

Gestion des opérateurs reconfigurables au sein des circuits

Jean-Christophe Prévotet

IETR

jean-christophe.prevotet@insa-rennes.fr



Plan de la présentation

- Contexte
 - Rappel sur les architectures reconfigurables
- Reconfiguration statique
 - Exemple sur FPGA
- Reconfiguration dynamique
 - Exemple sur RSoC
- Vers une reconfiguration "intelligente" ...

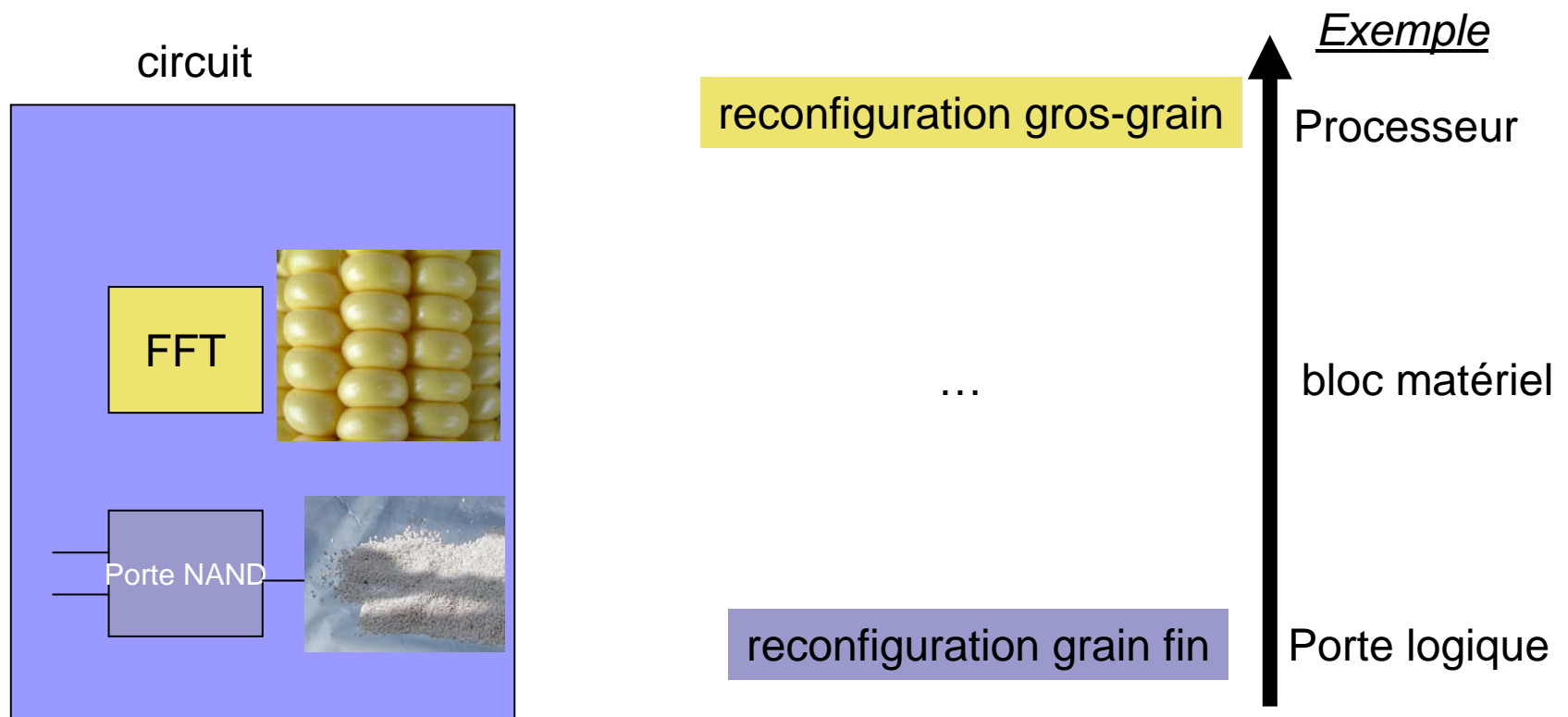


Contexte

- Une architecture **matérielle** reconfigurable peut modifier sa fonctionnalité
 - Elle permet de s'adapter à un traitement au cours du temps
- La structure du circuit n'est pas figée !
- La reconfiguration peut s'effectuer à plusieurs niveaux

Une histoire de grains...

