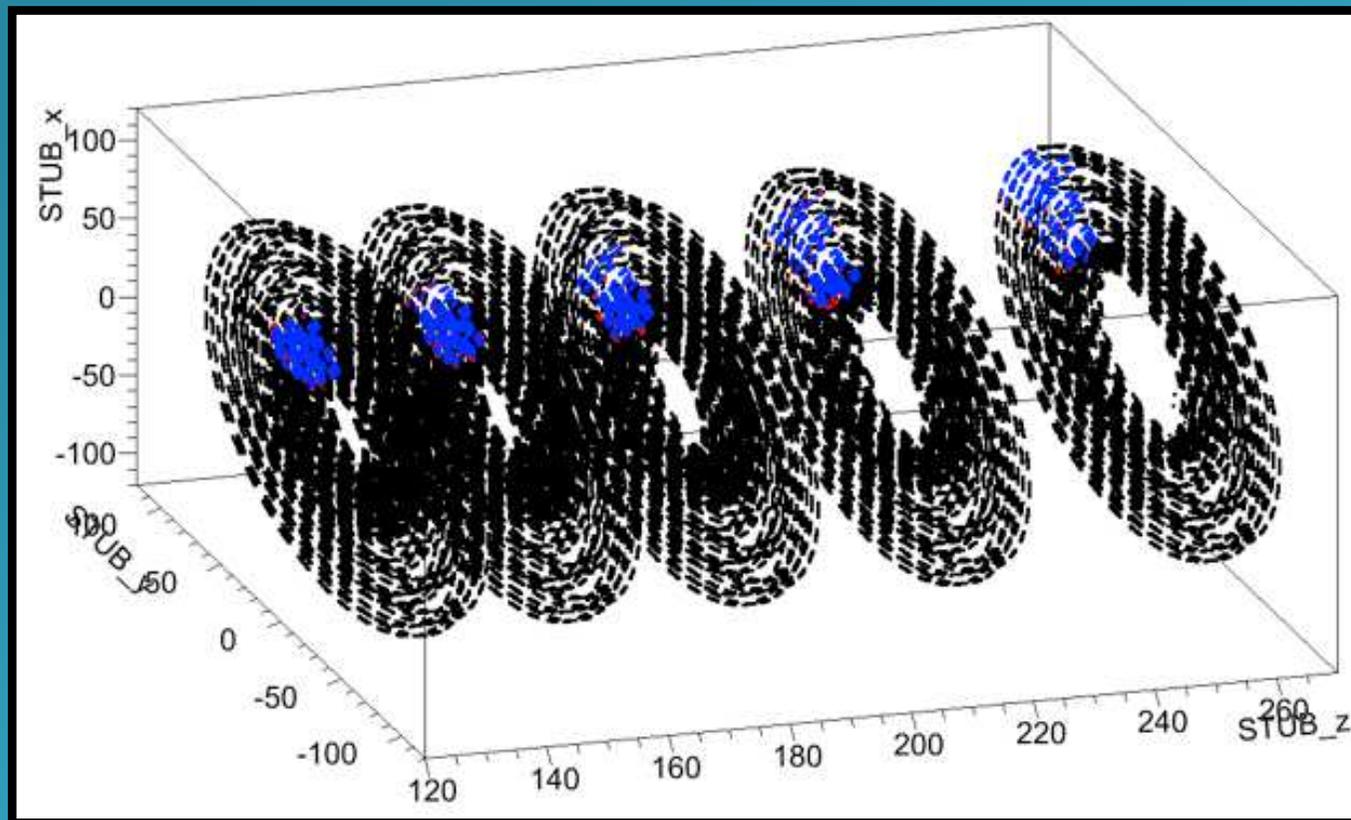


# Clustering et Filtrage rapide de traces pour le L1 track trigger de CMS



## Le projet HL LHC et l'upgrade Phase II de l'expérience CMS



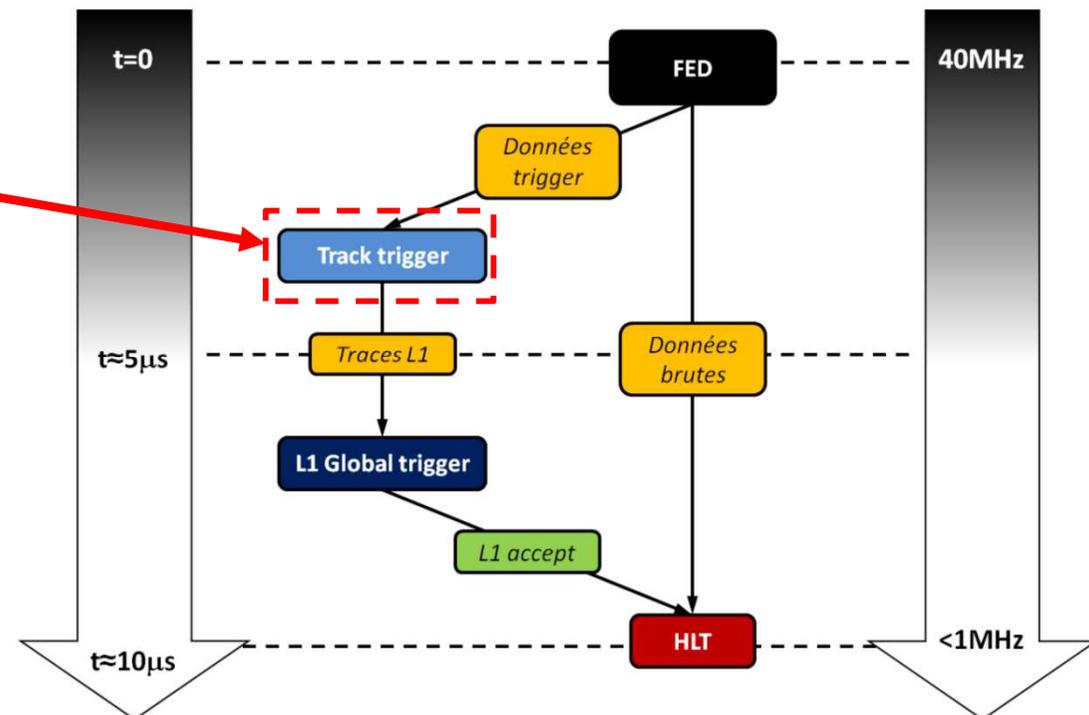
**High  
Luminosity  
LHC**

- Grande augmentation de la luminosité
- Très grande activité dans les calorimètres et les chambres à muons
- Le L1 trigger actuel aurait donc une mauvaise sélectivité

Le L1 trigger devra utiliser les informations du tracker !

