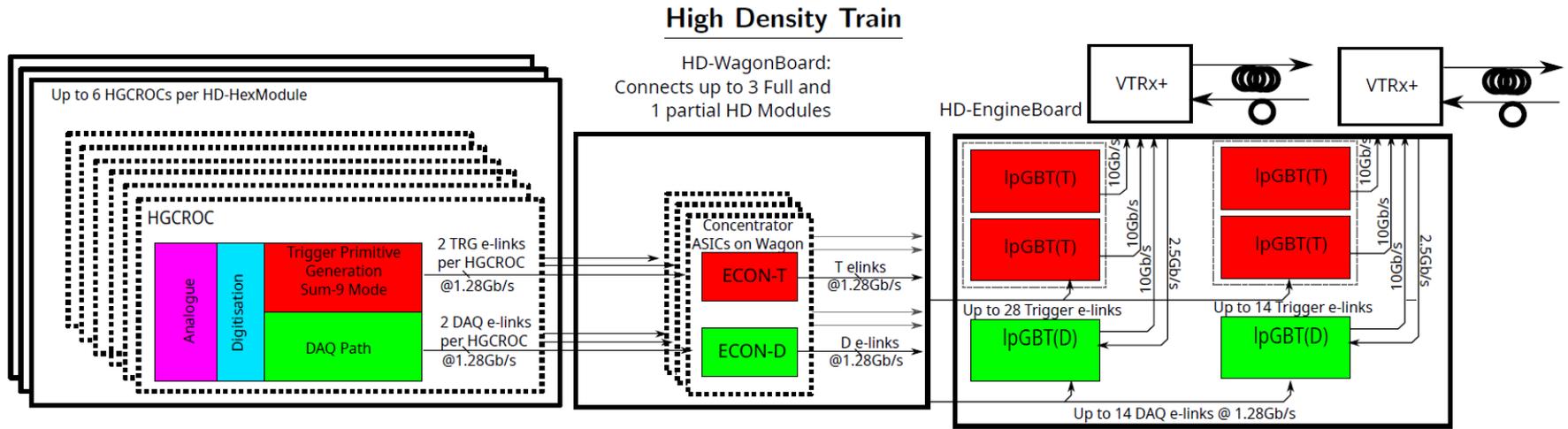
Modules (hexaboards) ⇔ Wagons ⇔ Engines ⇔ BE



Credits: M. Noy, CERN

Rôle des Wagons HD

- Envoi des données trigger et données acceptées des hexaboards vers le « HD engine » (liens 1.28 Gbits/s)
- Transmission informations d'horloge et de contrôle rapide (FC) du « HD Engine » vers les « hexaboards » (320 MHz)
- Configuration de tous les ASICs front-end (par I2C)



Représentation du chemin de données Front-End

Credits: M. Noy, CERN

Historique Wagons HD au LLR

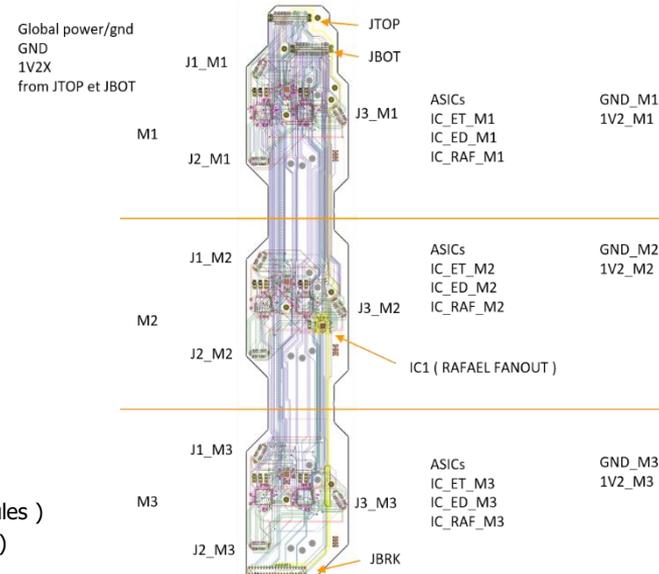
- Décembre 2022: appel de la collaboration HGICAL (K. Gill, subsystem manager & P. Aspell responsable électronique) pour la prise en charge du design des Wagons HD (ou de certaines variantes parmi les 8 à concevoir)
- Début 2023: discussions techniques initiales CERN (P.Aspell, M.Noy) / Université Minnesota (N.Strobbe)/ LLR (Franck, Éric)
- Avril 2023: accord que le LLR étudierait la faisabilité en amorçant le design d'un wagon
- A mon arrivée au LLR (mai 2023), prise en charge de la conception d'une variante du Wagon HD (3 modules), selon l'organisation suivante:
 - Schéma électronique fourni par l'UMN (format Altium)
 - Au LLR:
 - ◆ Conversion schématique Altium => Cadence Allegro, remise en forme...
 - ◆ Collaboration avec le service CAO PCB du LPCA (Marie-Lise Mercier) pour le placement-routage
 - ◆ Interfaces avec le CERN et l'UMN pour les validations mécaniques / électroniques
 - Après validation par la collaboration du design
 - ◆ Fabrication du PCB et assemblage par l'UMN
 - ◆ Tests électriques et validations au CERN

Première variante WH-30A1 (Ananas)



- Novembre 2023:
 - Finalisation du design en 10 couches
- Décembre 2023: « Mini-revue » du design (UMN, CERN)
 - Mécanique quasi validée (quelques modifications de positionnement à appliquer)
 - Tests d'intégrité de signal (lignes de transmission notamment) ont révélé des soucis sur certaines lignes. Cela nécessite une reprise de l'empilement. Passage à 12 couches
- Février=>Mai 2024: routage 12 couches + revue
- Juillet 2024: fourniture fichiers de production fabrication prototype par UMN puis tests
- Octobre 2024: découverte d'un bug sur le dessin initial => correctif urgent => fourniture des fichiers [WH-30A1_v2](#)
- Entre-temps, correction « hardware » du bug sur les wagons v1 pour continuer les tests système au CERN

