Outils logiciels pour la conception et la validation d'un processeur électronique complexe:

le trigger L0 à muons de LHCb

(**Pierre-Yves Duval CPPM**)

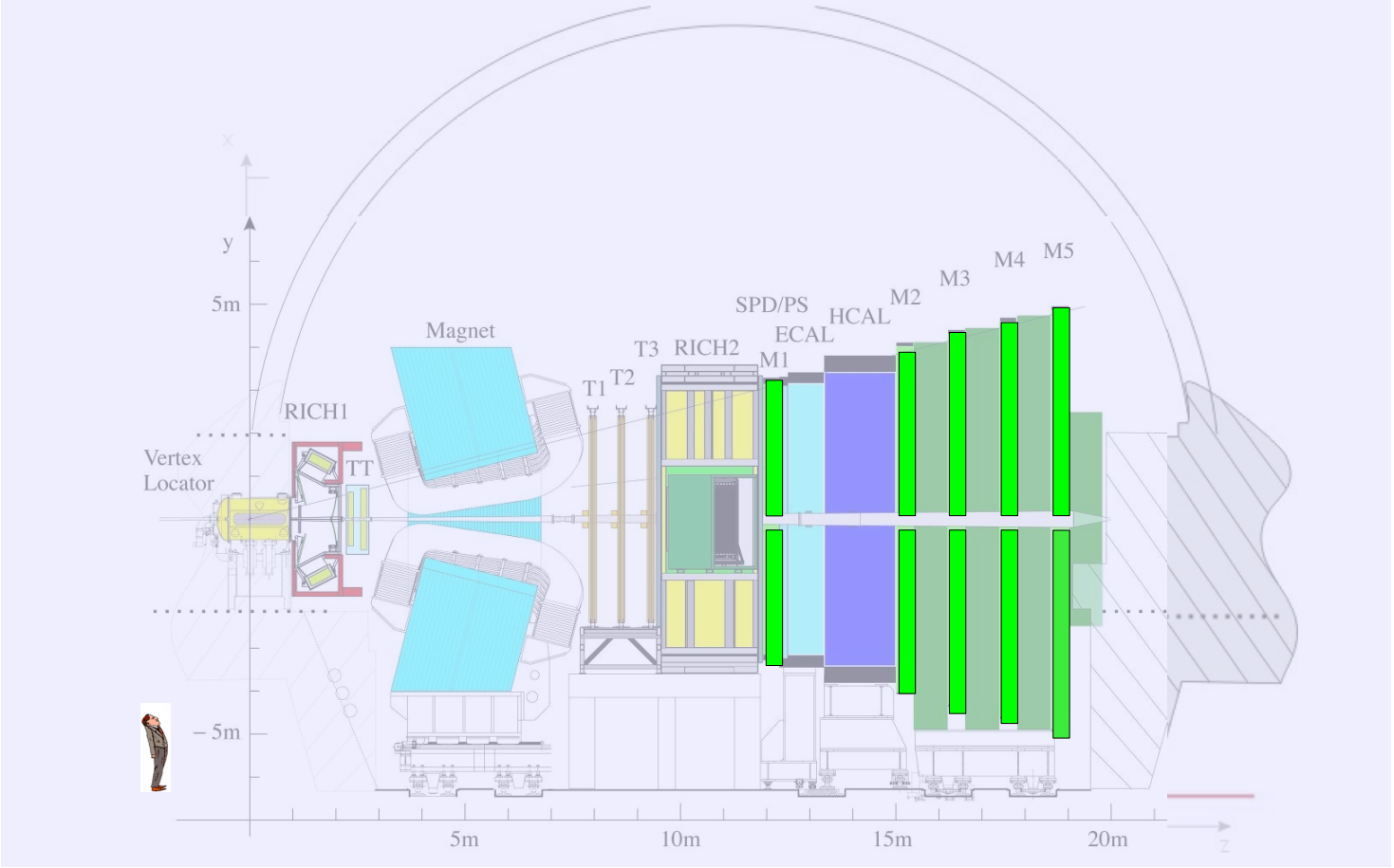
E. Aslanides, J.P Cachemiche, J. Cogan, P.Y. Duval, F. Hachon, R. Le Gac, O. Leroy, F. Marin and A. Tsaregorodsev

Journées informatique IN2P3-IRFU, Obernai Septembre 2008

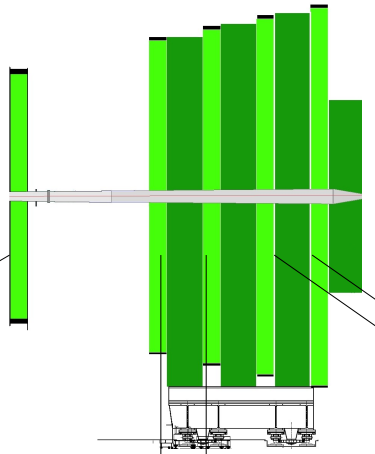
Présentation

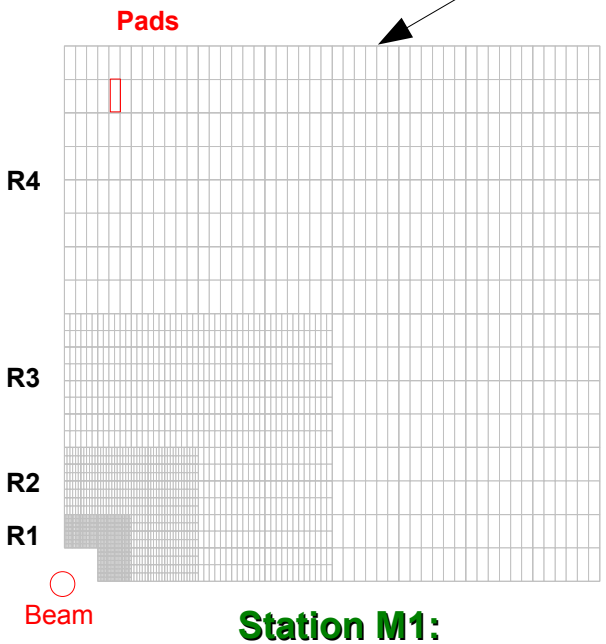
- Spécification fonctionnelle du trigger L0 pour le détecteur de LHCb
- Spécification d'implantation en électronique: le processeur hardware
- Le modèle logiciel de référence en simulation : l'émulateur bit à bit
- Validation du hardware par comparaison au modèle de référence
- Conclusions