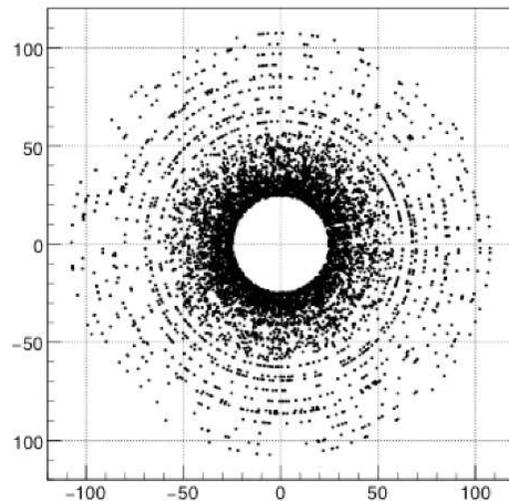
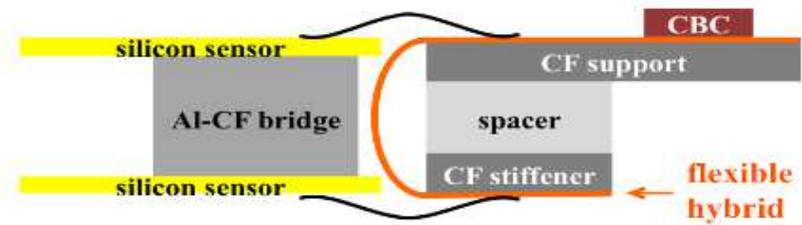
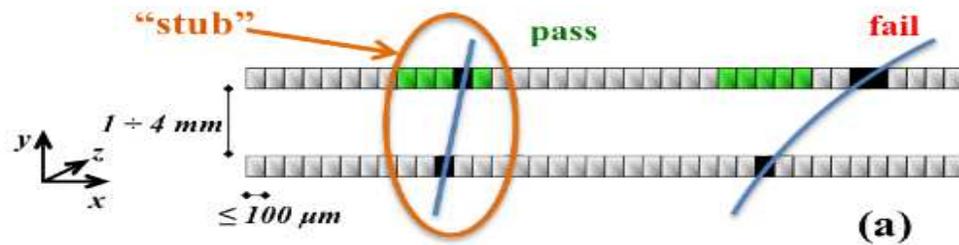
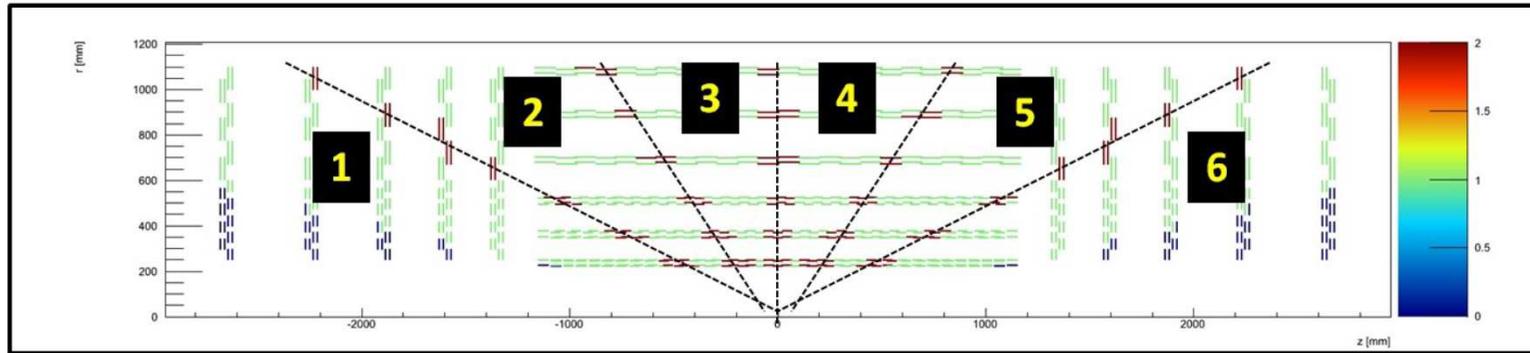
Projet FastTrack