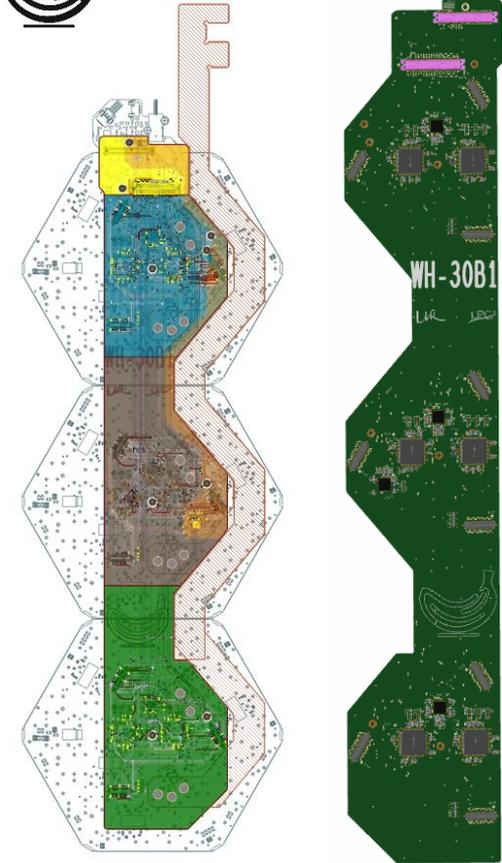
- 500x87mm²
- Components mounted on the BOTTOM layer
- 4 power and ground domains (Engine, 3 modules)
- 200~ differential pairs (320 MHz or 1.28 Gb/s)

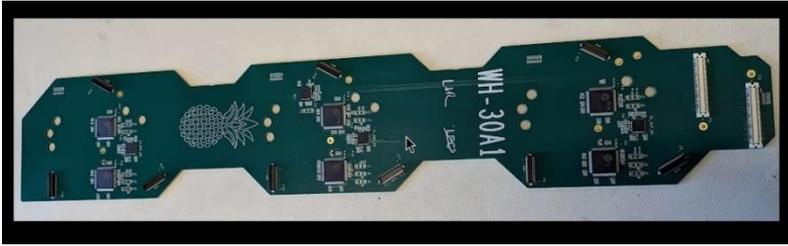


Seconde variante WH-30B1 (Banane)

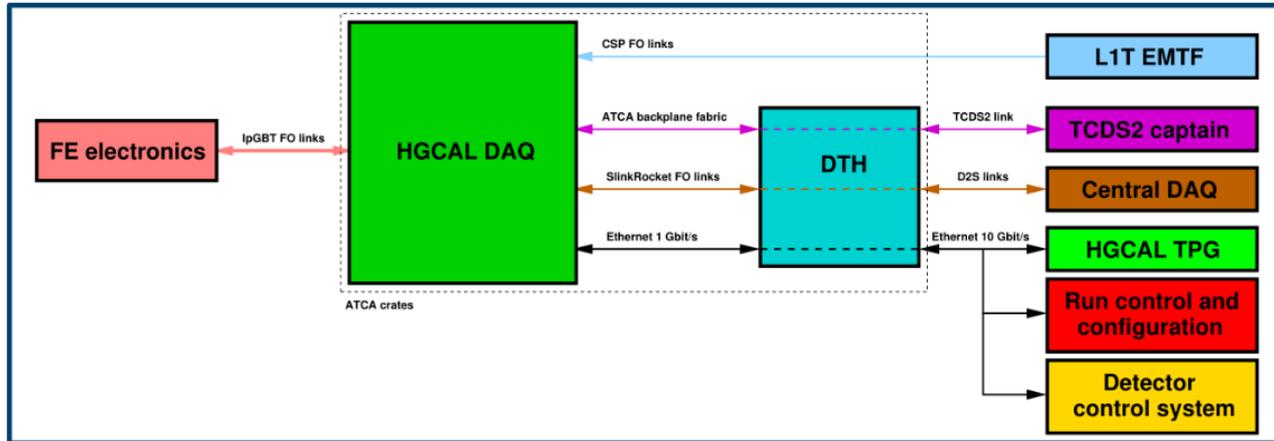


- Août 2024:
 - adaptation du schéma
 - mise à jour géométrie
(puis correction du schéma en octobre à cause du bug découvert sur la 1^{ère} variante)
- Novembre 2024: routage terminé
- Décembre 2024:
 - mini-revue
 - correction des points les plus critiques par Rémi et fourniture des fichiers finaux