- Architectures matérielles reconfigurables

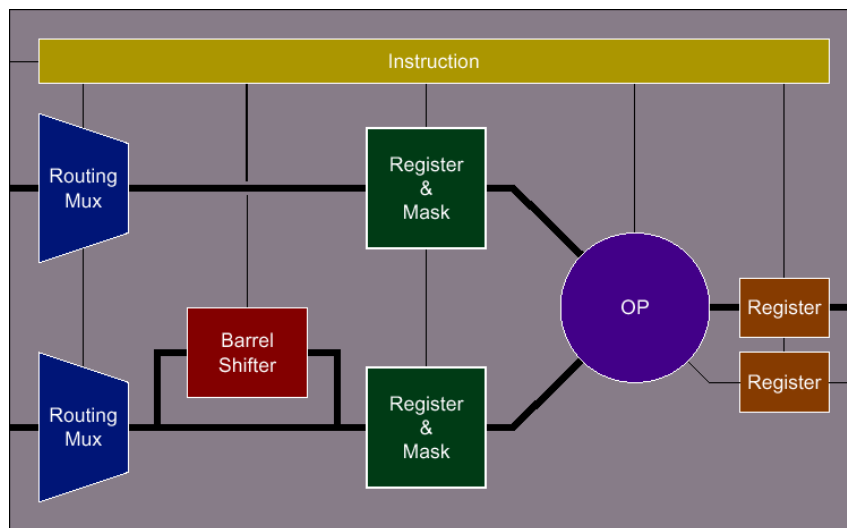
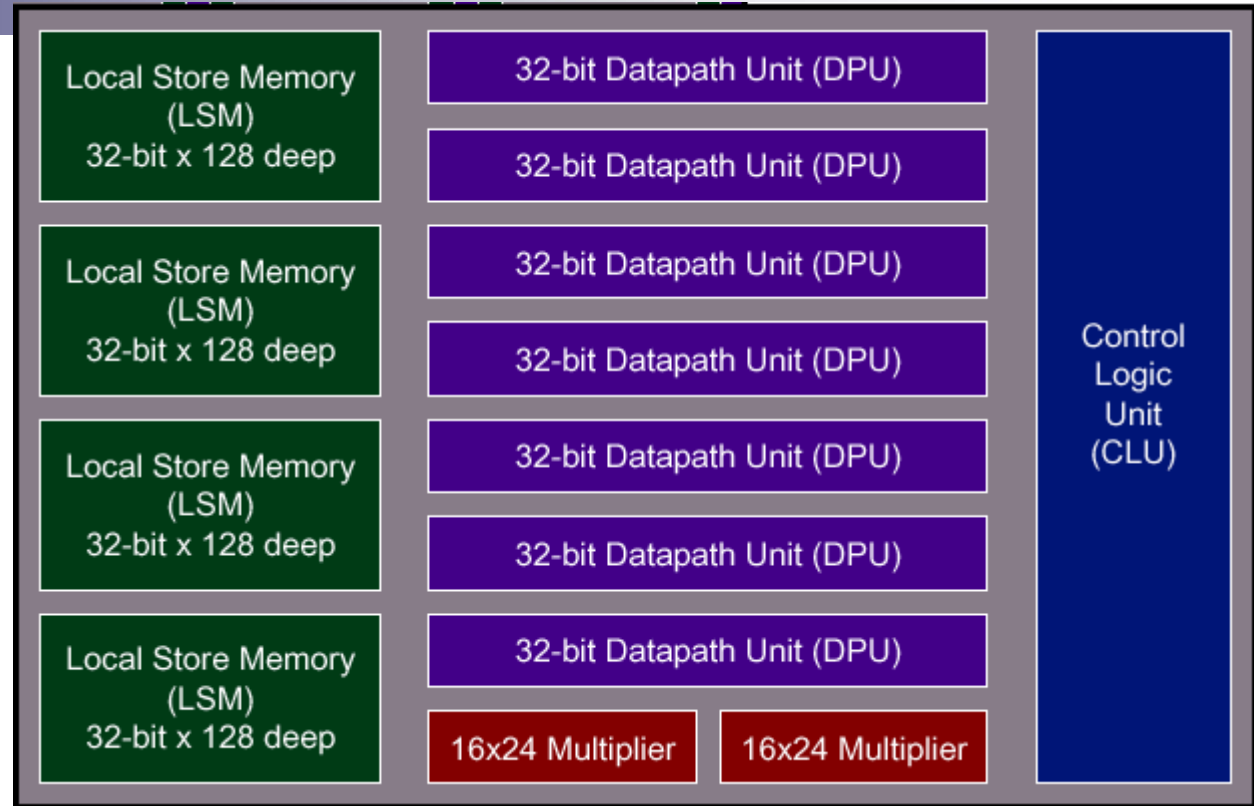