Reconstruction rapide de trace par l'utilisation de mémoires associatives

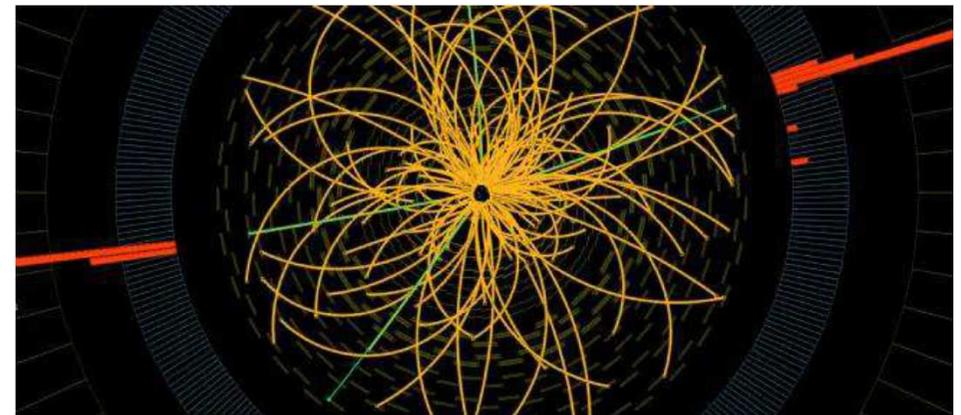


# Contexte

## Le tracker de l'upgrade Phase II CMS

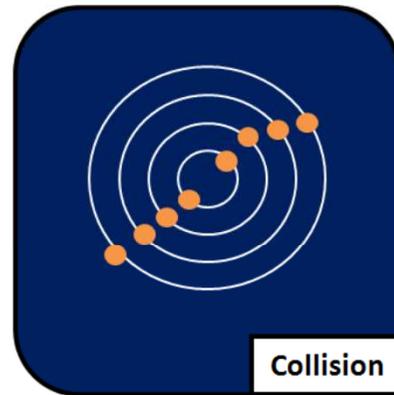


FastTrack

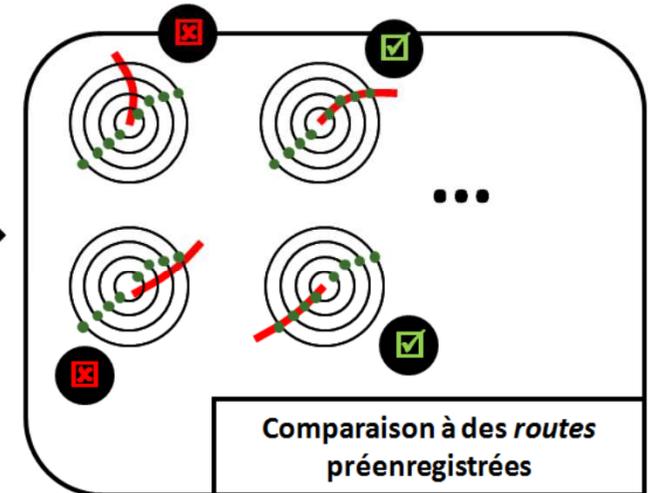


# La solution Mémoires Associatives

## Principe de la pattern recognition 3D

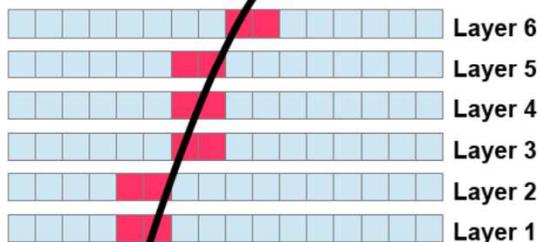


1 Envoi des signaux dans la mémoire associative

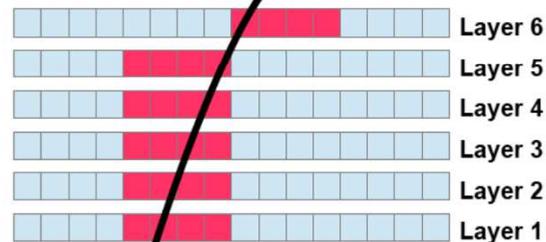


2 Envoi des routes compatibles

On peut jouer sur de nombreux paramètres afin de diminuer la taille des banques de patterns tout en gardant une bonne sélectivité  
Exemple : La taille des SuperStrips (coordonnées dégradées)