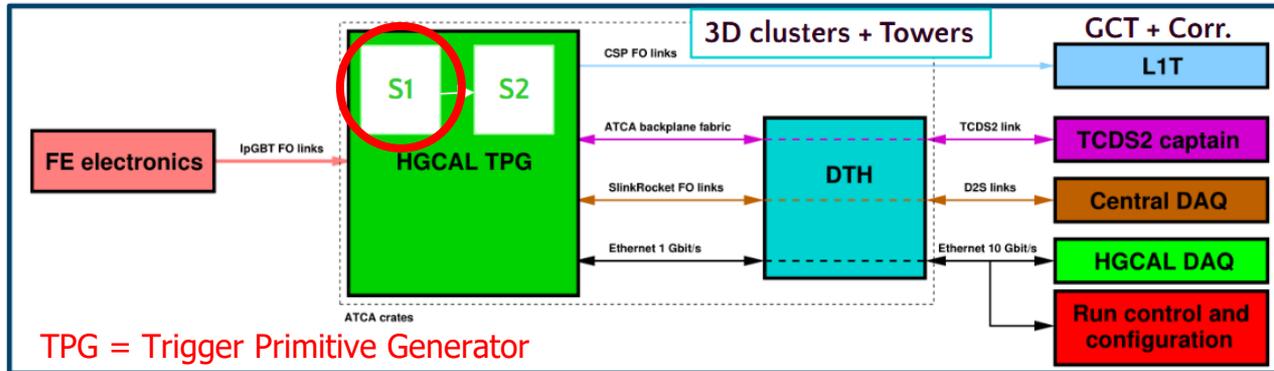


Système Back-end de lecture HGCAL



Flux de données DATA

Données conservées par le Trigger de niveau 1



TPG = Trigger Primitive Generator

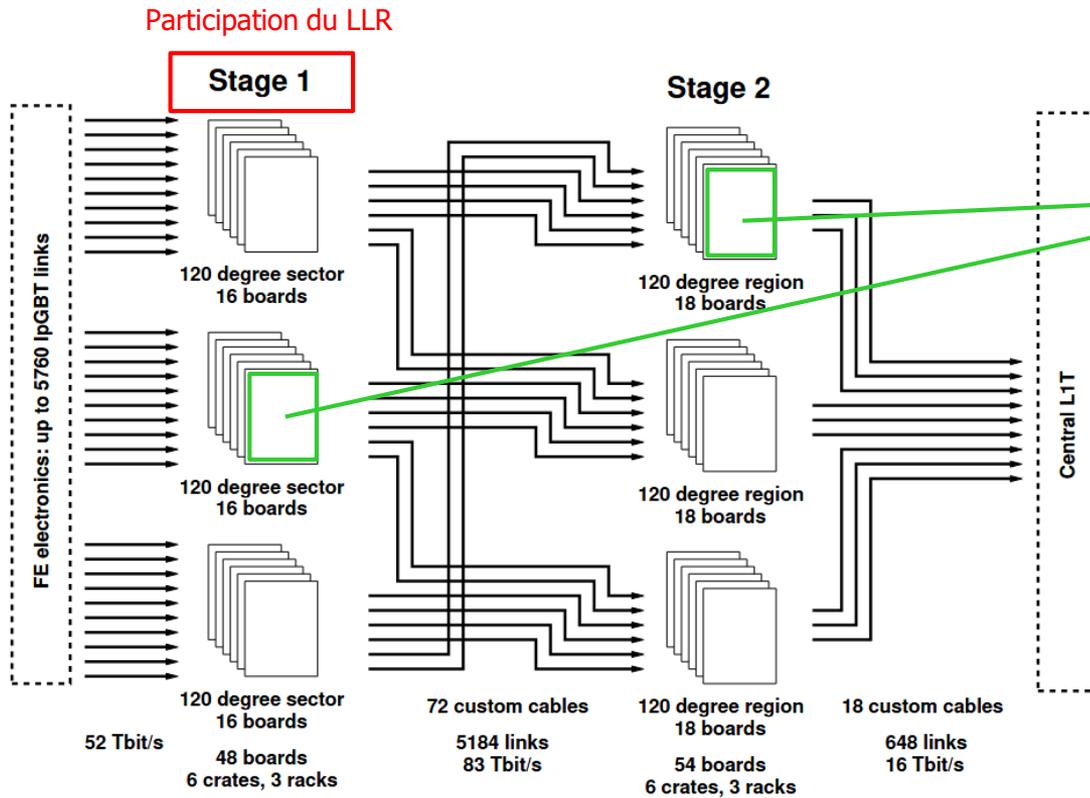
Flux de données Trigger

Toutes les 25 ns
Alimente le Trigger de niveau 1

Le LLR est impliqué dans le
« TPG Stage 1 »

Architecture globale d'un endcap de TPG

Credits: ICL



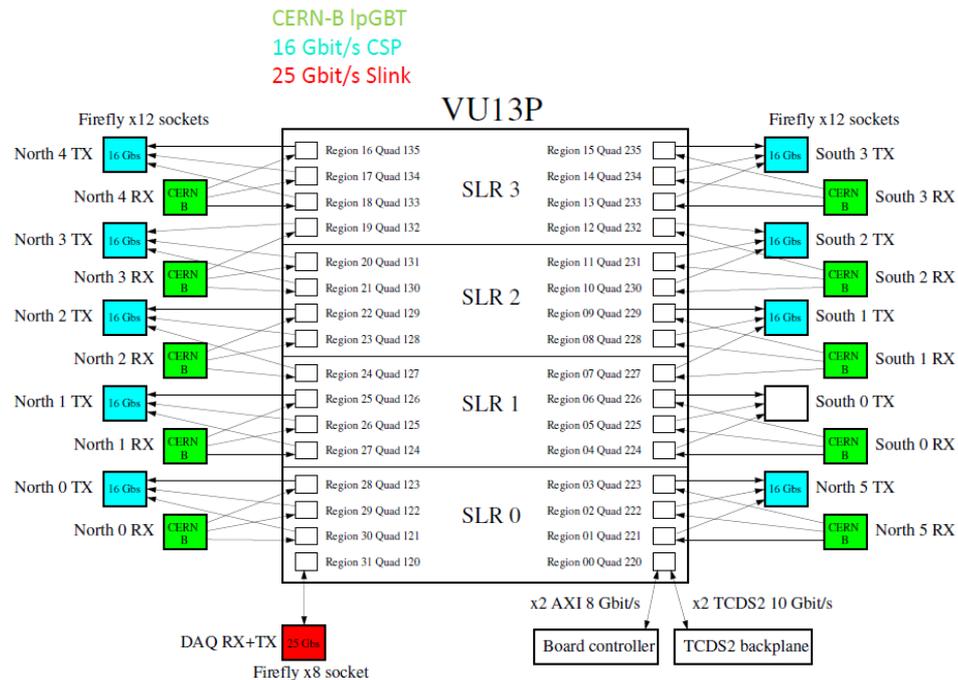
Cartes = Serenity S

A base de FPGA Virtex Ultrascale+
(VU13P)

Jusqu'à 128 entrées et 128 sorties
optiques 16/25 Gb/s

FPGA du TPG Stage 1: Entrées/Sorties

- **120 entrées (RX)** issues du front-end
(7x32 bits @ 40 MHz chacune)
- **108 sorties (TX)** vers le TPG Stage 2
 - 6*64 bits @ 40 MHz chacune
 - Multiplexées en temps (facteur 18)
 - 18 groupes de 6 sorties
 - 1 Evt = 18x6 mots de 64b
- **1 sortie « Common Readout »** : à chaque évènement accepté par le Trigger Niveau 1, 1 packet contenant un « aperçu » des entrées et sorties de cet instant