Spécifications
Chambres à muons du détecteur LHCb

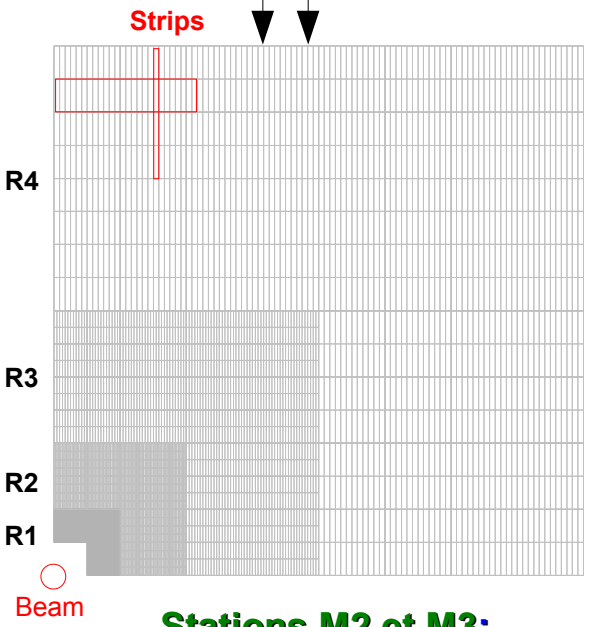


Spécifications
Granularité des 5 stations à muons

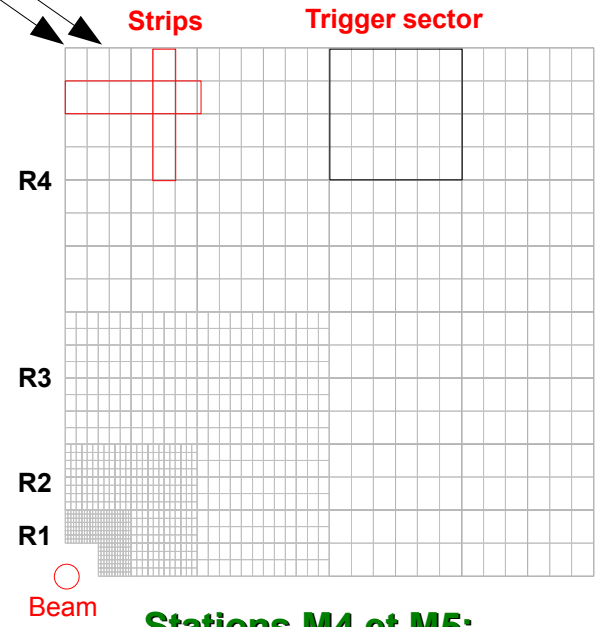




Station M1:
Seulement des PADs

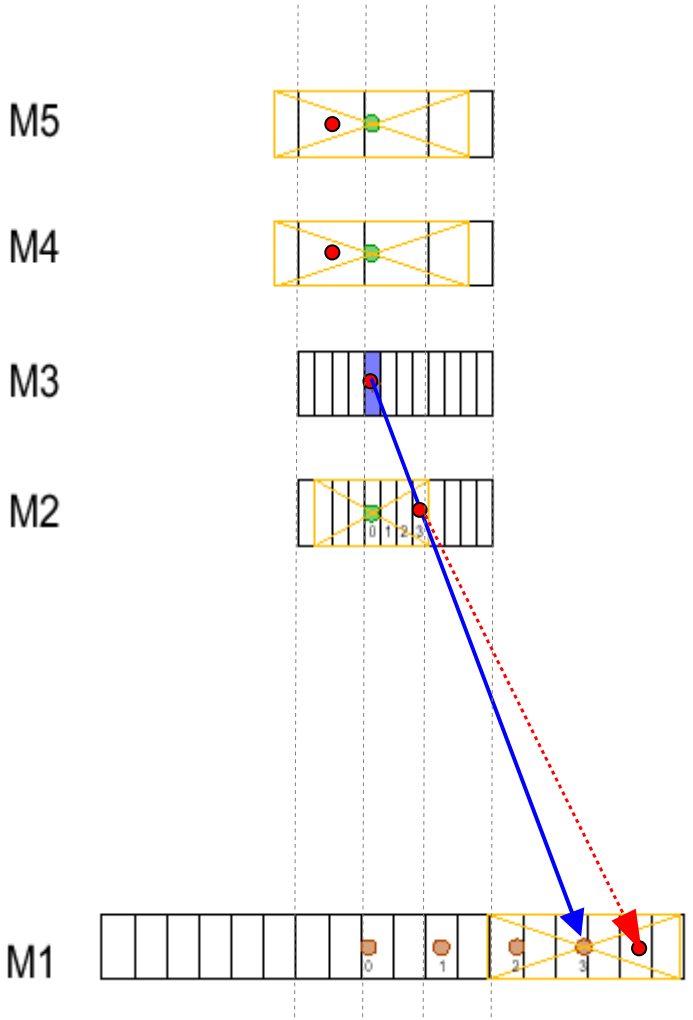


Stations M2 et M3:
Seulement des SRTIPS
Deux fois plus dense que M1



Stations M4 et M5:
Mélange de PADs et STRIPS
Deux fois moins dense que M1

Spécifications
Algorithme de sélection



Algorithme:

Recherche d'impactes dans M2, M4 et M5 dans une fenêtre centrée sur un impact dans M3

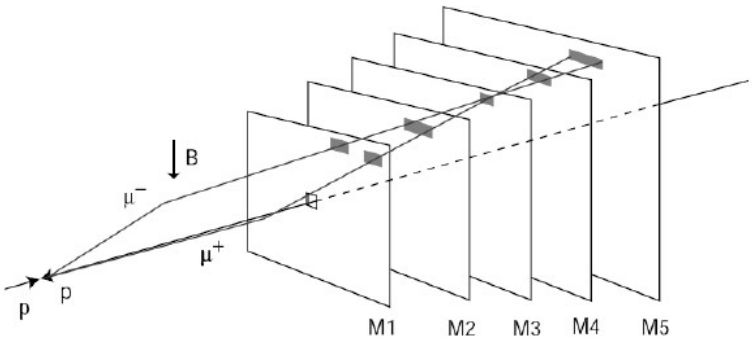
Recherche d'un impact dans M1 dans une fenêtre centrée sur une projection de M3 M2

L'impulsion transverse est calculée à partir

des coordonnées des impacts dans M2/M1

Structure projective

Algorithme ne nécéssitant que des opérations logiques.



Spécifications
Flux et performances

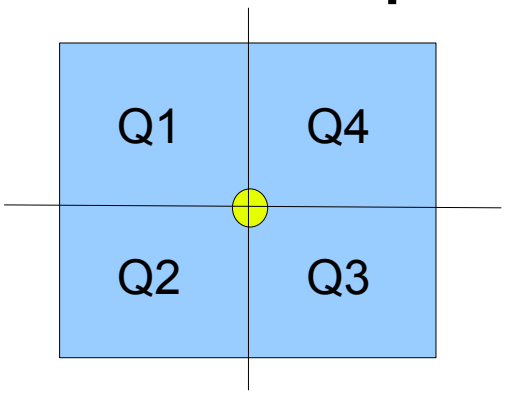
- Un flot de données de 25920 canaux lus toutes les 25ns

- Temps de sélection (exécution de l'algo sur toutes les données) **1,2 µs**

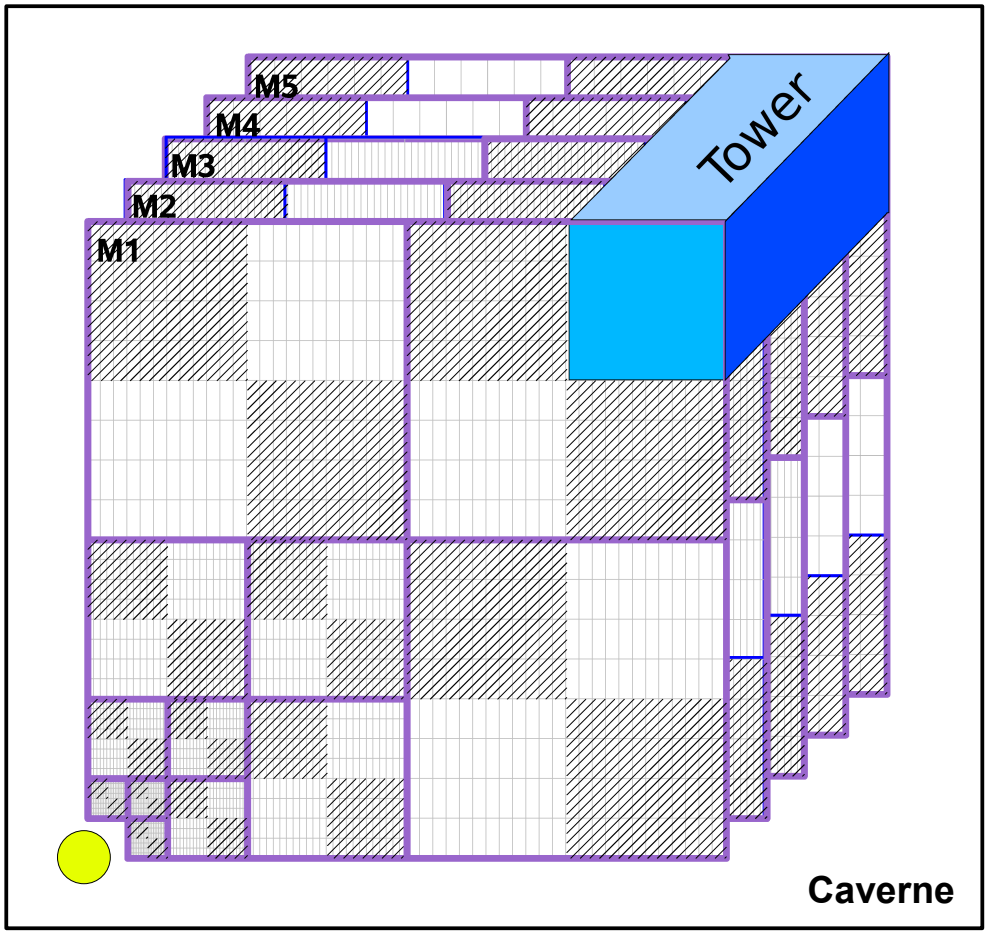
Conséquence:

Processeur Hardware avec architecture pipeline et massivement parallèle de type SIMD (Simple Instruction Multiple Data)

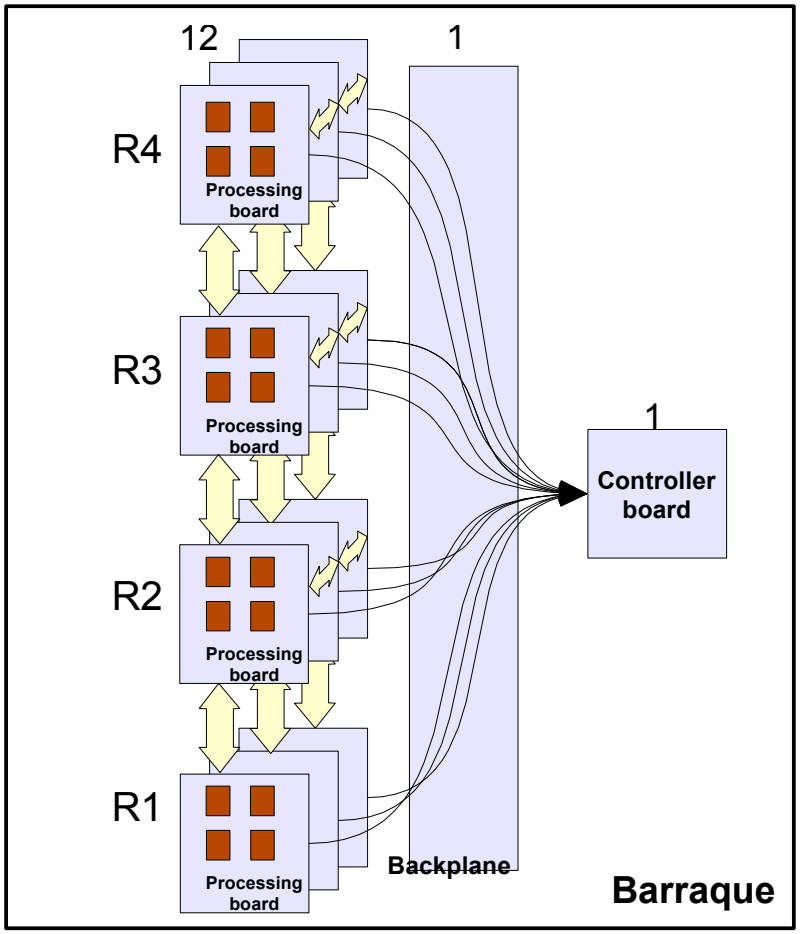
Découpage en 4 processeurs indépendants couvrant ¼ du détecteurs