Architectures gros-grain

- La configuration est réalisée au sein d'opérateurs matériels
 - Configuration d'une UAL : choix de l'opération à effectuer
 - Configuration d'un bloc FFT : choix sur le nombre de points
 - ...
- Avantages
 - Rapidité de la configuration
- Inconvénients
 - Moindre flexibilité
 - Opérateurs plus génériques donc moins optimisés :
 - On ne peut réaliser que quelques fonctions logiques

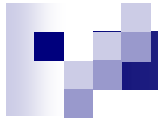
Exemple



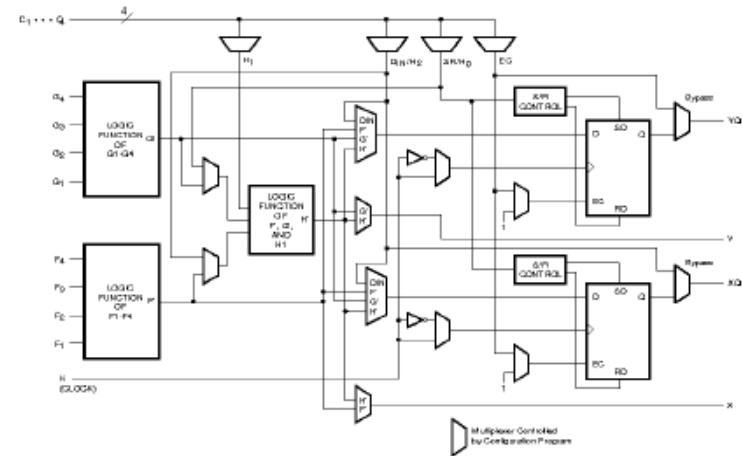
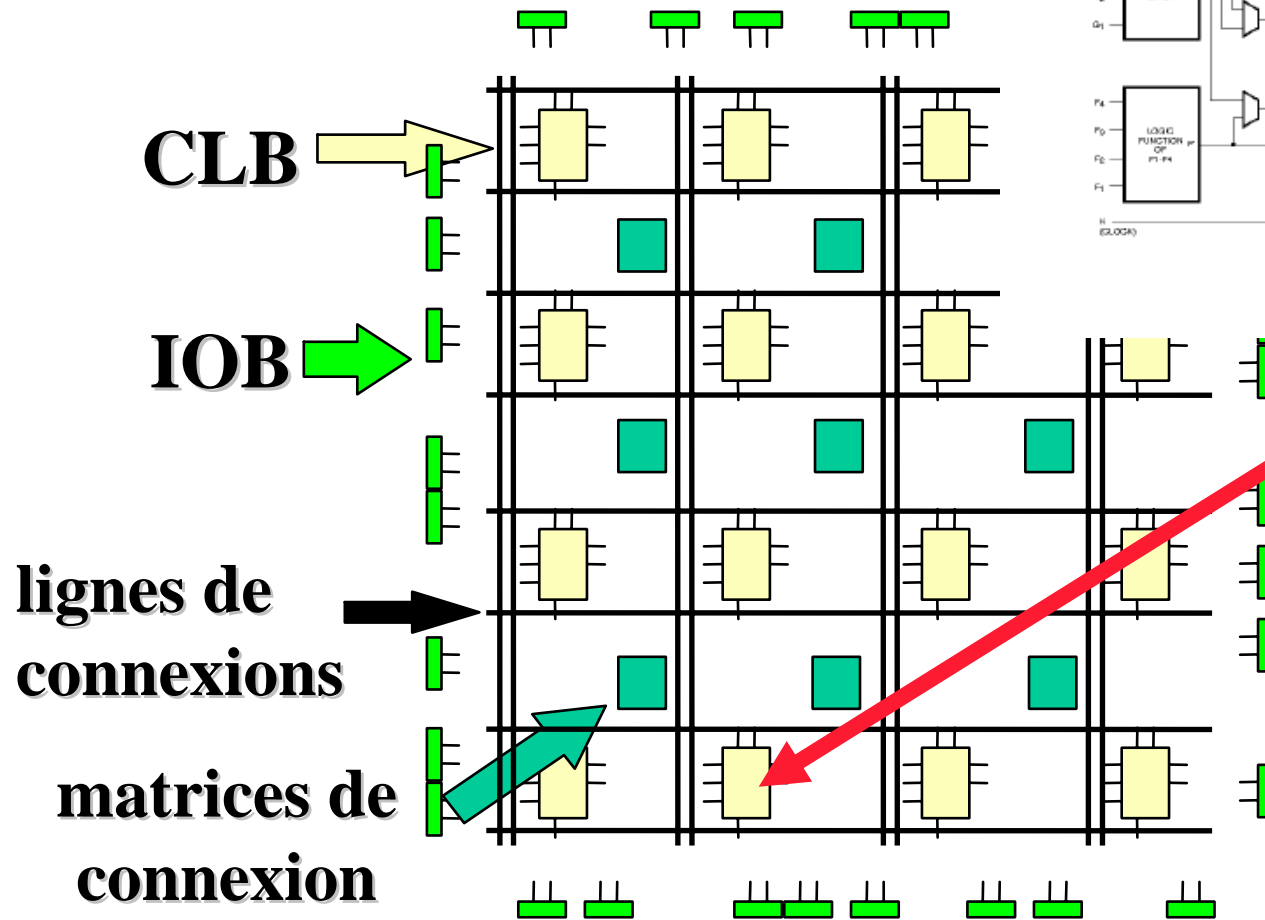