Ex : supertrip size = 2 strips

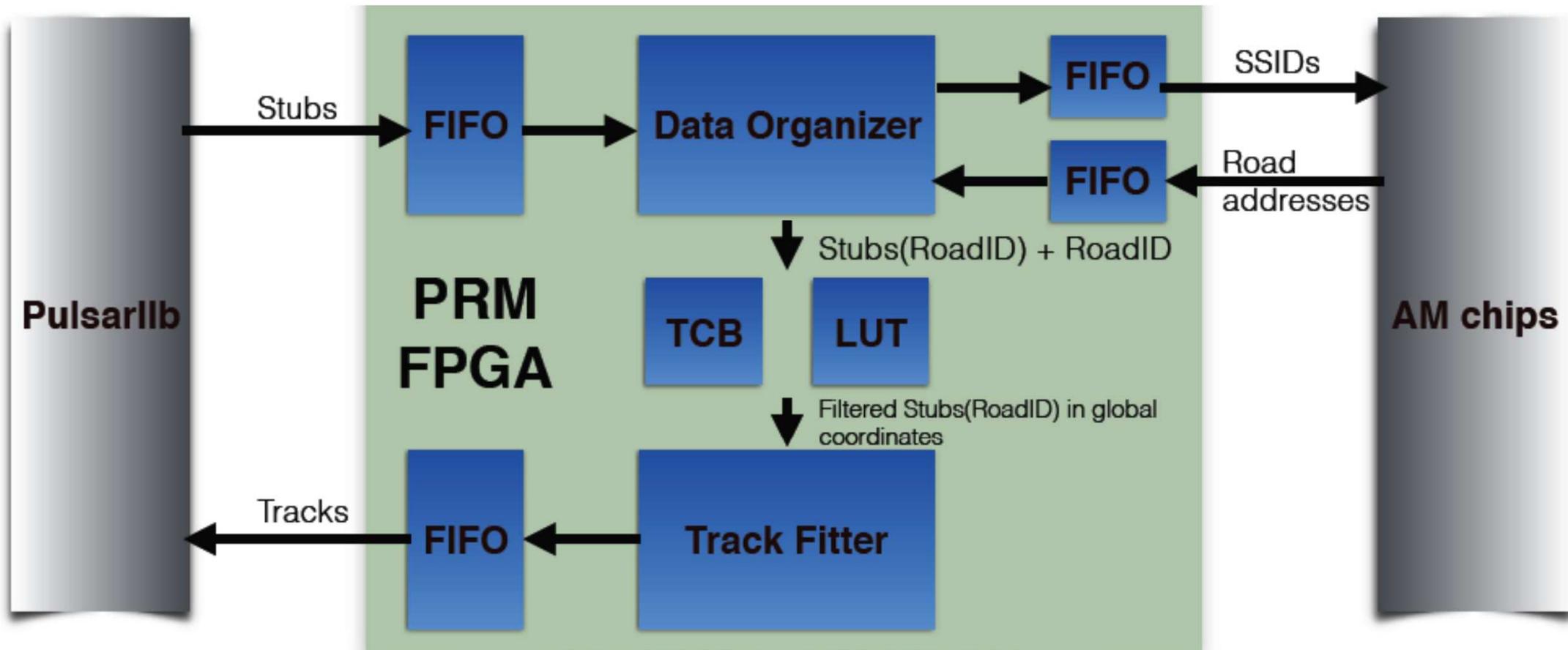


Ex : supertrip size = 4 strips



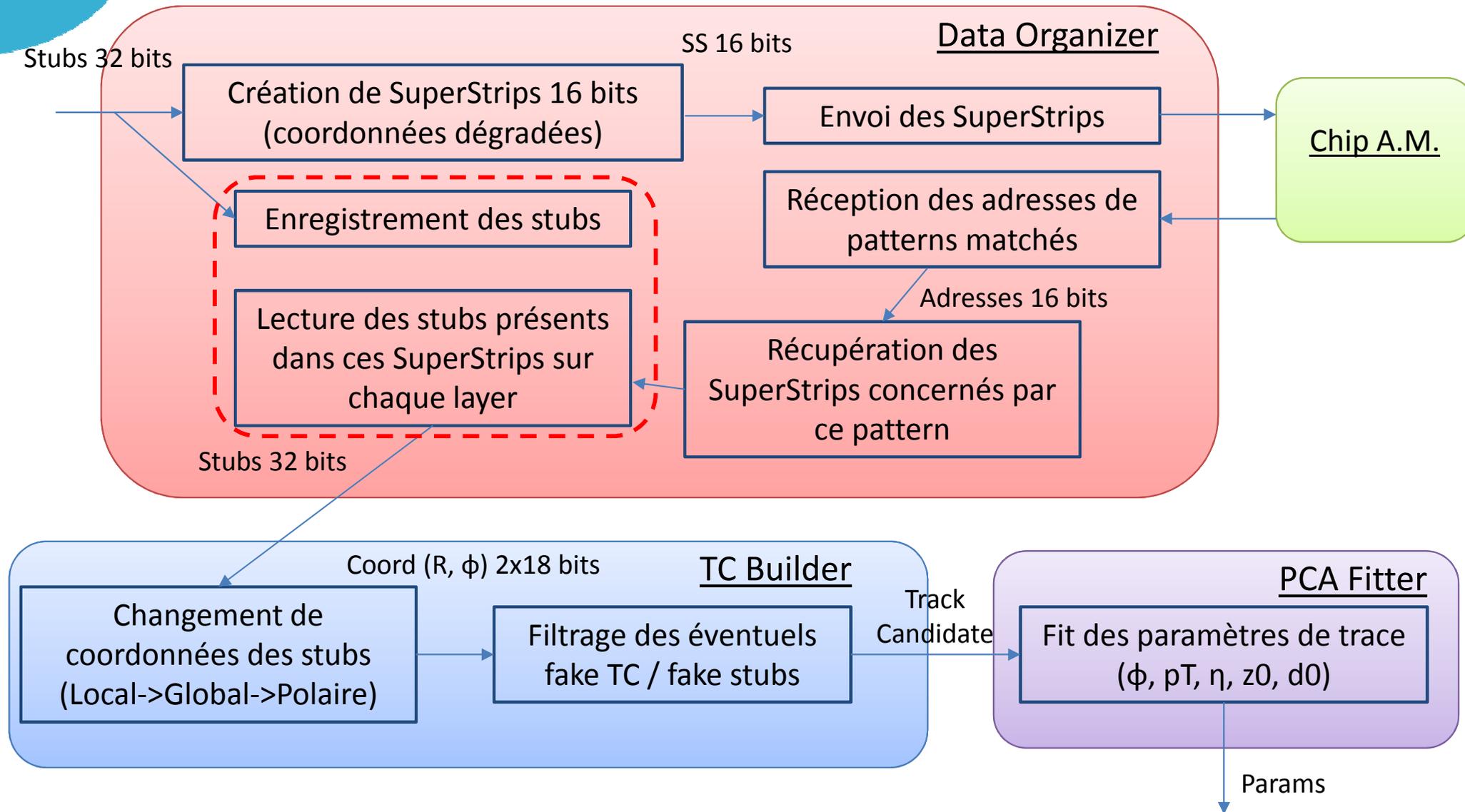
# La solution Mémoires Associatives

Le Firmware dédié à la mise en œuvre des mémoires associatives



# La solution Mémoires Associatives

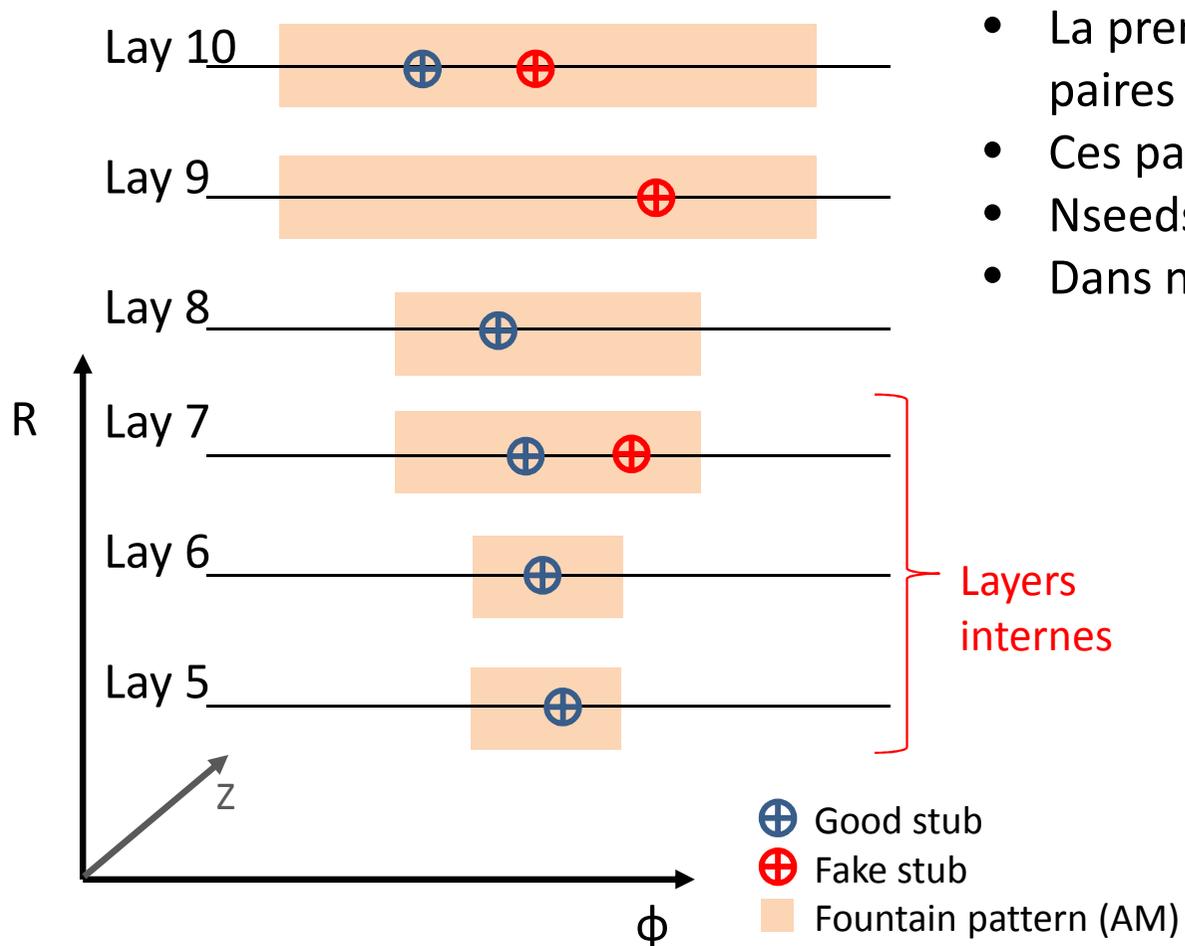
## Le Firmware dédié à la mise en œuvre des mémoires associatives