Credits: P. Dauncey, ICL

Schéma du firmware TPG Stage 1

Un développement intégré dans un environnement complet (EMP framework)

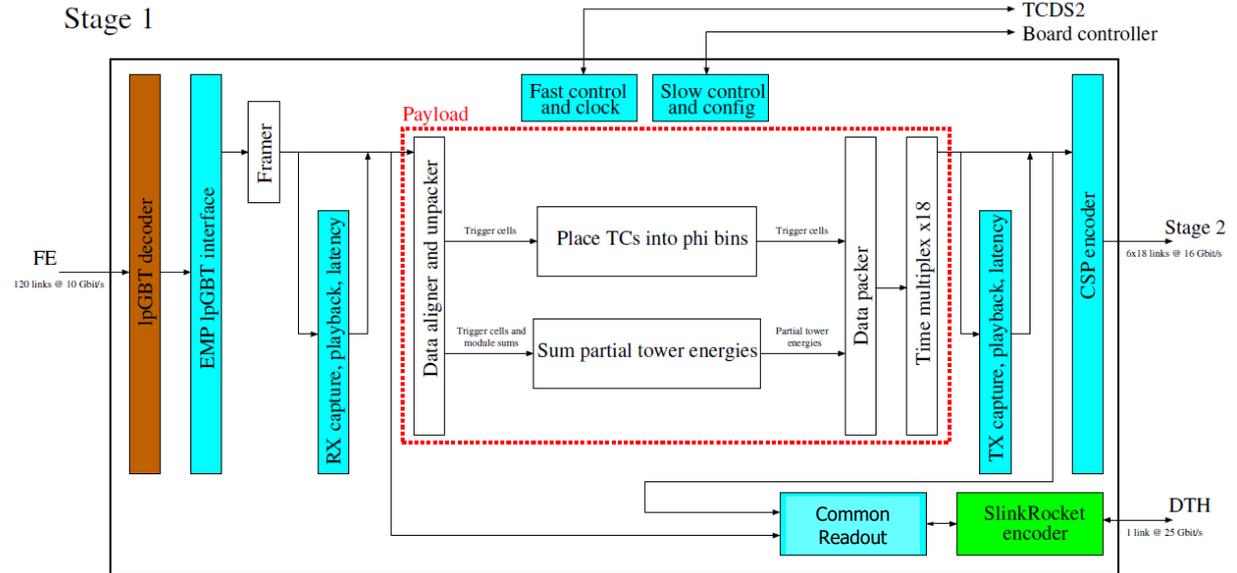
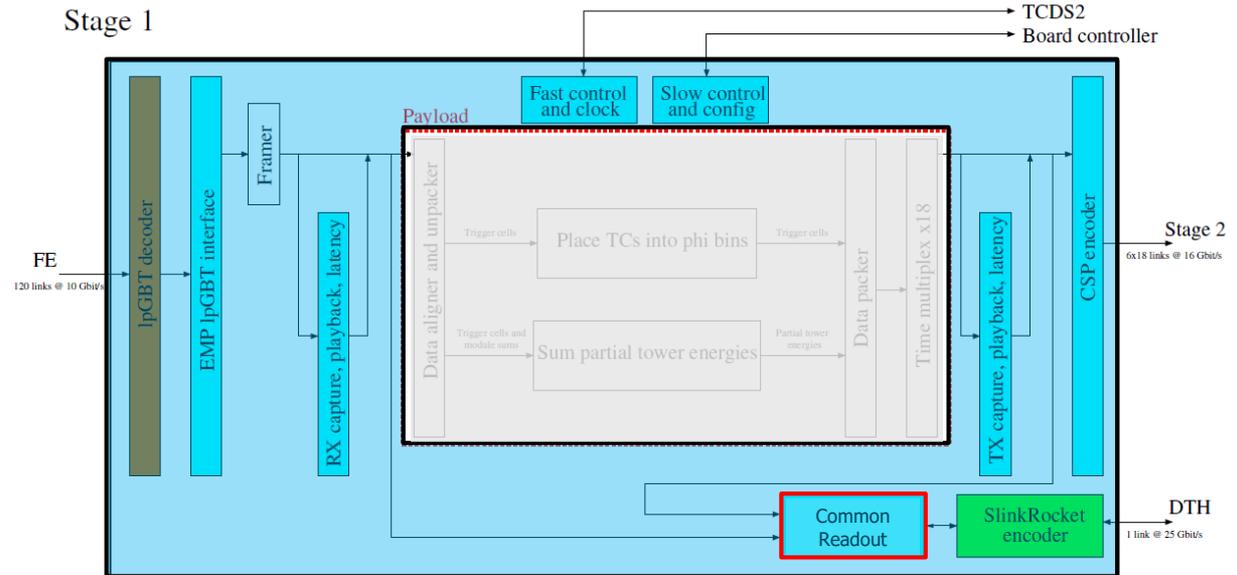


Schéma du Firmware TPG Stage 1

Un développement intégré dans un environnement complet (EMP framework)

- **EMP wrapper** (développement commun pour toutes les cartes Serenity):
 - Configuration/Timing
 - Desér/Sérialisation IOs
 - Emulation entrées et capture des sorties via « EMP buffers »