Spécifications fonctionnelle de l'implantation



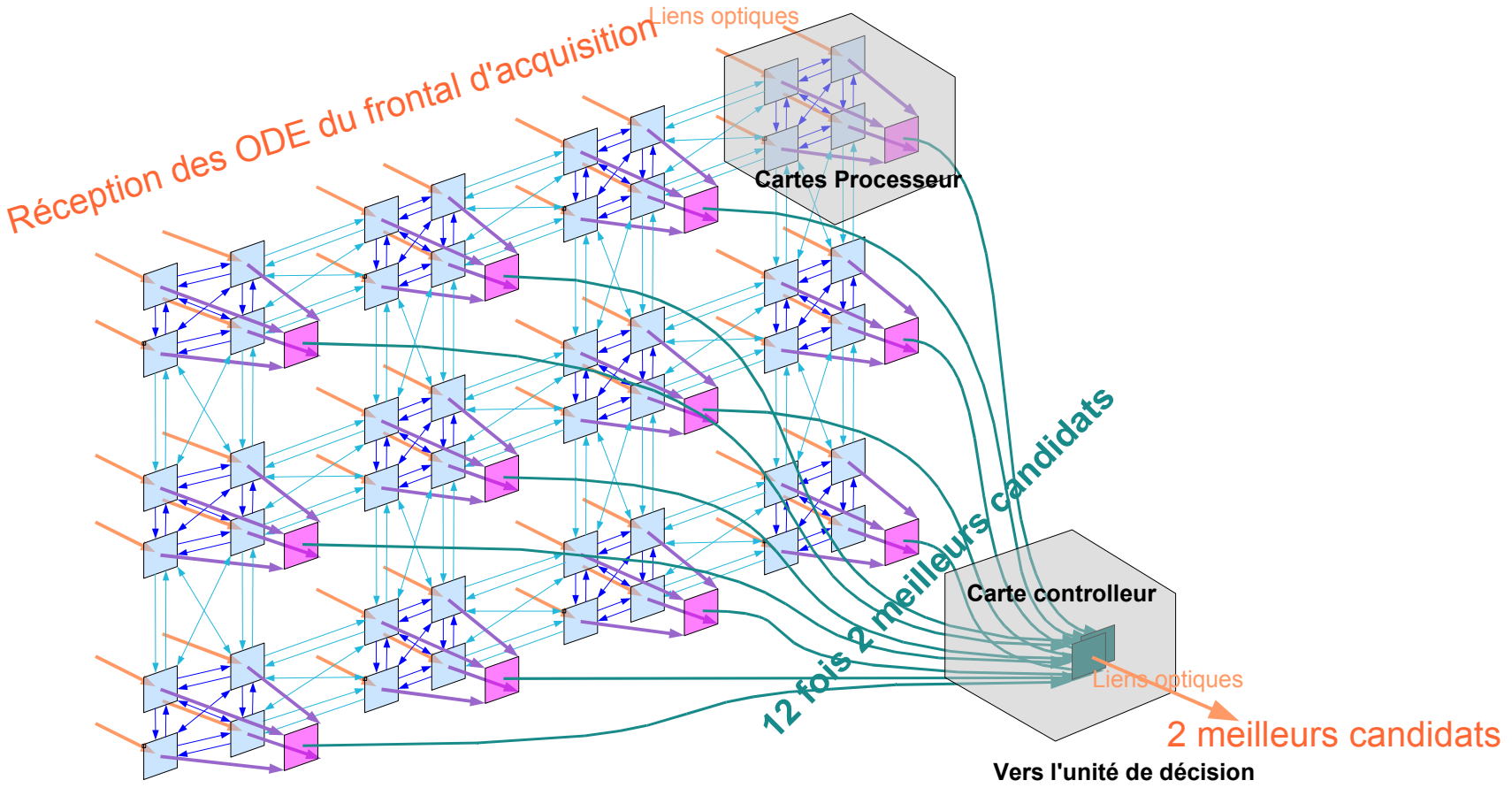
312
fibres
optiques



Architecture pour ¼ du détecteur: un processeur

Exécution de la sélection et échange de données

62 gros FPGA interconnectés pour ¼ du détecteur (248 gros FPGA pour tout le système)



Interconnection pour un quart de détecteur

Quelques chiffres clef

Liaisons série à haute vitesse 1.6 Gbits/s

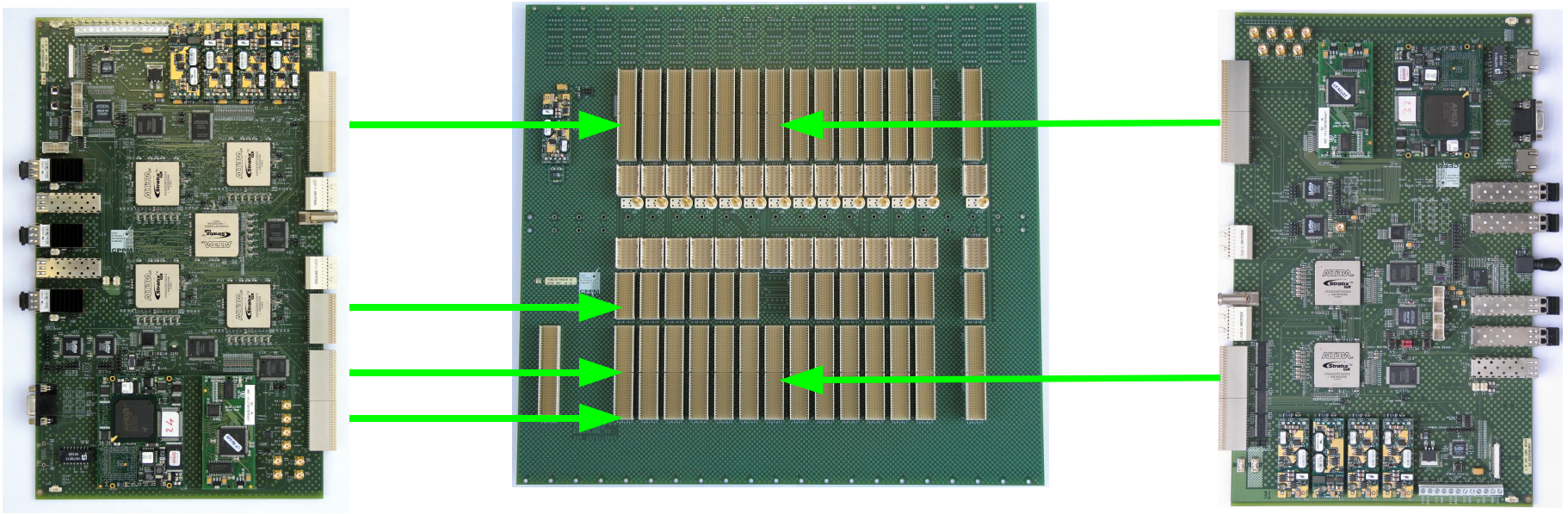
- A permis de diviser par un facteur 16 le nombres de fils à implanter

Communication des données sur fibres optiques

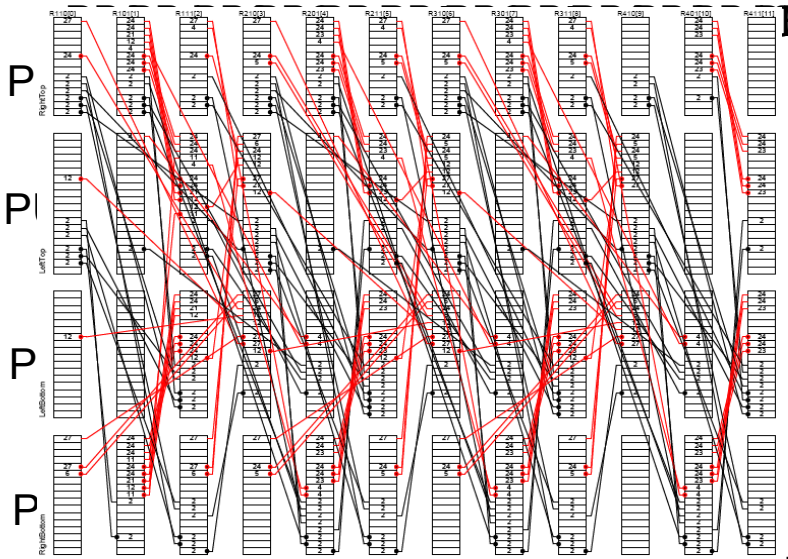
- 1248 fibres entrée venant des cartes ODE du frontal des chambres à muons @ 1.6Gbps
- 112 fibres en sortie vers les buffers du DAQ L1 et l'unité de décision L0 @1.6Gbps

•Les échanges entre les FPGA voisins du processeur sont fait sur des liaisons en cuivre à haute vitesse

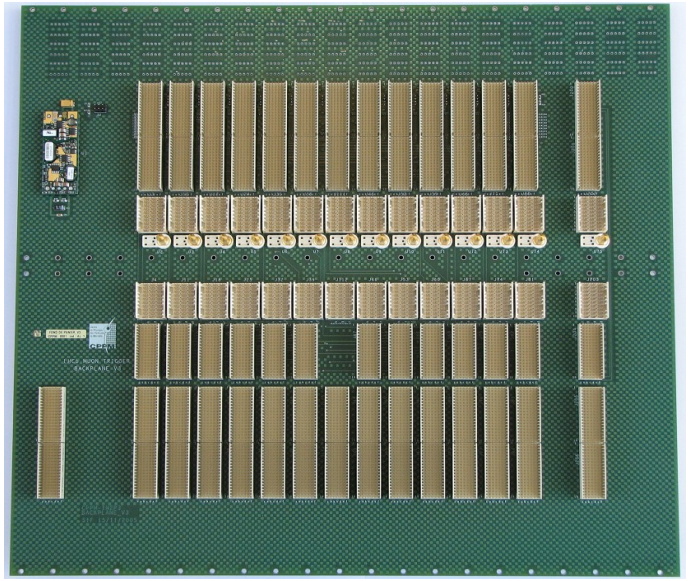
- 92 liaisons en cuivre par carte de calcul @ 1.6Gbps
- 110 liaisons en cuivres entre cartes via le fond de panier du processeur @ 1.6Gbps