Architecture grain-fin

- Configuration au niveau logique
 - Implantation d'une fonction logique
 - Proche du niveau porte
- Avantages
 - Grande flexibilité
 - Optimisation du matériel
- Inconvénients
 - Temps de la reconfiguration (qq ms)



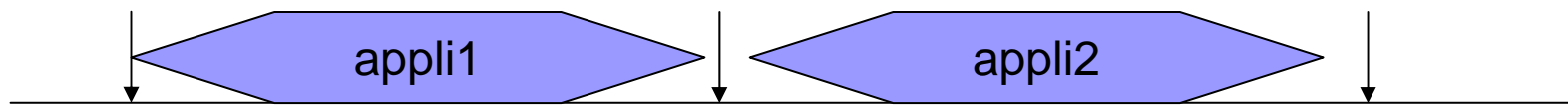
Exemple : le FPGA



Types de reconfigurations

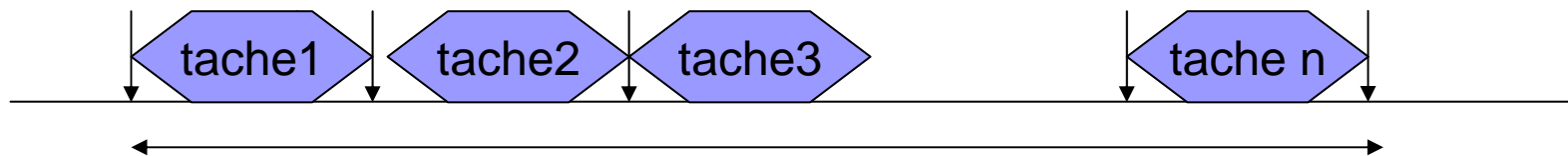
■ Configuration statique

- exécution d'un traitement global entre deux configurations



■ Configuration dynamique