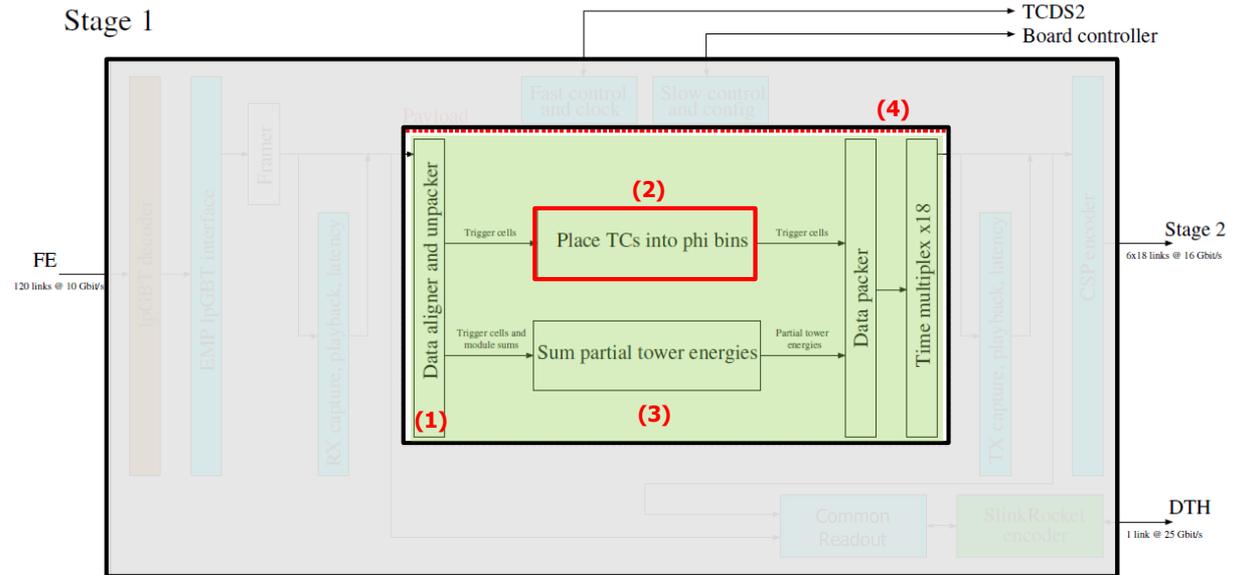


Common Readout:
Développement spécifique HGCAL
Imperial College London et Olivier

Schéma du Firmware TPG Stage 1

Un développement intégré dans un environnement complet (EMP framework)

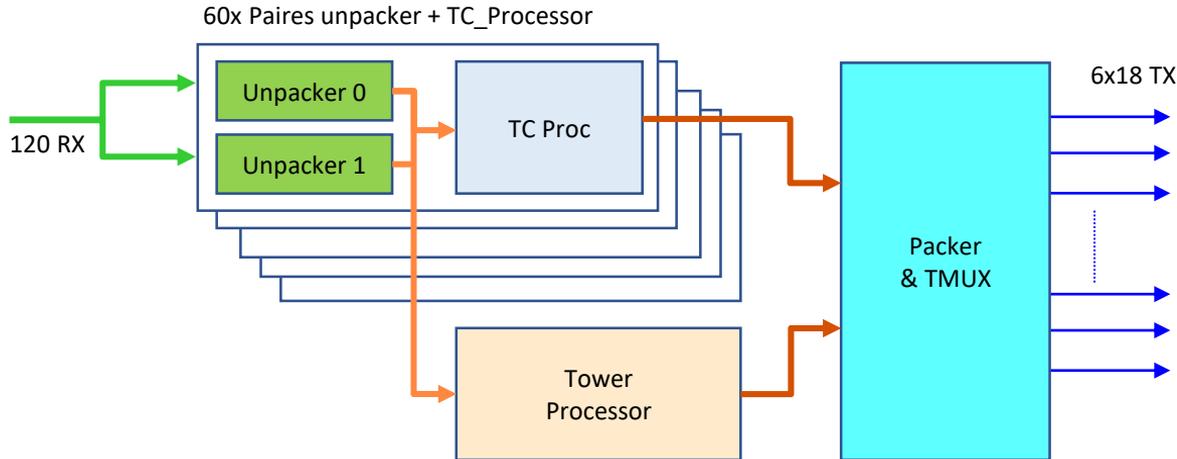
- EMP wrapper (développement commun pour toutes les cartes Serenity):
 - Configuration/Timing
 - Desér/Sérialisation IOs
 - Emulation entrées et capture des sorties via « EMP buffers »
- « **Payload** », unité de traitement spécifique au TPG Stage 1:
 - Horloge à 320 MHz
 - Développement collaboratif:
 - Unpacker (1) Imperial College
 - **Trigger Cell Processor (2)** Florence
 - Tower Processor (3) Split
 - Packer + TMUX (4) Split+ICL



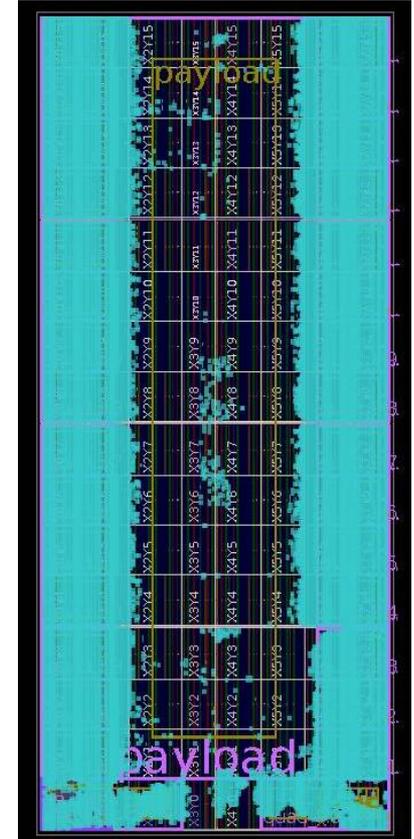
Intégration du Firmware

- Bloc TC_Processor développé/testé par Florence
- Désormais, engagée dans l'intégration du firmware complet
 - Assemblage de tous les blocs de la payload du TPG Stage 1