# Track Candidate Builder

## Présentation de l'algorithme de filtrage

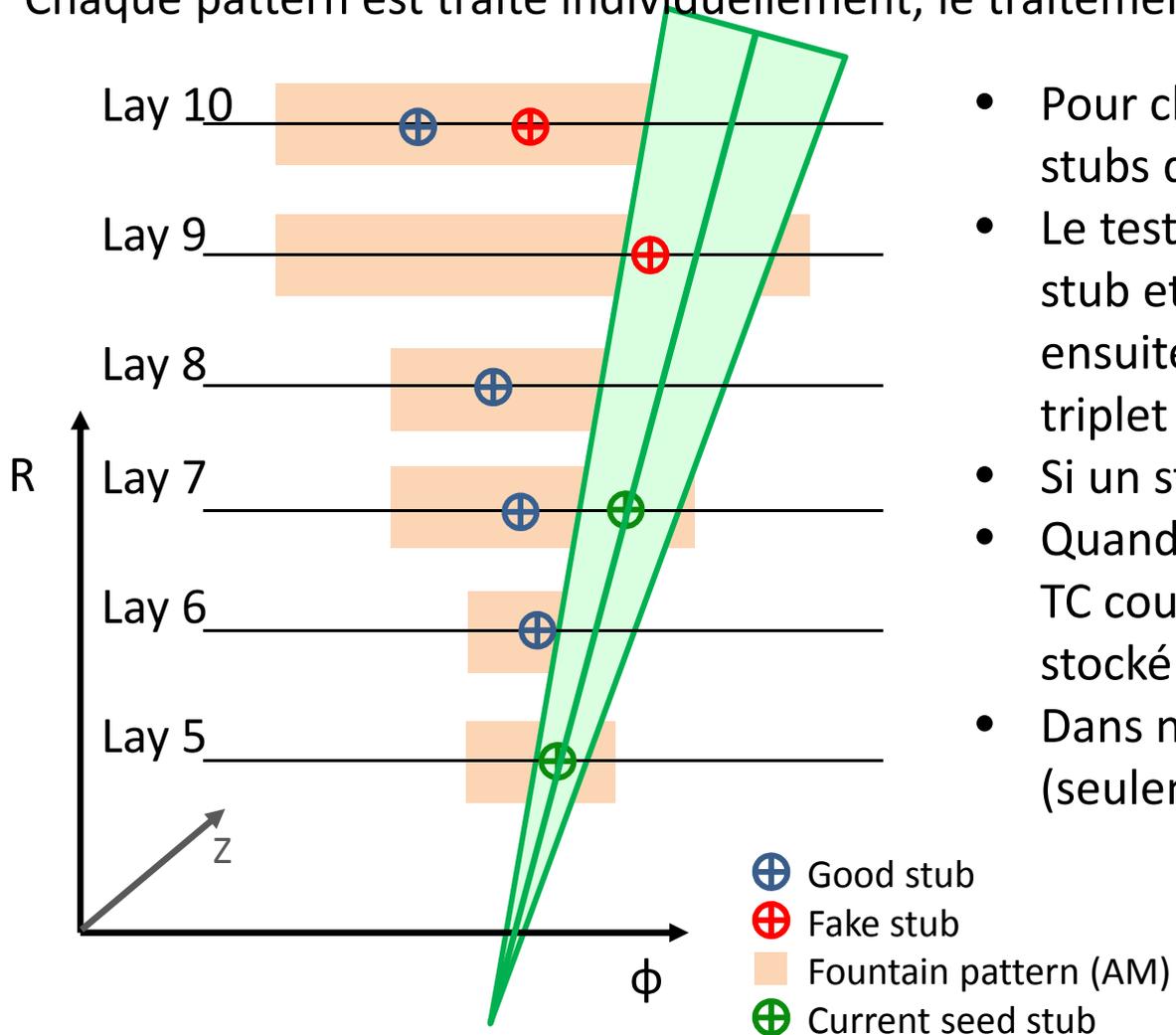
Chaque pattern est traité individuellement, le traitement est opéré simultanément sur  $(R, \phi)$  et  $(R, Z)$



- La première étape consiste à générer toutes les paires possibles de stubs des layers internes
- Ces paires sont les seeds 3D de notre algorithme
- $N_{seeds} = (N_5 * N_6) + (N_6 * N_7) + (N_5 * N_7)$
- Dans notre exemple il y a 5 seeds différentes