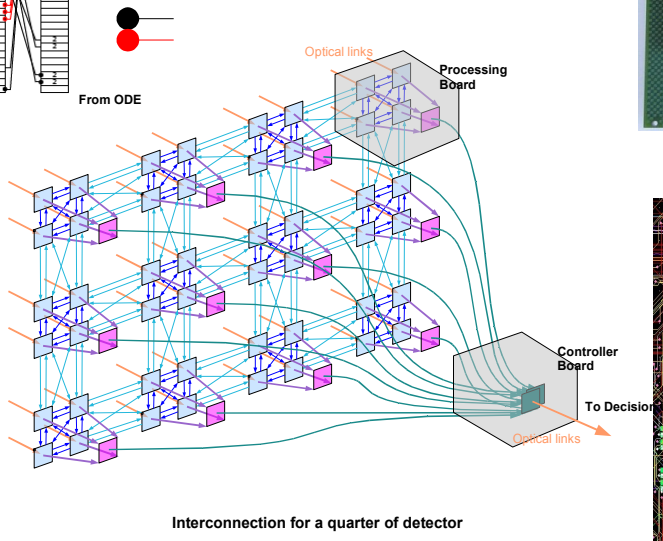
Thème majeur: maîtriser la complexité



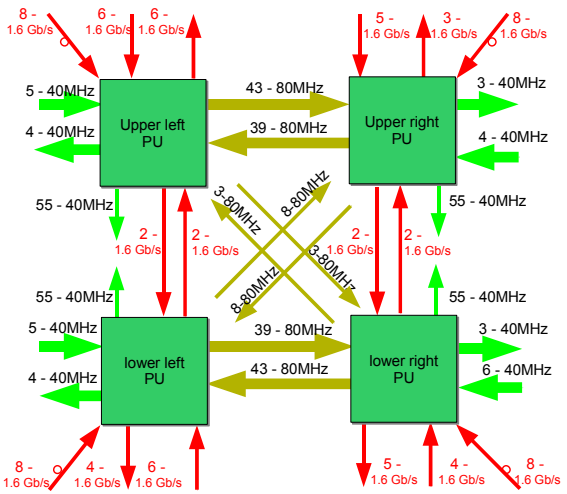
Fond de panier:
- 110 liens sériels
- 288 liens parallèles



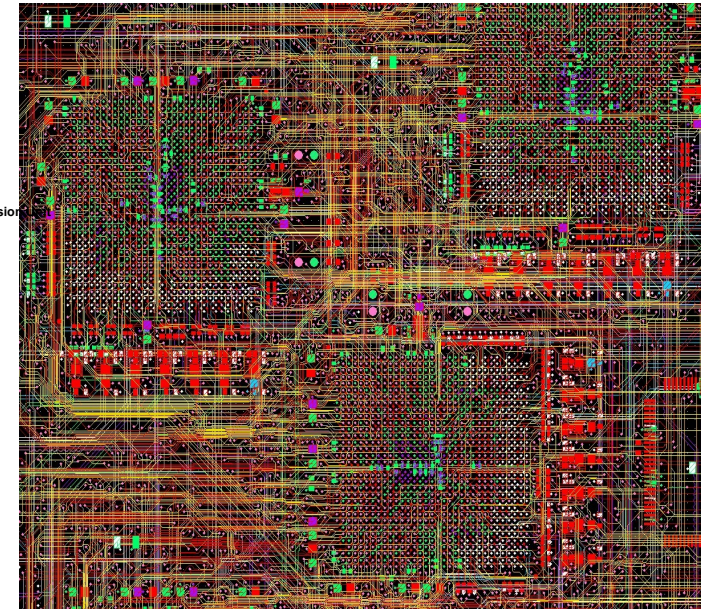
Liaisons séries entre composants:
- 92 intra-carte
- 110 inter-cartes



Interconnection for a quarter of detector



4608 algorithmes de recherche s'exécutant en parallèle pour 1 quart du trigger



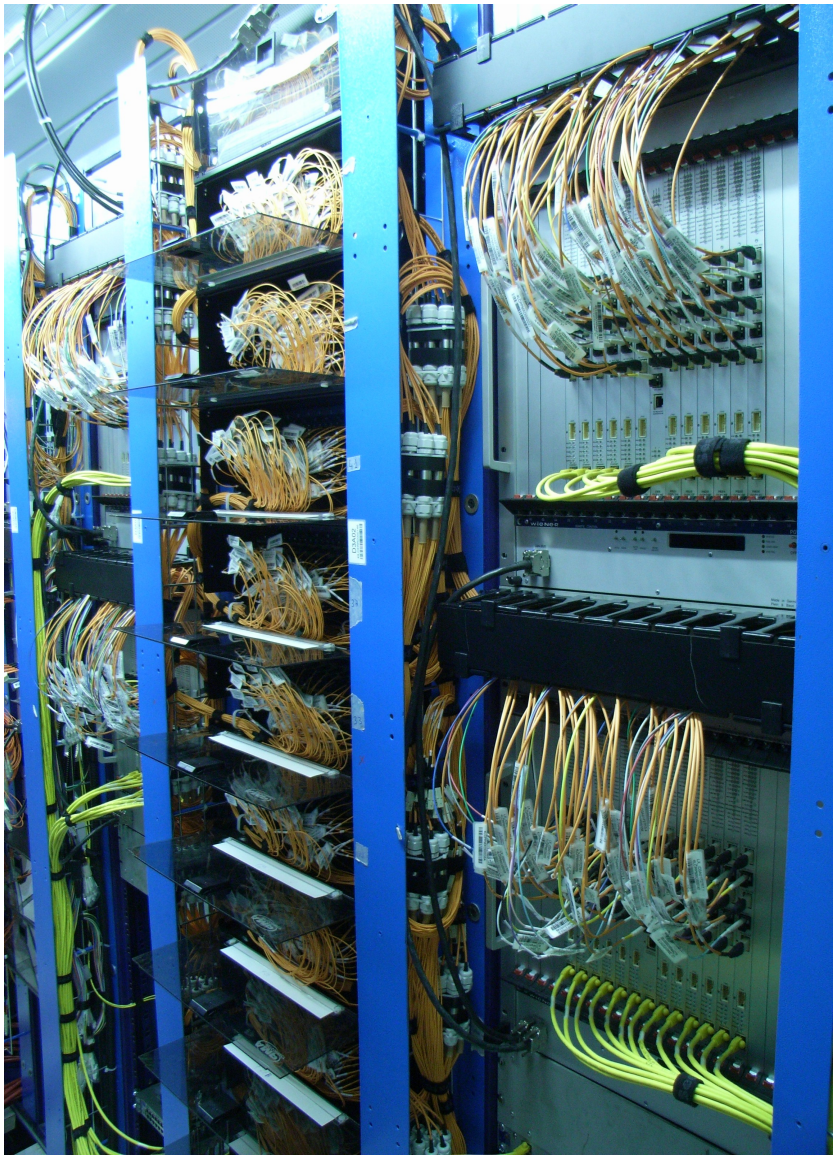
Thème majeur: maîtriser la complexité

La raison principale de la complexité d'un tel projet est les **grands nombres**:

- de capteurs sur le détecteur
- d'interconnexions entre les cartes électroniques du détecteur et le trigger
- d'interconnexions entre les composants matériels et logiques du trigger

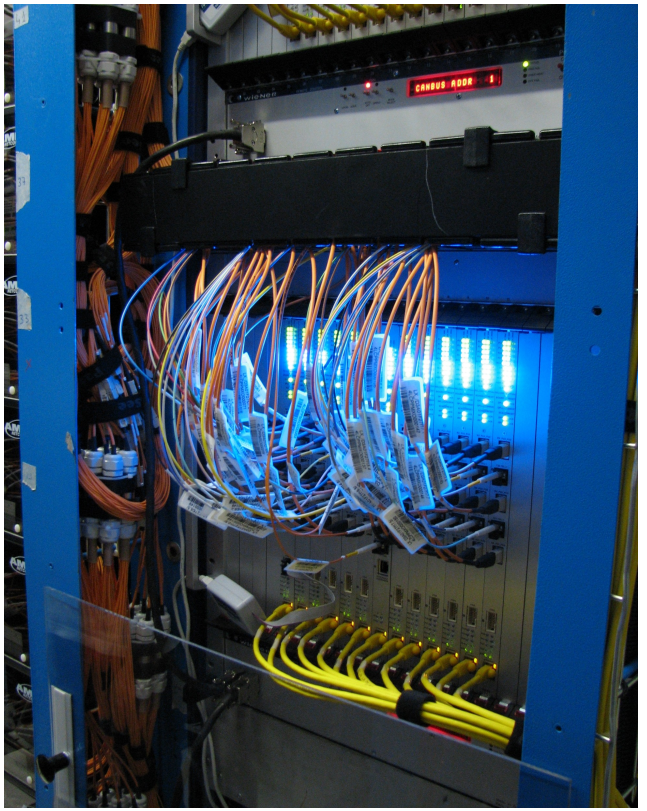
Bien qu'ayant été conçu sur des géométries et selon des découpages réguliers, de **nombreuses singularités locales** ont dues être introduites pour optimiser les performances , le prix et les conditions de réalisation du détecteur.

Comment je prouve que ça marche ????