EMP Wrapper:
~30% ressources FPGA

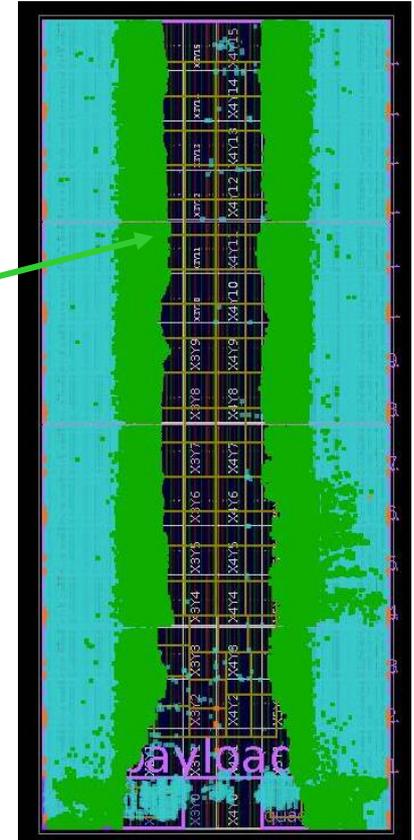
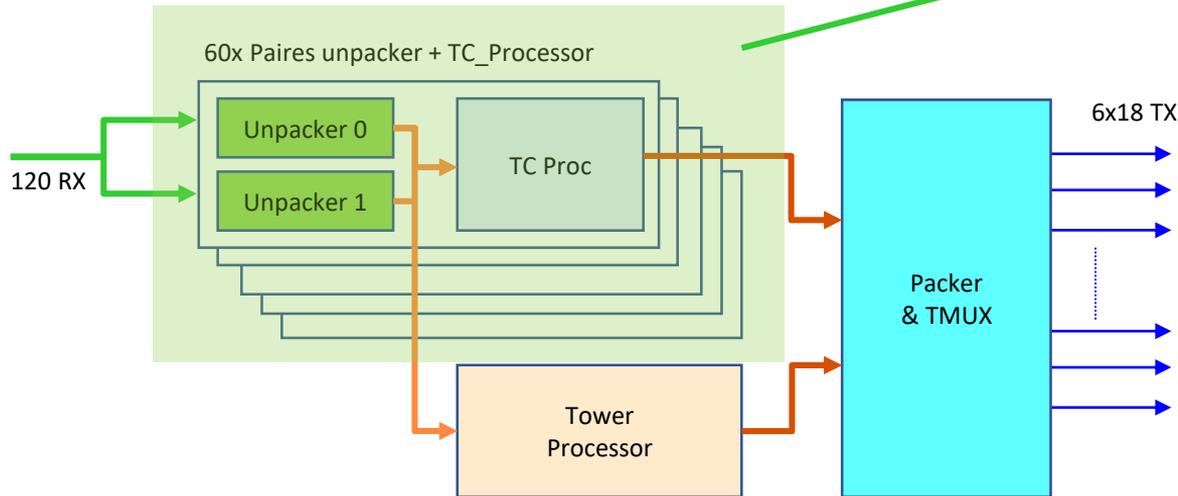


Ressources du FPGA, Payload vide



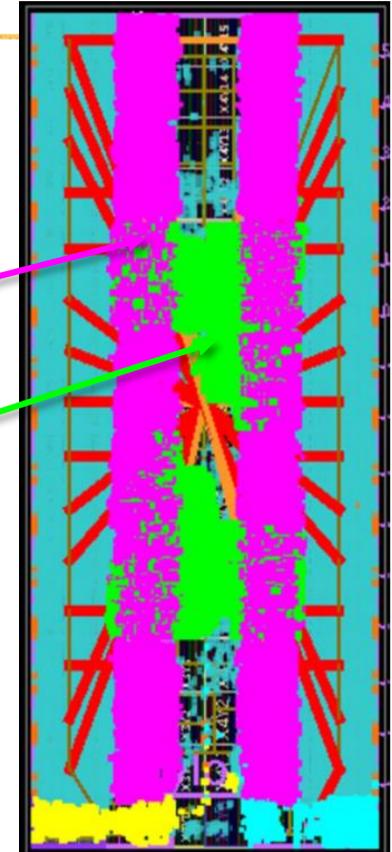
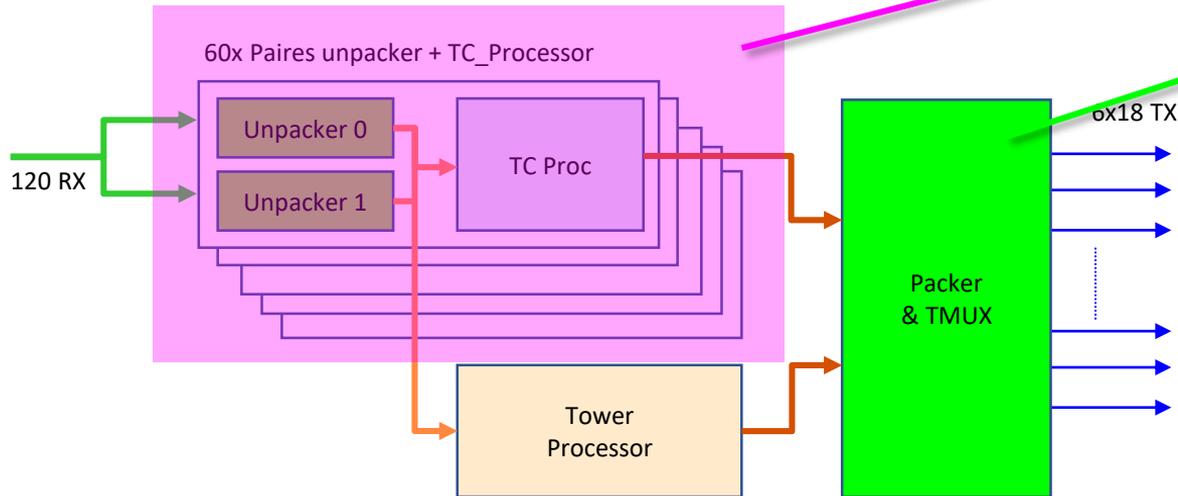
Intégration du Firmware

- Bloc TC_Processor développé/testé par Florence
- Désormais, engagée dans l'intégration du firmware complet
 - Assemblage de tous les blocs de la payload du TPG Stage 1
- Firmware partiel Unpacker + TC processor