- exécution d'une partie de traitement entre deux configurations

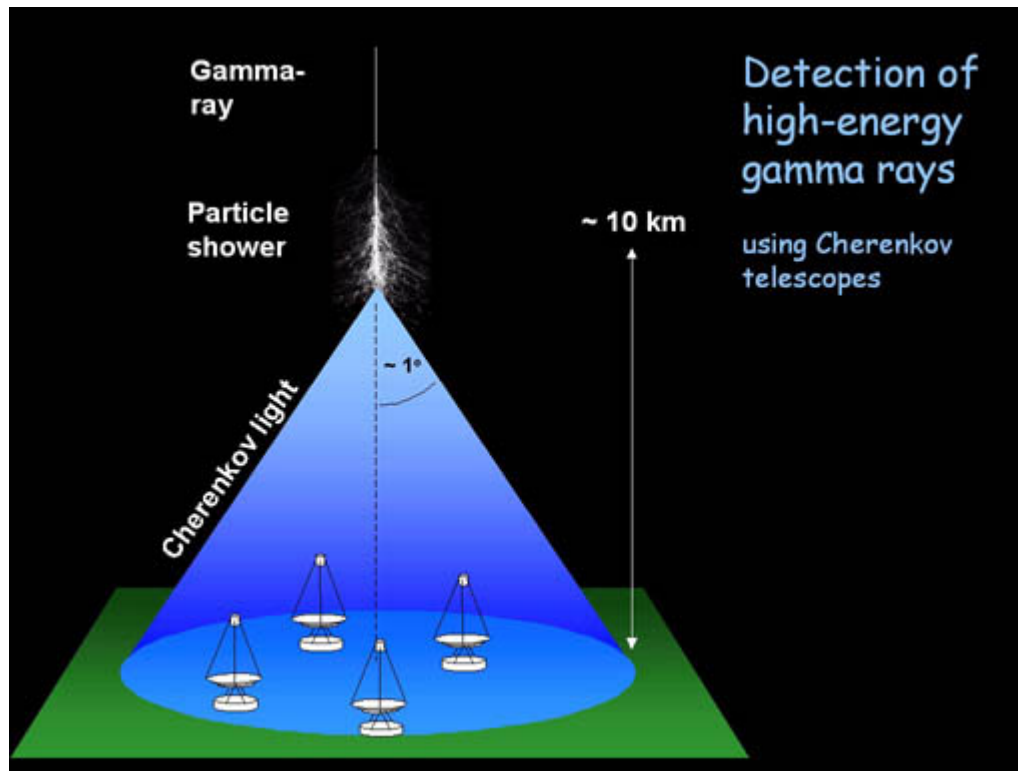




Reconfiguration statique

Exemple d'application

Application en physique



Astronomie Gamma

Objectif

Détection de sources ponctuelles de rayonnement gamma par analyse des particules secondaires émises par effet Cherenkov

Le projet HESS



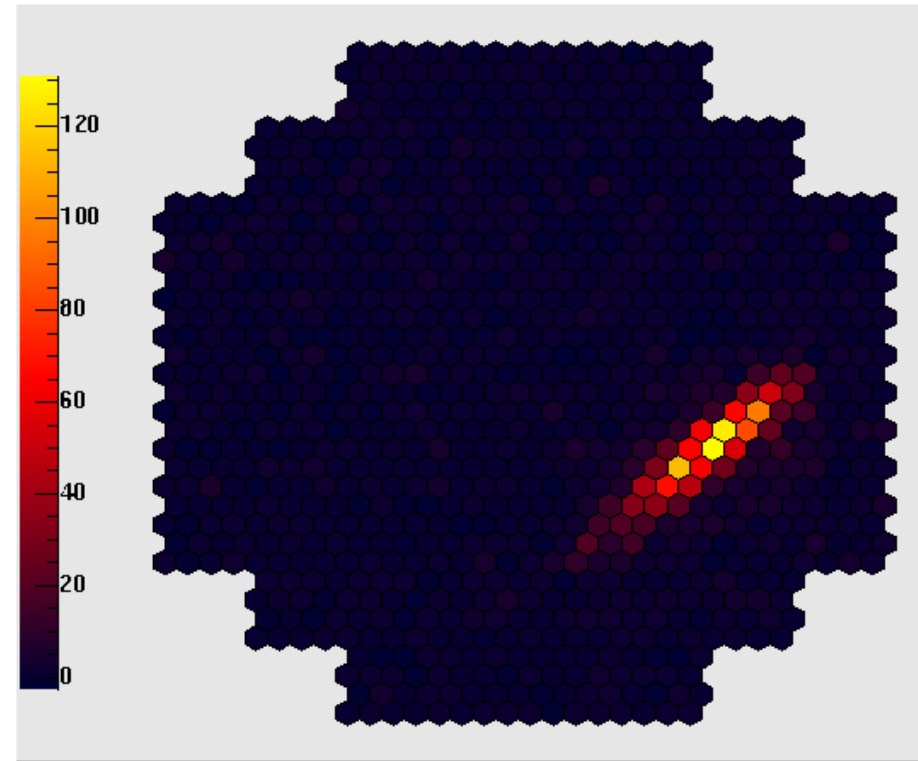
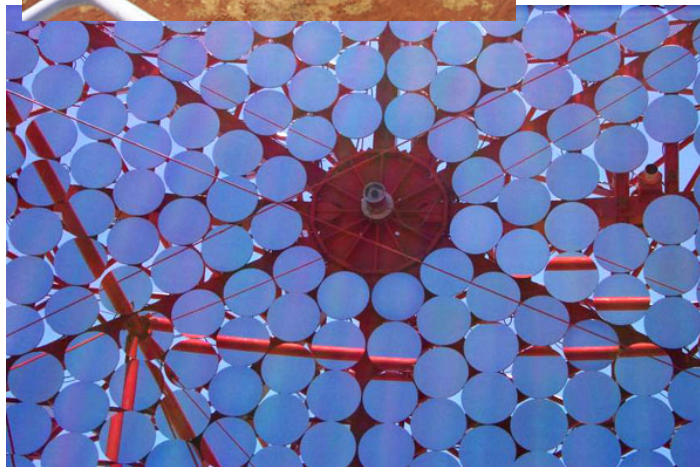
Quatre télescopes situés en Namibie