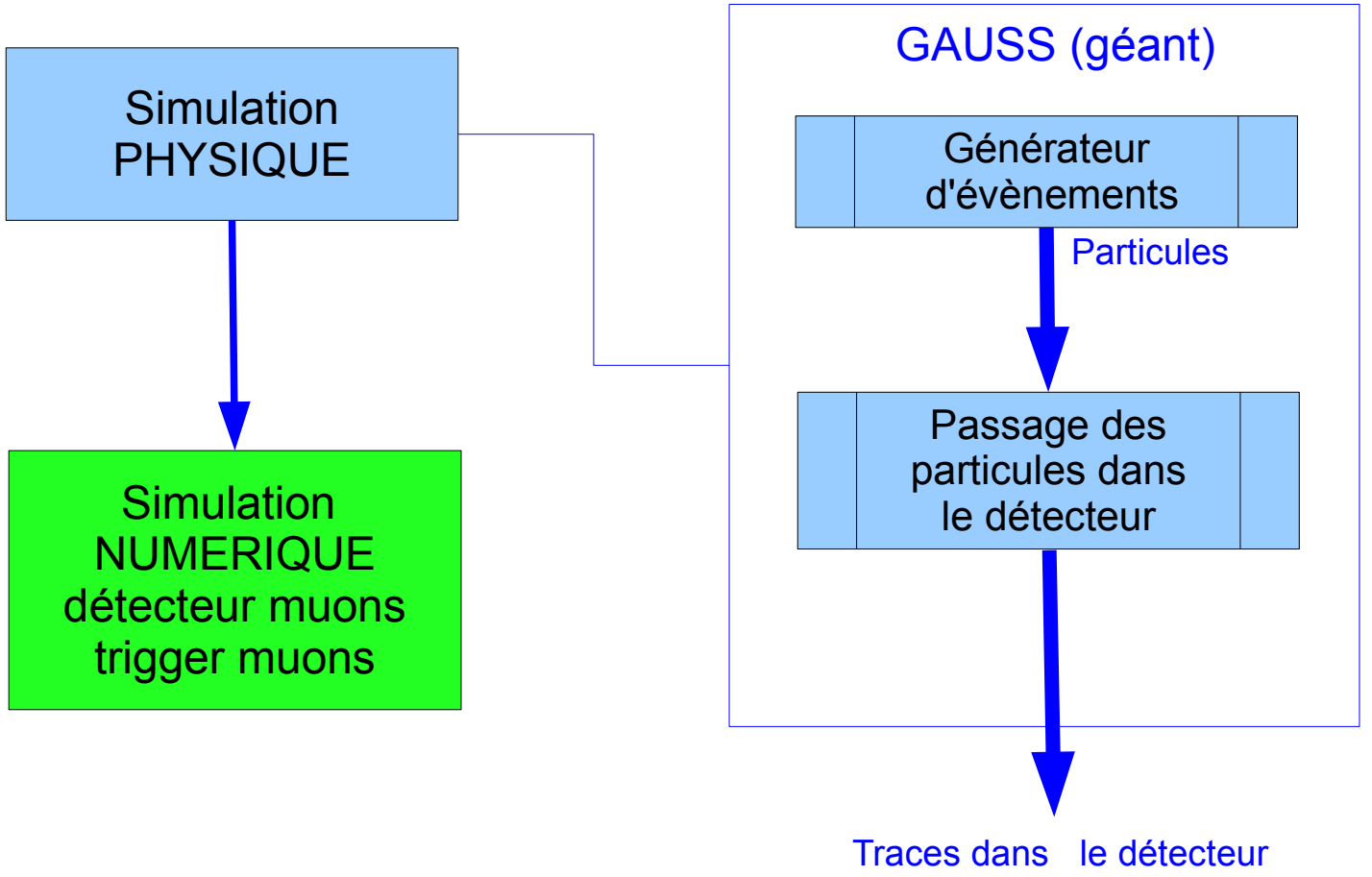


Faire face à la complexité:

l'aide de l'informatique



La chaîne de simulation



- Simulation collision PP
- Particules générées au point d'interaction

- interaction matière-particules dans tout le détecteur
- effet du champ magnétique

Simulation numérique du trigger

Simulateur du trigger

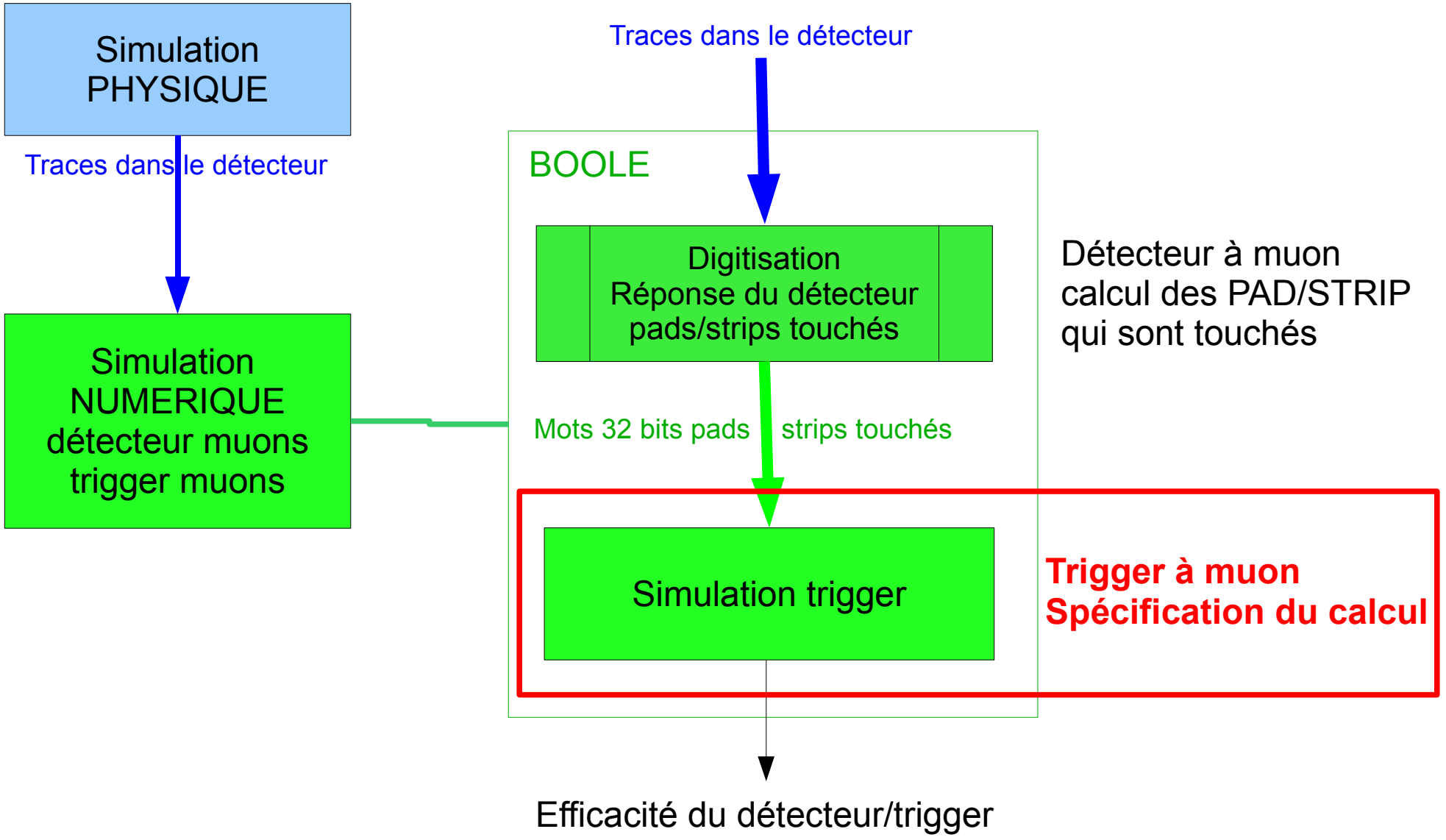
Le détecteur et son trigger ont été **conçus et optimisés** à l'aide d'un programme réalisant les fonctions de trigger.

Il a permis d'optimiser:
- la géométrie du détecteur
- le nombre de points de mesure
- l'effet de certains paramètres de l'algorithme comme la taille des fenêtre de recherche des impacts (hit) dans le détecteur (voir l'algorithme).

Le résultat:

Une **spécification de réalisation** du détecteur et du trigger

La chaîne de simulation: spécification du trigger



Implantation hardware du trigger

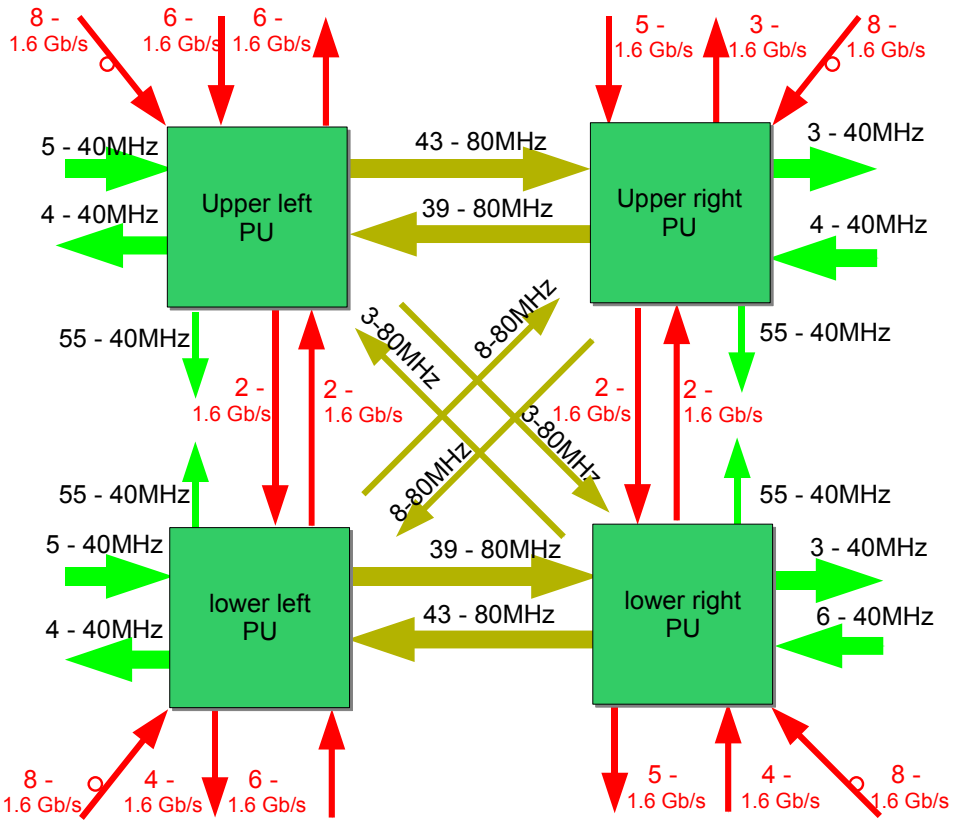
Les pécification on données:
- la segmentation du détecteur
- les informations de mesures à traiter
- le principe de l'algorithme de sélection des muons

Les électroniciens peuvent se mettre à travailler en prévoyant les moyens d'injection nécessaires de **patterns de test** dans le trigger hardware on pourra vérifier que sur les mêmes données le **simulateur et le processeur réel donnent les mêmes résultats.**

Problème:
Il y a un monde entre celui des programmes séquentiels écrits en C++ et celui des composants hardware traitant des bits. En cas de différence il n'existe **aucun moyen d'expliquer la source des différences entre les deux machines d'exécution**.