## Présentation de l'algorithme de filtrage

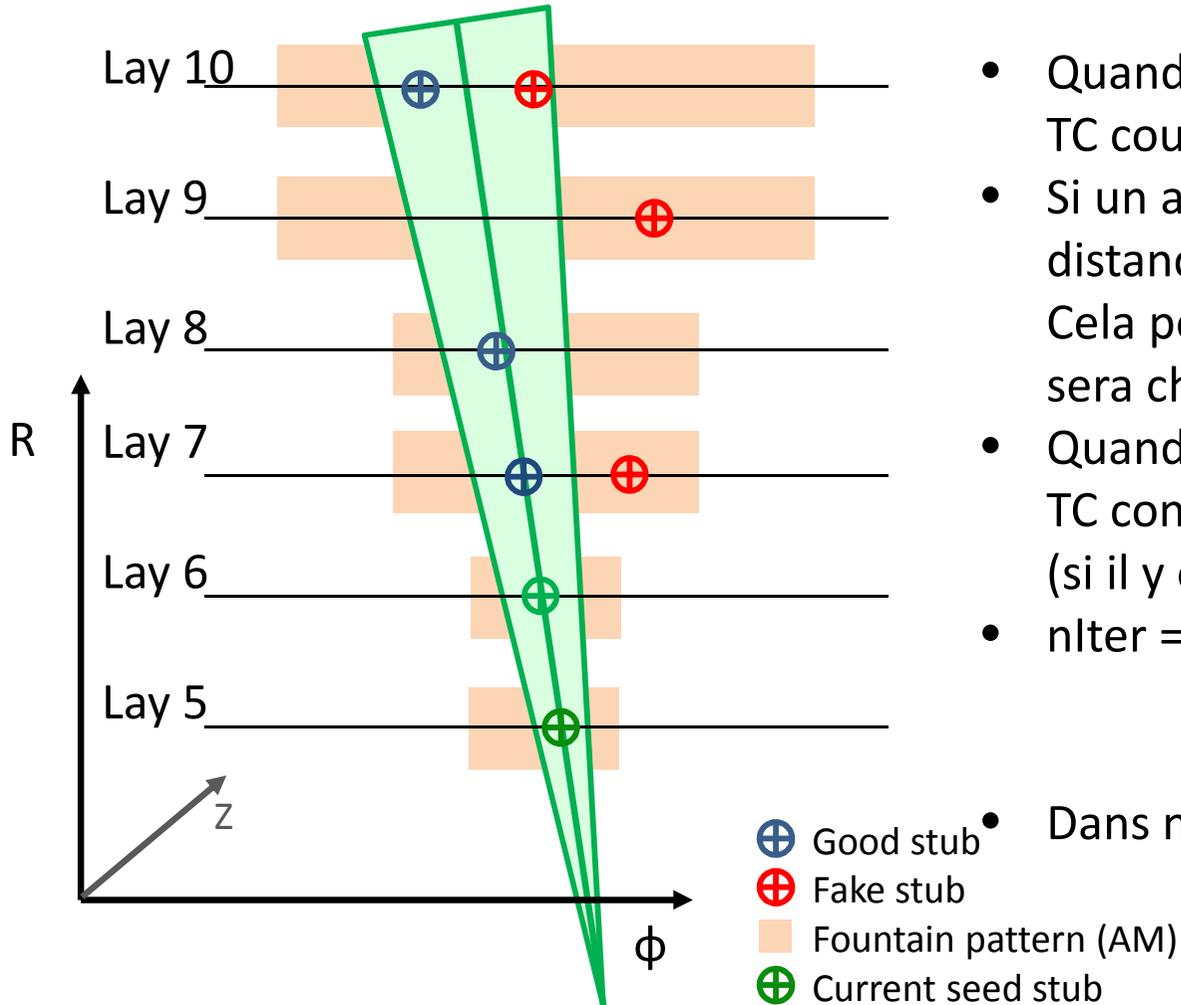
Chaque pattern est traité individuellement, le traitement est opéré simultanément sur  $(R, \phi)$  et  $(R, Z)$



- Pour chaque seed 3D, un test est opéré sur tous les stubs des layers plus externes.
- Le test consiste en un calcul de distance entre le stub et la projection de la seed 3D. Cette valeur est ensuite comparée à un seuil (ce seuil dépend du triplet de layers concernés).
- Si un stub passe le test, il est ajouté au TC courant
- Quand tous les stubs externes ont été testés, si le TC courant contient au moins 5 stubs, le TC est stocké dans une mémoire intermédiaire.
- Dans notre exemple le TC n'est pas retenu (seulement 3 stubs présents dans le TC).