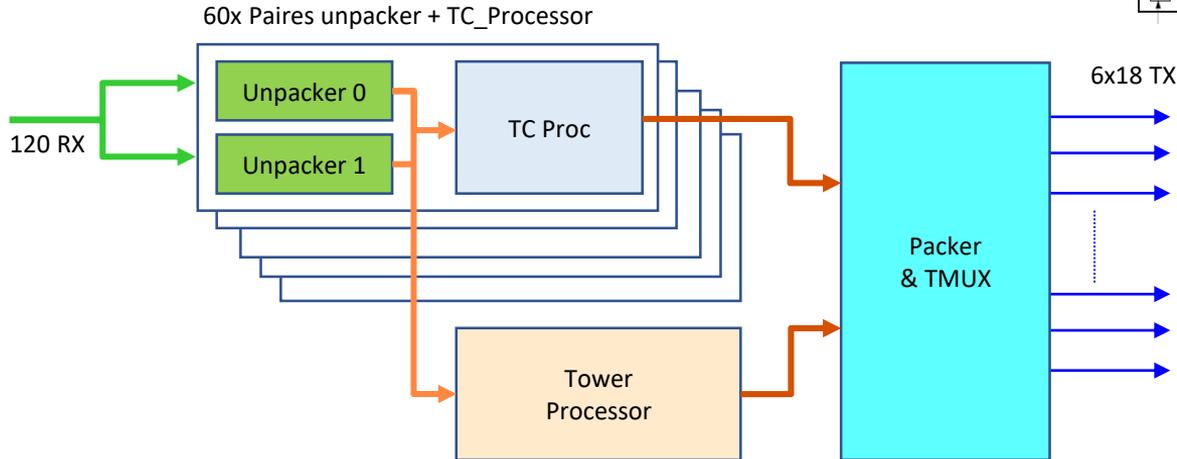
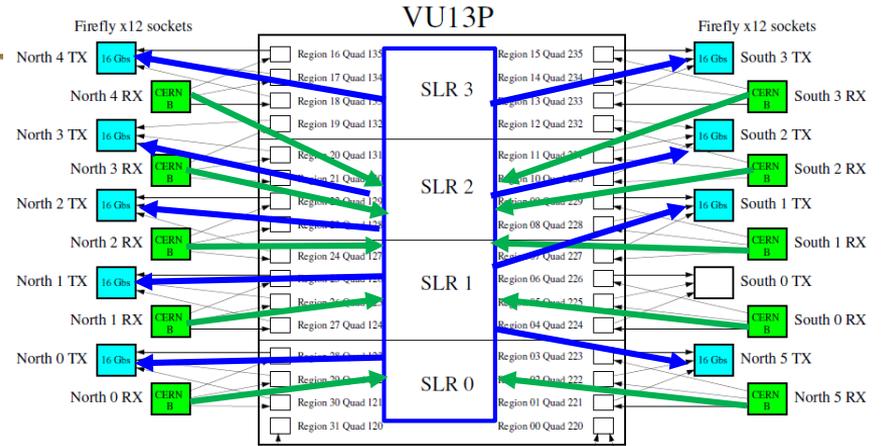
Intégration du Firmware

- Bloc TC_Processor développé/testé par Florence
- Désormais, engagée dans l'intégration du firmware complet
 - Assemblage de tous les blocs de la payload du TPG Stage 1
- Firmware partiel Unpacker + TC processor + Packer/TMUX



Intégration du Firmware

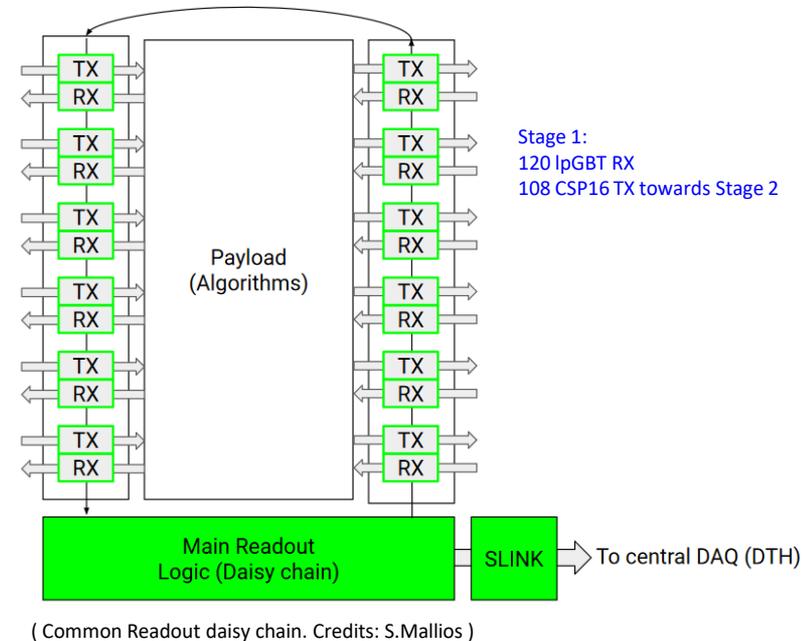
- Firmware total avec Tower Processor: en cours d'optimisation
- Un environnement très contraignant:
 - >60% ressources LUT du FPGA
 - Horloge rapide 320 MHz
 - Temps de compilation > 6-7 heures
 - Tower Processor + Packer = goulet d'étranglement



Bloc « Common Readout »