Solution:
Développer un **simulateur en C++ qui montre précisément les traitements des bits** de données du détecteur par les objets du hardware dans le calcul de sélection.

Implantation hardware du trigger



L'analyse du traitement du trigger montre que :

- la partie algorithmique de chaque boite est simple et facile à vérifier quand on connait les entrées et les sorties de chaque boite.

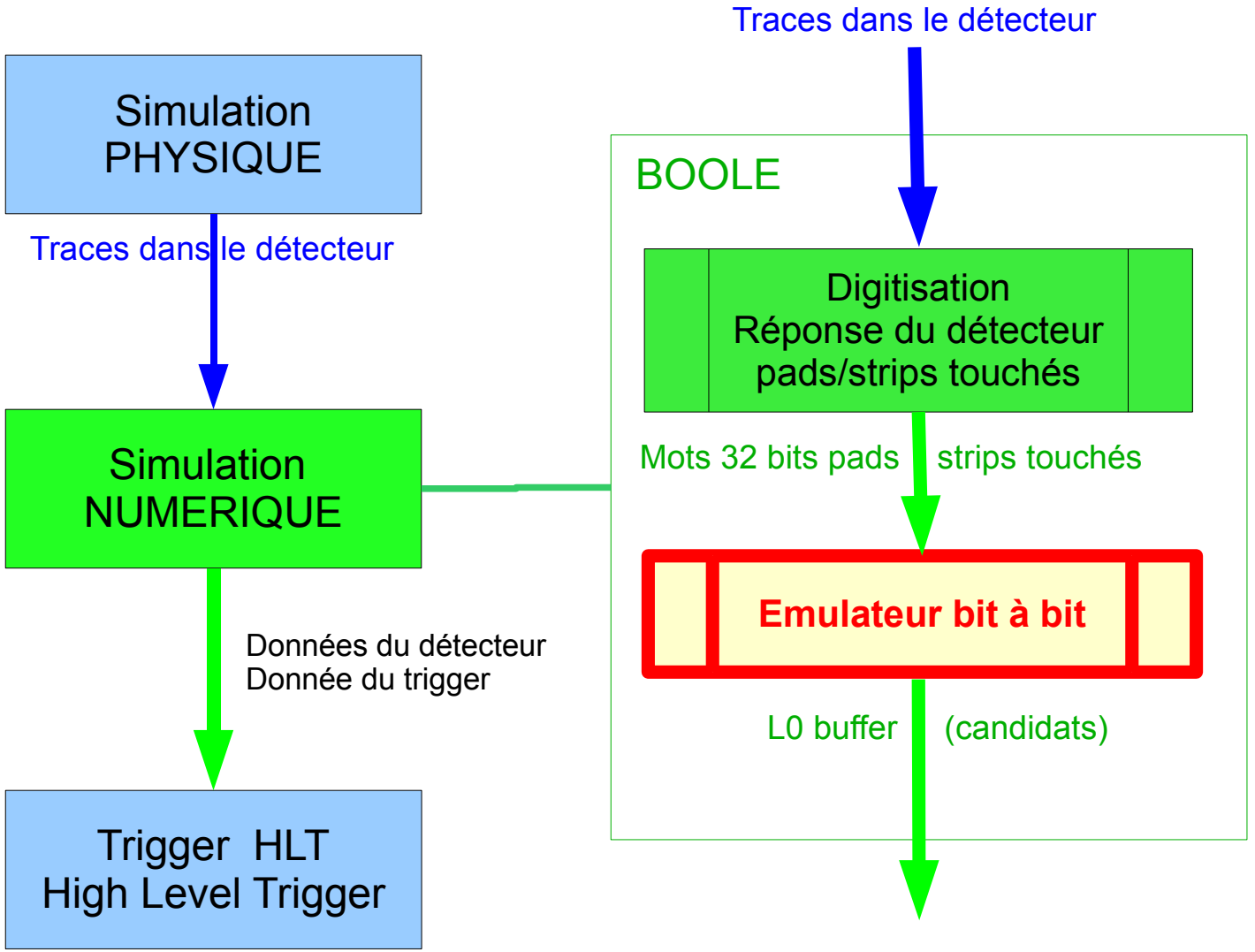
- les sources d'erreurs les plus probables sont liées aux cheminements des données:

  - interconnexions
  - brassages
  - sérialisation/désérialisation
  - pertes ou altérations électroniques

  entre les boites de calcul.

L'effort a donc été mis sur la modélisation de ces chemins dans le nouveau simulateur.

La nouvelle chaîne de simulation



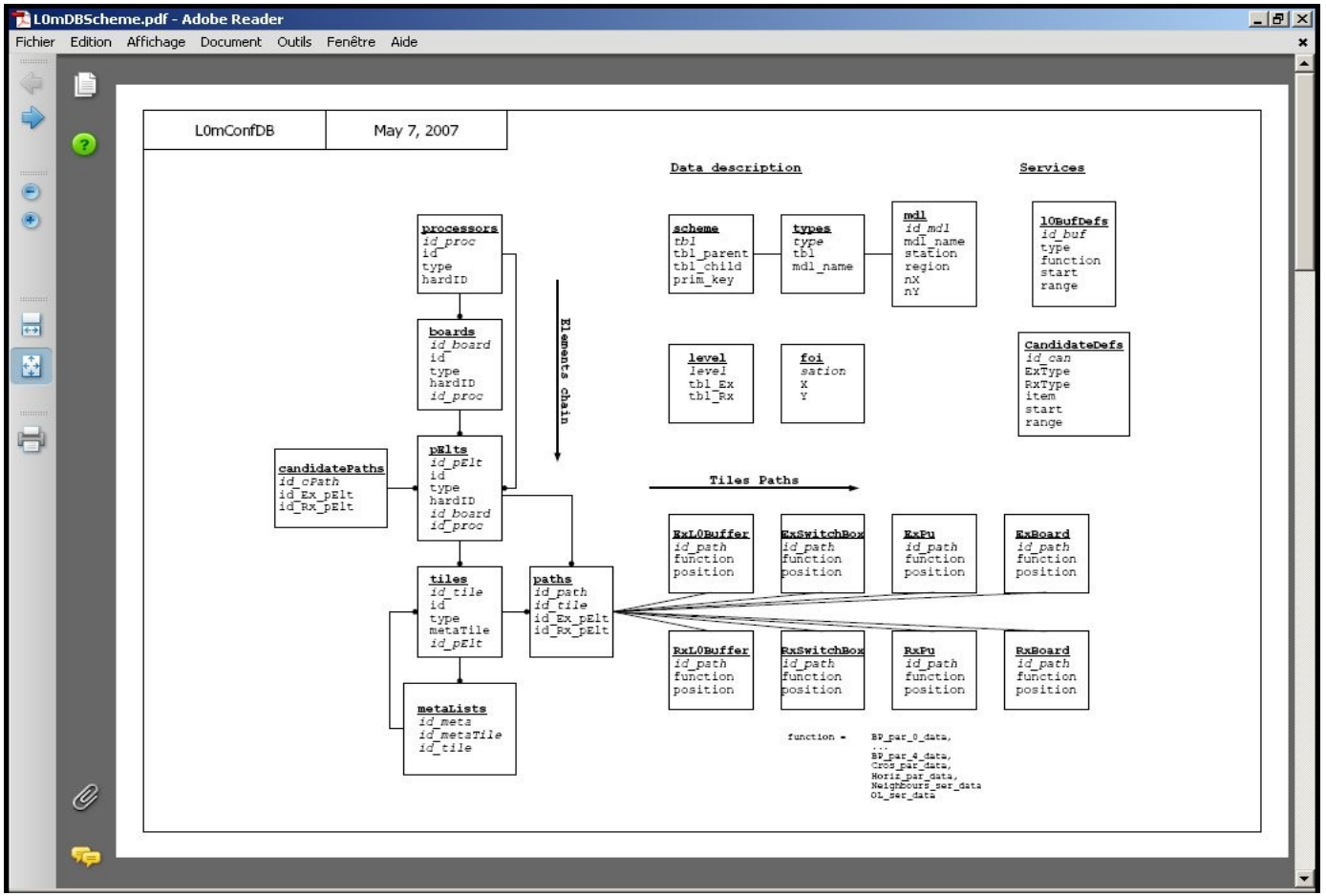
- calcul des PAD/STRIP qui réagissent
- réaction ADC

Modèle de référence du calcul numérique avec traçage des bits de pads et strips à travers le processeur