## Présentation de l'algorithme de filtrage

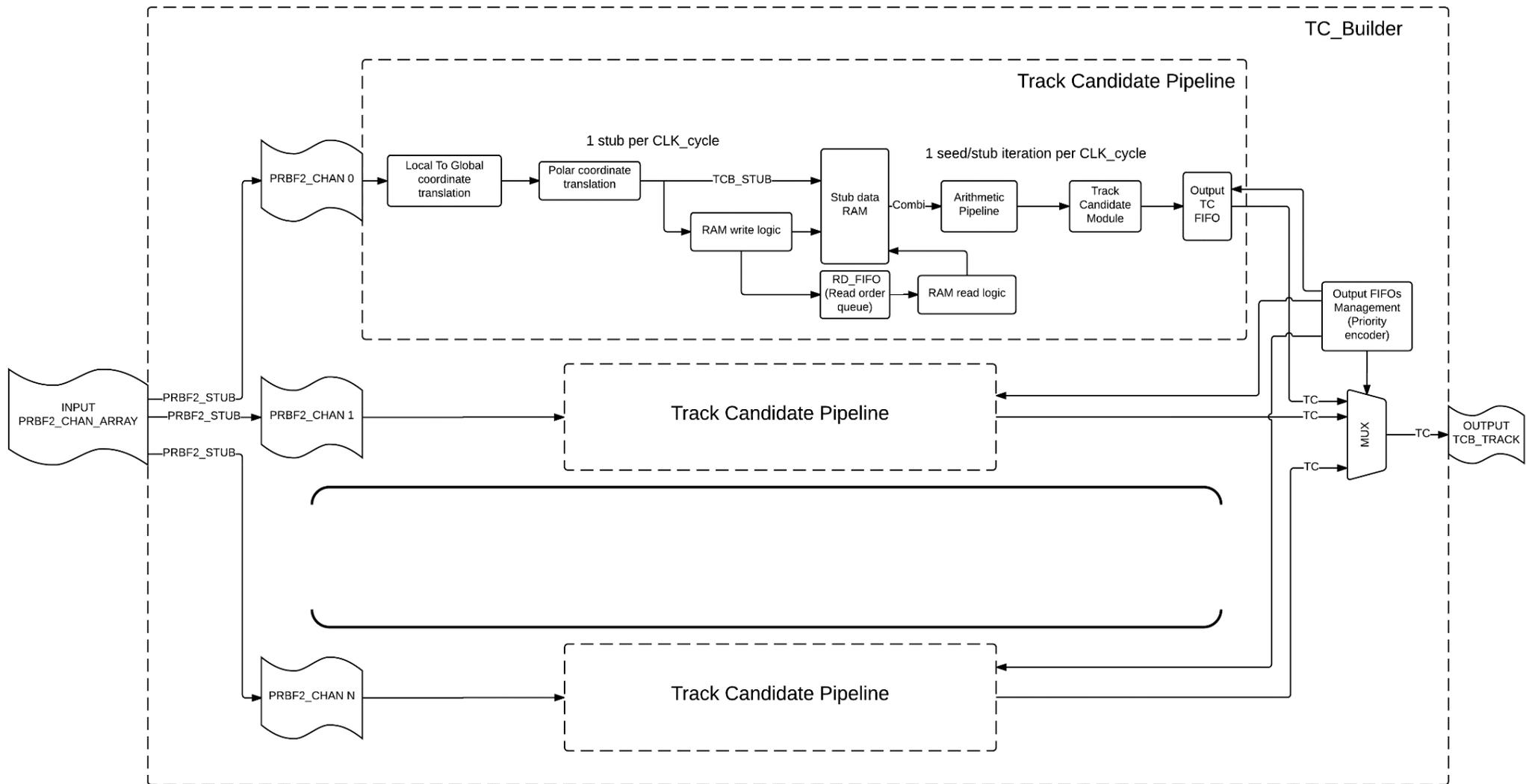
Chaque pattern est traité individuellement, le traitement est opéré simultanément sur  $(R, \phi)$  et  $(R, Z)$



- Quand un stub passe le test et est enregistré dans le TC courant, la distance seed/stub est mémorisée.
- Si un autre stub passe le test pour le même layer, les distances d'alignement sur  $(R, \phi)$  sont comparées. Cela permet d'assurer que le meilleur alignement sera choisit pour tous les layers.
- Quand les itérations sur les seeds sont terminées, la TC contenue dans le stockage intermédiaire est lue (si il y en a une), jamais plus d'un TC par pattern.
- $nIter = (N5 * N6) * (N7 + N8 + N9 + N10) + (N6 * N7) * (N8 + N9 + N10) + (N5 * N7) * (N8 + N9 + N10)$
- Dans notre exemple  $nIter = 22$

# Track Candidate Builder

## Architecture



## Banc de test du système complet

Développement d'une carte mezzanine basée sur une puce AM05

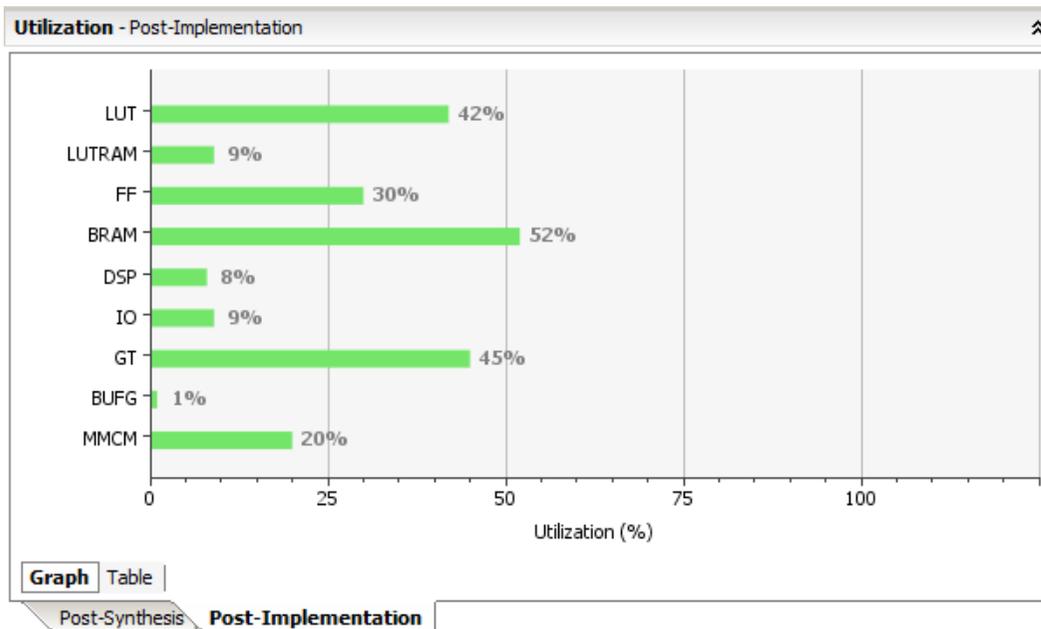
- 2048 patterns
- 8 bus série 2Gb/s en entrée
- 1 bus série 2Gb/s en sortie



Création d'un banc de test sur une carte d'évaluation Xilinx

- Communication PC/carte par IPBus à 1Gb/s
- Reconnaissance de patterns sur puce AM05
- Reconstruction/Filtrage de traces dans le FPGA

Ressources nécessaires à l'implémentation du firmware de la **chaîne complète** sur un FPGA Kintex Ultrascale (banc de test basé sur une carte d'évaluation kcu105)



## Data Organizer

- Traitement de 6 layers
- Possibilité de stockage de 1000 stubs/layer
- 20 pattern builder en sortie