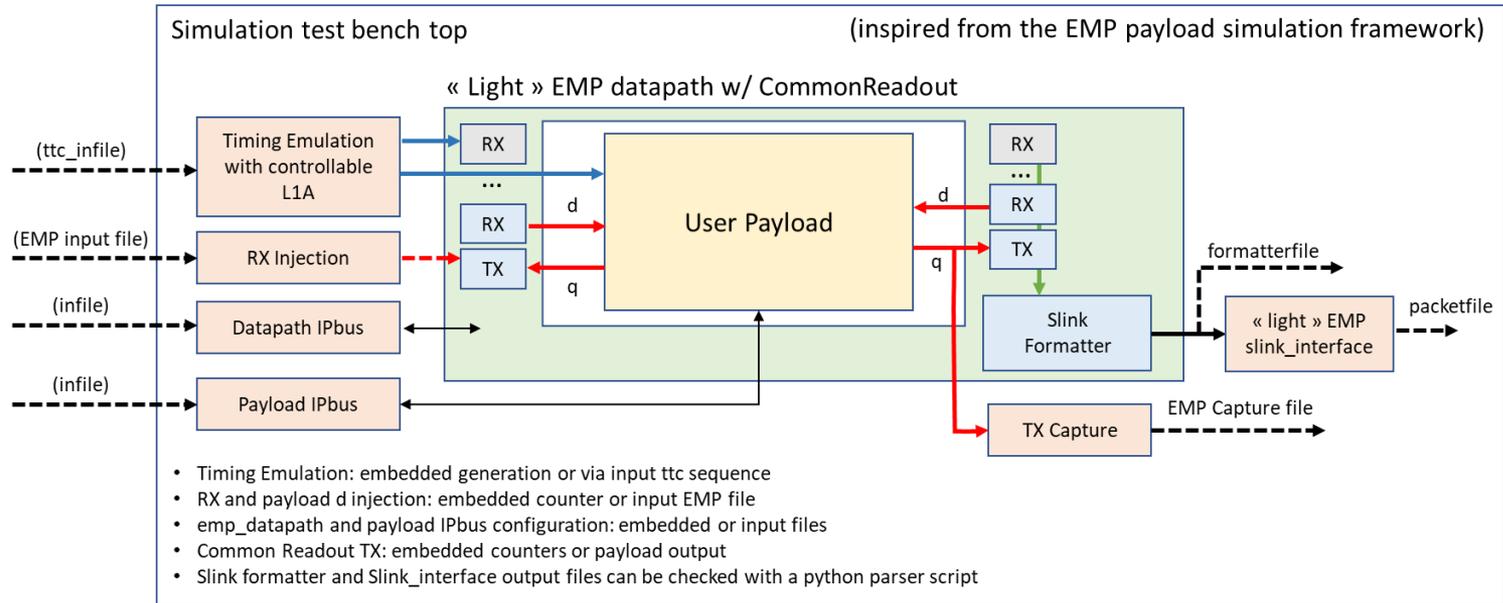
- Le Common Readout (CRO) est une fonctionnalité insérée dans l'EMP wrapper (périphérie englobant la payload du firmware). Développement initié par S.Mallios (ICL), rejoint ensuite par Olivier
- 1 bloc par entrée/sortie du FPGA (Stage 1: 228)
- A chaque trigger L1A, envoi vers un lien série dédié d'un paquet contenant les données reçues et émises sur chaque lien RX et TX (selon configuration)
- Gestion d'un mode étendu:
Envoi des données RX/TX d'une fenêtre temporelle autour de l'instant t_0 du L1A :
$$t_0 - hw \leq t \leq t_0 + hw$$

Paramètre $hw \leq 3$
- Conçu sur une base de « daisy chain » parcourant toute la périphérie du circuit
- Empreinte en ressources FPGA pour le TPG Stage 1:
9% LUT, 7% FF, 13% BRAM, 36% Ultra-RAM



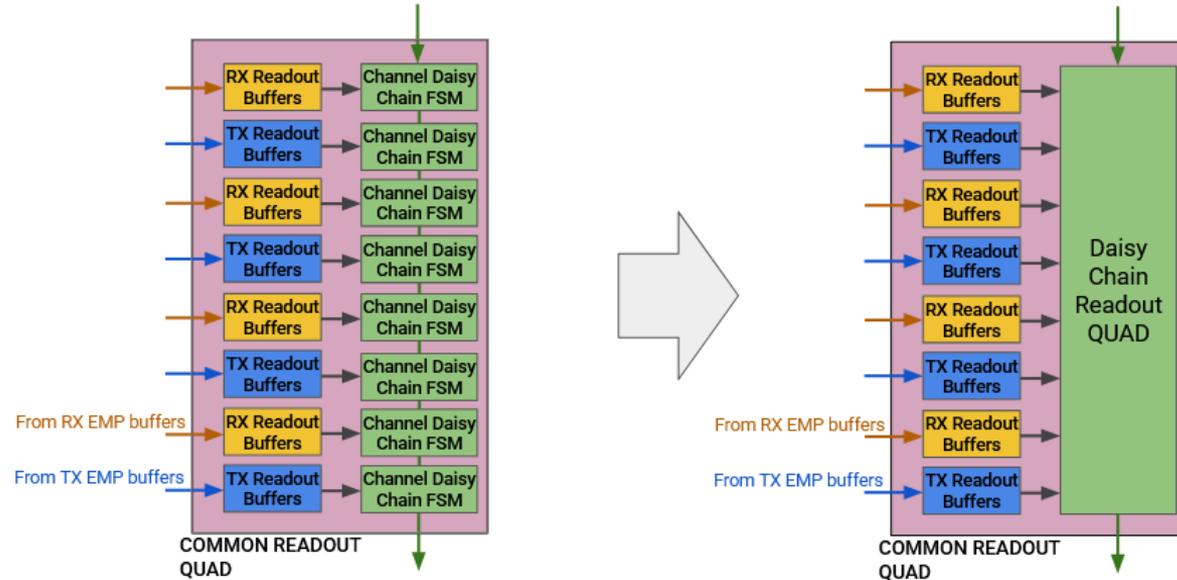
Contribution LLR au « Common Readout »

- Déploiement pour le cas du TPG Stage 1
- Améliorations du code VHDL, débogage. Mise en place de simulation système complète (+ scripts d'analyse python)
- Validation sur hardware



Contribution LLR au « Common Readout »

Optimisation de la machine d'état de la daisy chain.



To speed up the daisy chain propagation and improve the timing specs, factorized the (up to) 8 RX/TX channel FSMs into a unique QUAD FSM

Bloc désormais opérationnel

- ⇒ An empty packet (only the main debug channel data) now takes 6 BXs instead of 37 BXs to traverse the full daisy chain.
- ⇒ Improvement of the critical path (initially located in the busy signal back propagation) and slight reduction of the resource usage.