L'émulateur bit à bit : la DB de configuration

Le coeur du modèle du processeur est une base de données sqlite qui contient :

- l'identification de tous les bits d'informations issus du détecteur et leur position sur le détecteur
- la liste de tous les composants logiques hardware traitant ces bits
- les chemins de circulation de tous ces bits entre les composants hardware

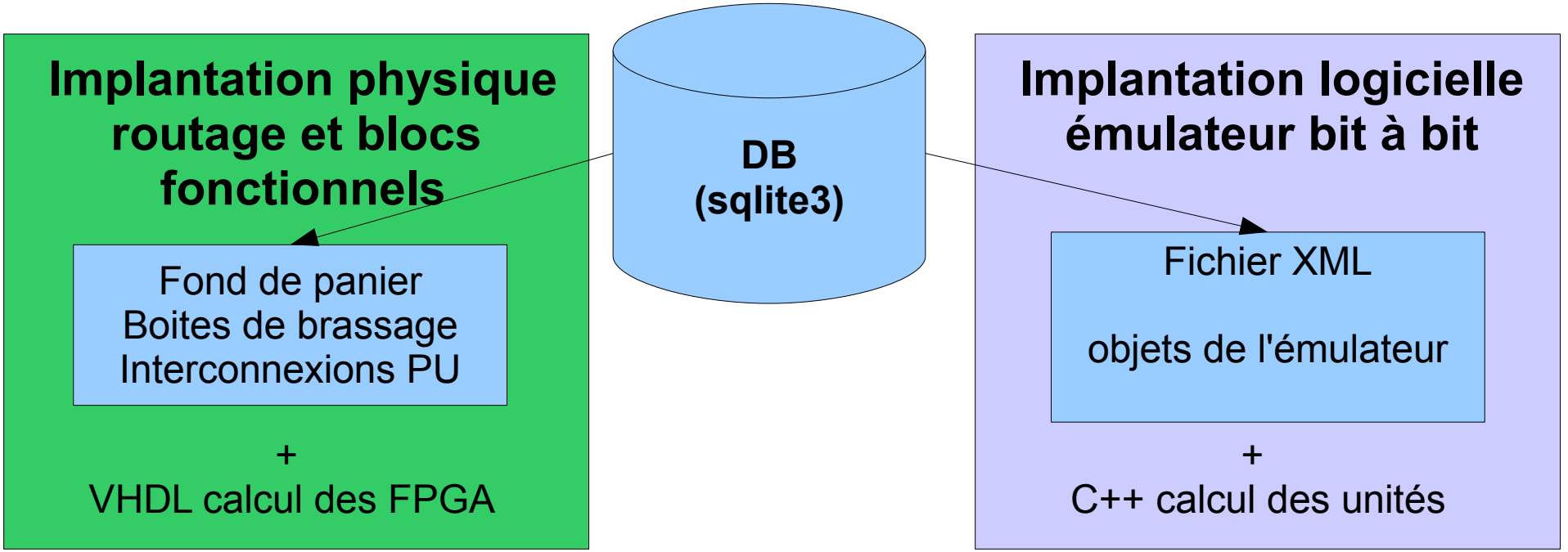


L'émulateur bit à bit : la DB de configuration

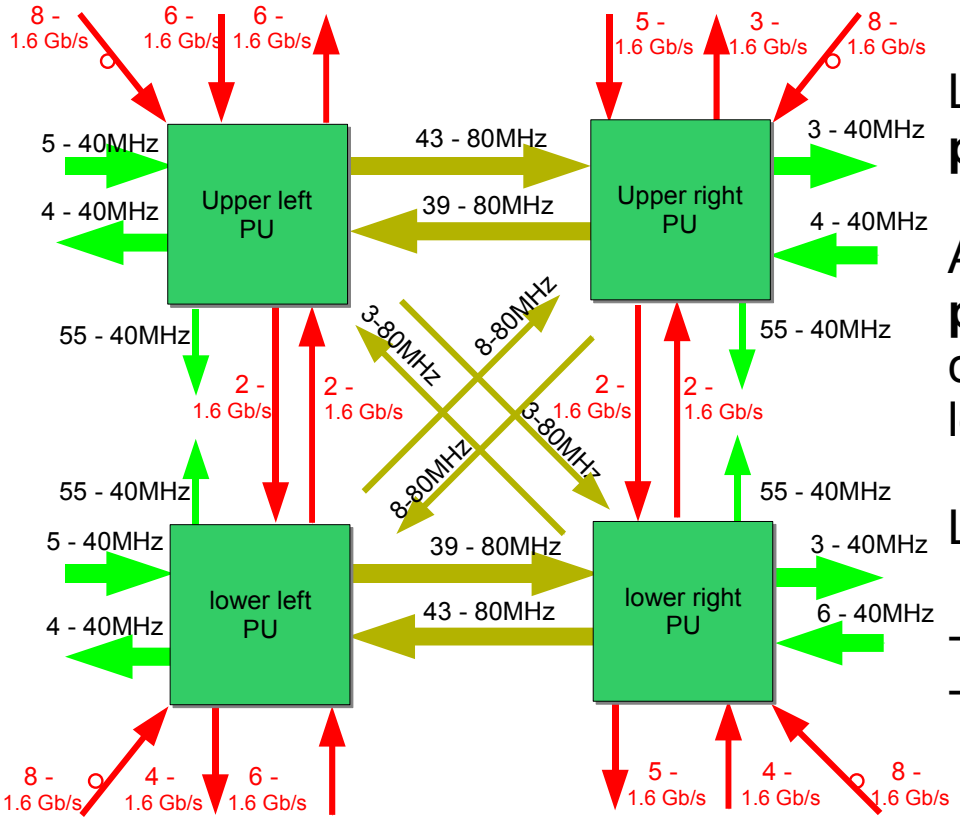
Elle permet d'extraire automatiquement la liste des objets et de leurs interconnexions/relations pour obtenir:

- les objets pour le modèle software exécutable: l'émulateur bit à bit en C++

- les éléments d'inter-connexions du hardware:
  - le câblage du fond de panier
  - les boites de brassage et d'interconnexions entre les Processing Units



L'émulateur bit à bit: les interconnexions



L'identifiant de chaque bit permet de **localiser sa position géographique sur le détecteur**.

Avec les informations de la DB on peut **tracer la position de ce bit** aux différentes étapes du calcul effectué dans les composants hardware et le buffer L0.

Le simulateur bit à bit permet:

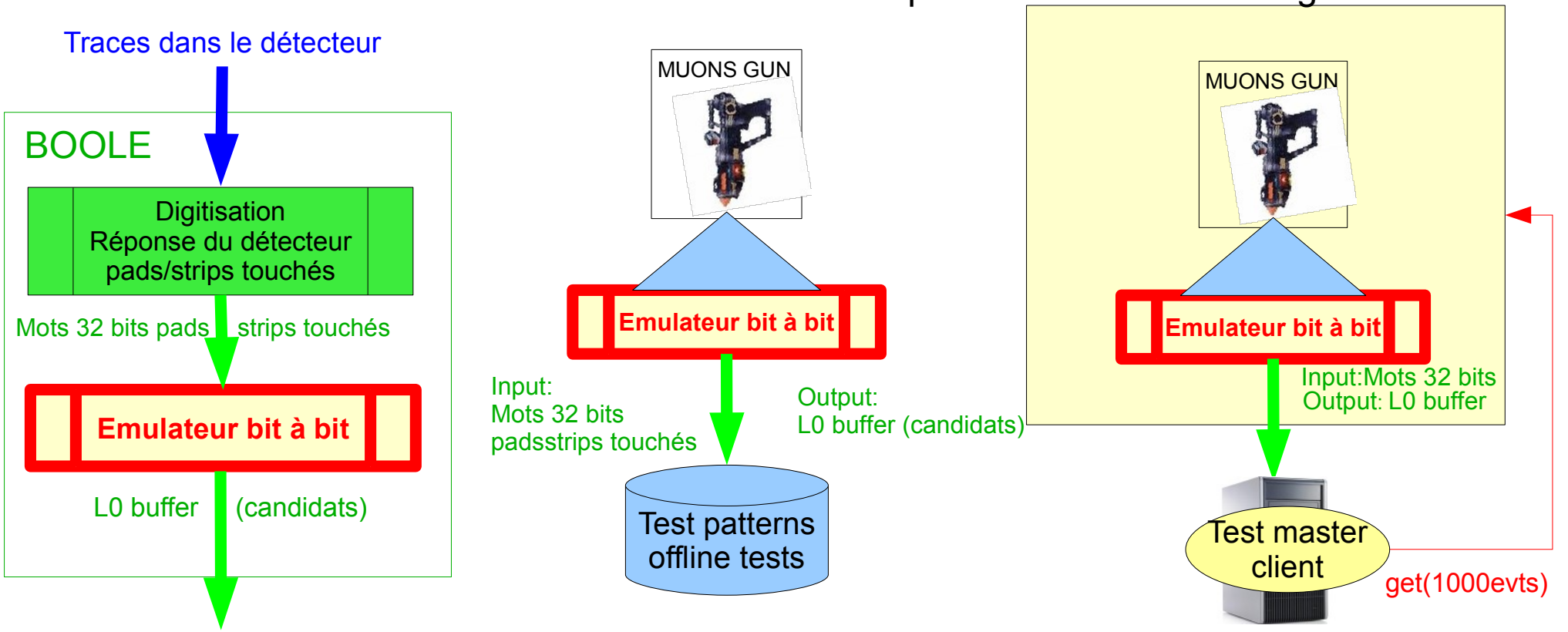
- vérifier le résultat final de la sélelection
- vérifier le bon cheminement des données

L'émulateur bit à bit: modes d'utilisation

1 - Couplé la **chaîne de simulation** standard sur les évènements générés

2 - Wrappé dans un module PYTHON il permet d'être couplé à un « **muons gun** » arrosant systématiquement tout le détecteur