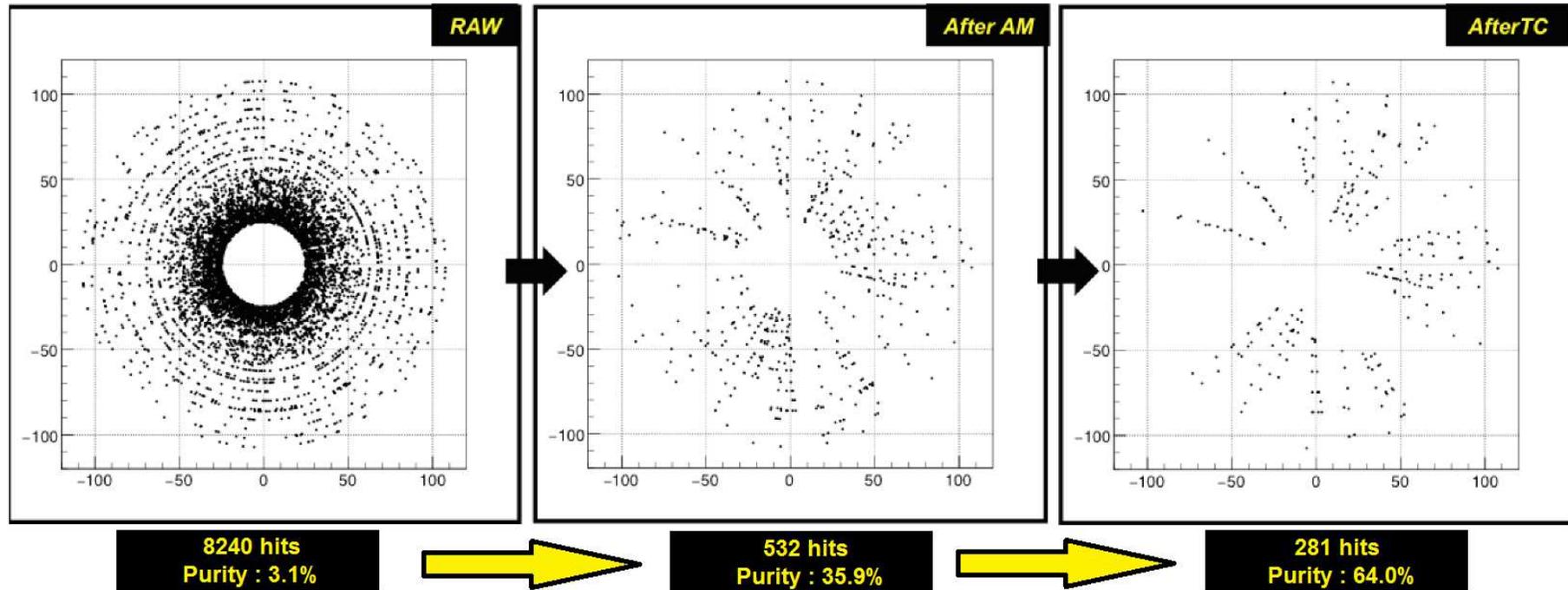
## Track Candidate Builder

- 10 Track Candidate Pipelines instanciés en parallèle
- Coordonnées codées sur 18 bits

Utilization - Post-Implementation

Resource	Utilization	Available	Utilization %
LUT	100722	242400	41.55
LUTRAM	9665	112800	8.57
FF	147539	484800	30.43
BRAM	310	600	51.67
DSP	151	1920	7.86
IO	49	520	9.42
GT	9	20	45.00
BUFG	6	480	1.25
MMCM	2	10	20.00

Efficacité (obtenues en simulation)



	Muons		Electrons		PU140		PU200	
	$p_T \geq 3$	$p_T \geq 10$	$p_T \geq 3$	$p_T \geq 10$	$p_T \geq 3$	$p_T \geq 10$	$p_T \geq 3$	$p_T \geq 10$
$\epsilon^{SR}$	100.0 $_{-0.05}$	100.0 $_{-0.05}$	97.0 $\pm$ 0.1	98.4 $\pm$ 0.1	98.0 $\pm$ 0.1	98.4 $\pm$ 0.2	98.1 $\pm$ 0.1	98.5 $\pm$ 0.2
$\epsilon^{TW}$	99.9 $\pm$ 0.05	99.9 $\pm$ 0.05	99.8 $\pm$ 0.05	99.8 $\pm$ 0.05	100.0 $_{-0.05}$	100.0 $_{-0.05}$	100.0 $_{-0.01}$	100.0 $_{-0.05}$
$\epsilon^{AM}$	99.0 $\pm$ 0.05	99.1 $\pm$ 0.05	94.9 $\pm$ 0.1	95.8 $\pm$ 0.1	97.8 $\pm$ 0.1	99.5 $\pm$ 0.1	97.8 $\pm$ 0.1	99.3 $\pm$ 0.2
$\epsilon^{CB}$	99.9 $\pm$ 0.05	99.9 $\pm$ 0.05	98.4 $\pm$ 0.1	98.5 $\pm$ 0.1	99.3 $\pm$ 0.05	99.4 $\pm$ 0.1	99.1 $\pm$ 0.1	99.3 $\pm$ 0.2
$\epsilon^{SR \rightarrow CB}$	98.8 $\pm$ 0.05	98.9 $\pm$ 0.05	91.4 $\pm$ 0.2	92.8 $\pm$ 0.2	95.2 $\pm$ 0.1	97.3 $\pm$ 0.2	95.0 $\pm$ 0.1	97.2 $\pm$ 0.3

## Conclusions sur le système actuel

- Méthode de reconstruction 3D permettant un filtrage très sélectif (très bonne réjection des fakes)
- La grande pureté des Track Candidate obtenus garantie d'excellents résultats post-fit
- Résultats en efficacité de reconstruction et latence (simulation) très encourageants
- Ressources utilisées pour l'implémentation du premier prototype acceptables

## Futurs développements

- Upgrade du banc de test présent à l'IPNL (nouveau chip AM, raffinement firmware,...)
- Validation des résultats de simulation de la chaîne complète grâce au HW
- Participation à la mise en place du banc de test commun au CERN