2.1 - Pour produire hors ligne une **banque (DB) de patterns de test**
2.2 - Pour être **serveur d'évènements de test** pour une validation en ligne



Le « MILLION » d'évènements

Objectif:

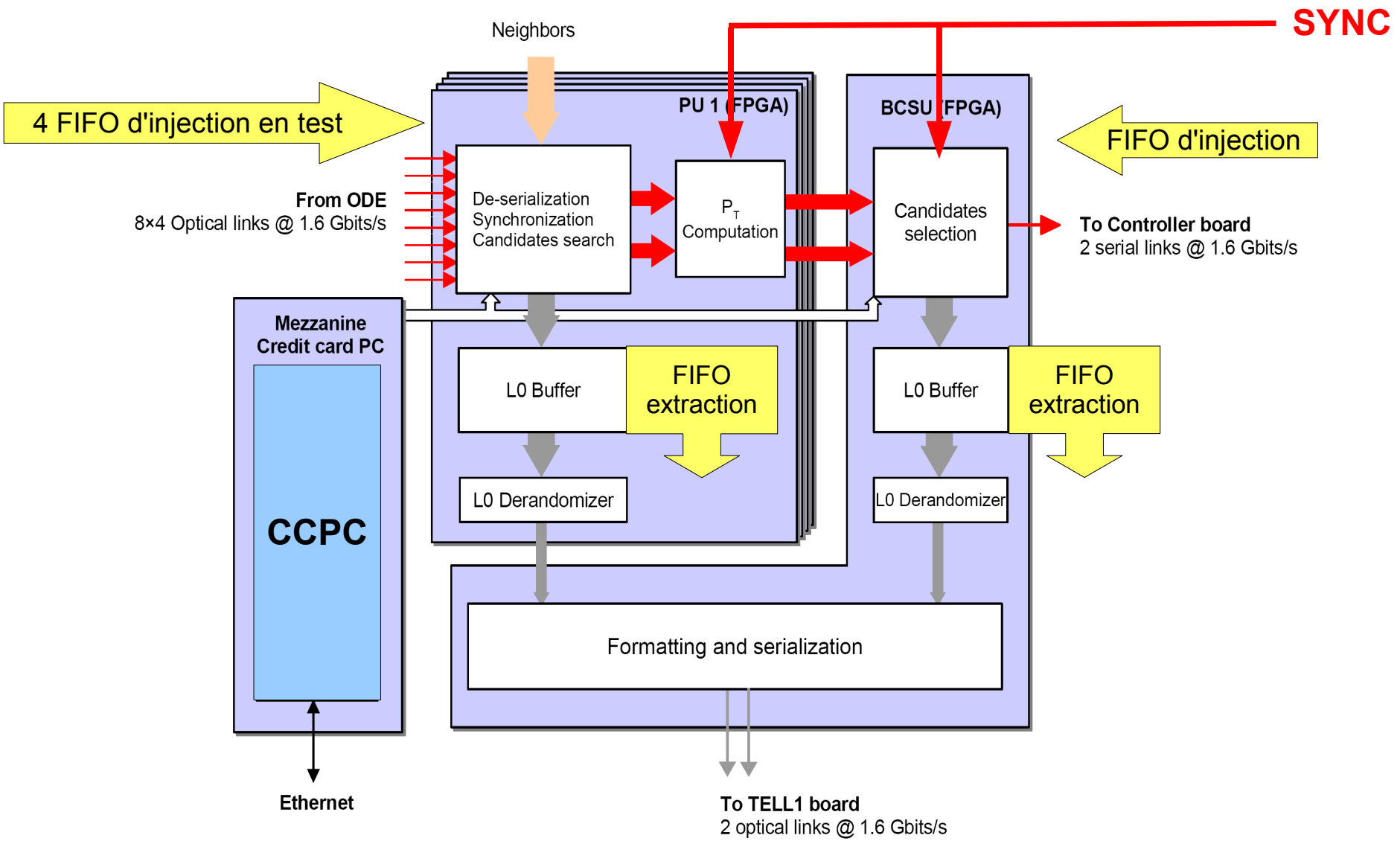
- Pouvoir **valider un processeur** hardware et firmware intégrés
- **Couverture du maximum** de cas plutôt qu'évènements réalistes
  MUON GUN pour avoir 1 à 2 muons par processing unit plutôt que
  1 à 2 par événement en simulation réaliste

Résultat:

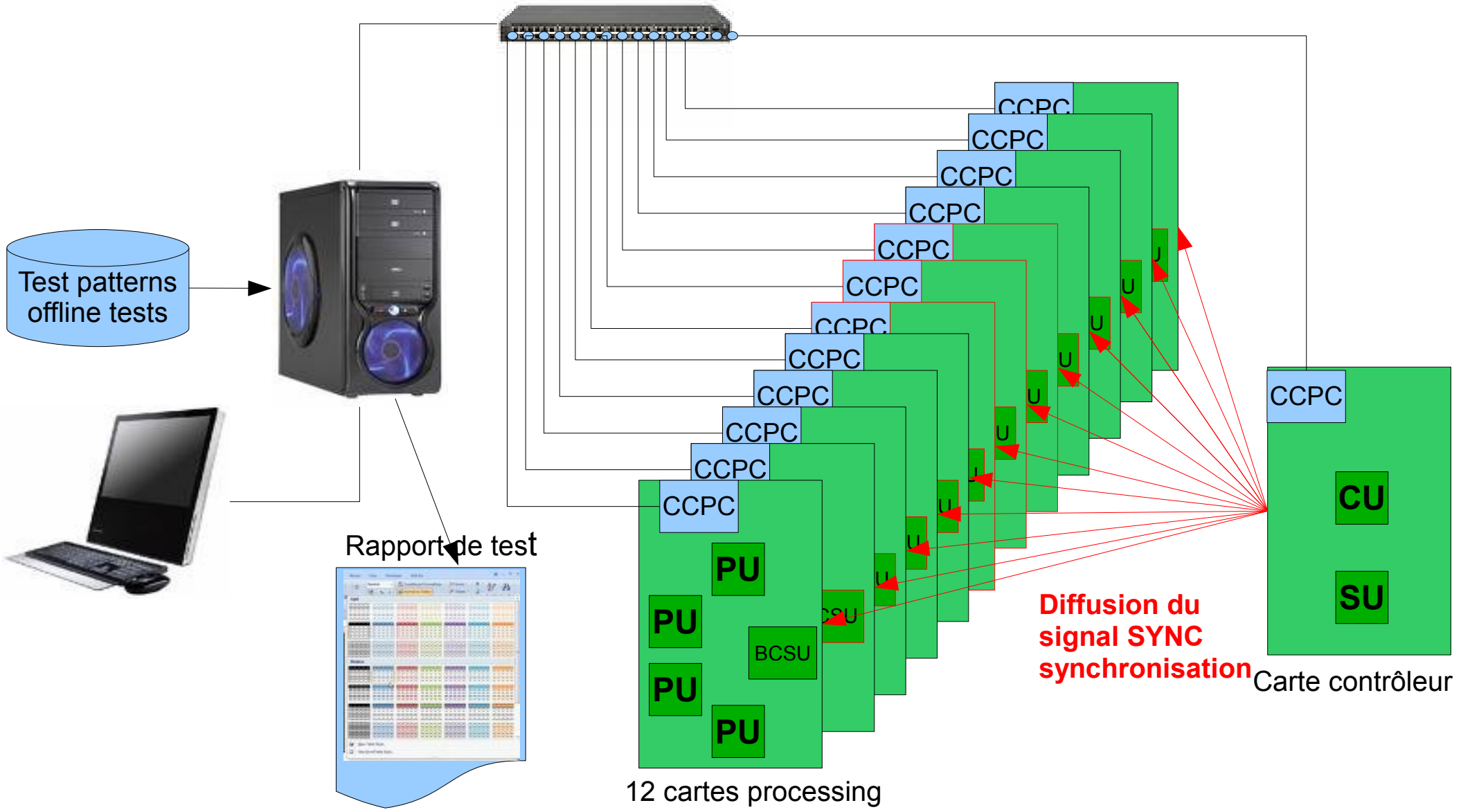
- Une **banque de 1 million d'évènement de référence** générés en mode programme autonome. Cette base peut-être transportée pour servir à valider le processeur au laboratoire ou au CERN à tout moment.

- Utilisable par **le système de test** sans plus avoir besoin de l'émulateur en ligne.

Validation
Testabilité d'une carte de calcul



Validation
Le système distribué de test



Validation
Le système distribué de test

Quelques chiffres:

- un millions d'évènements c'est **2,5 millisecondes** de fonctionnement du trigger

- Il faut **6 heures** à l'outil de test pour passer le million d'évènements:

    - la version en C 8 fois plus rapide que la version tout PYTHON initiale
    - le programme client maître (PC) est en PYTHON et les serveurs (CCPC) en C

- Il faut **10 heures** pour générer le million d'évènements avec le simulateur

Conclusions

Les outils informatiques nous ont permis de maîtriser le développement et la validation d'un **système électronique extrêmement complexe**:

- grand nombre d'interconnexions
- grand nombre d'objets de calcul
- algorithme explosé en tâches parallèles interdépendantes
- beaucoup de singularités locales en dépit d'un design générique initial homogène

Avec **une base de données décrivant tous les objets et interconnexion du système** on a pu générer semi-automatiquement :

- l'implantation logicielle du processeur sous la forme d'un émulateur bit à bit qui a servit de **spécification comportementale de référence**

- l'implantation hardware sous la forme d'un chassis contenant 62 FPGA dans 12 cartes de calcul et 1 carte de contrôle.

On a développé les outils informatiques de test qui permettent à tout moment de vérifier que **le processeur hardware est l'image fidèle de son modèle soft de référence**.