Collaboration d'une centaine de chercheurs venant d'une dizaine de pays

Amélioration de l'expérience : HESS2 (2009)

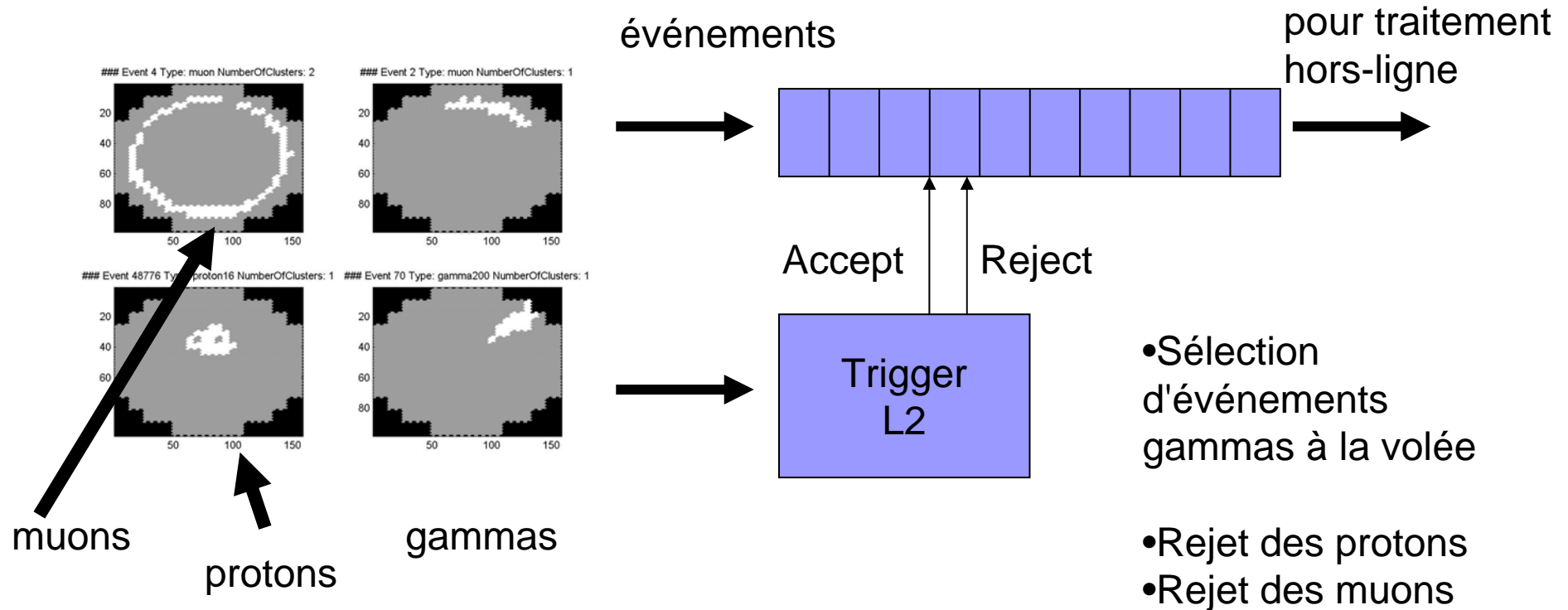
- ajout d'un télescope

Exemple d'image issue d'un télescope



2048 pixels binaires

Le Trigger



Thèse : Sonia Khatchadourian (3^{ème} année)

Collaboration ETIS / LLR /CEA

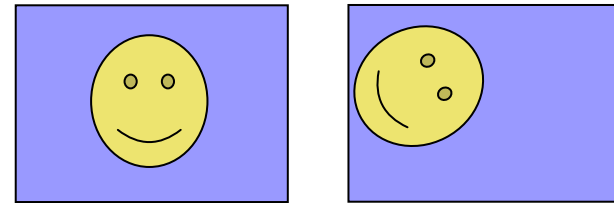
Conception du trigger L2

Traitement envisagé et contraintes

■ Prétraitement

- Invariance en translation et rotation des signatures de particules

- moments de Zernike

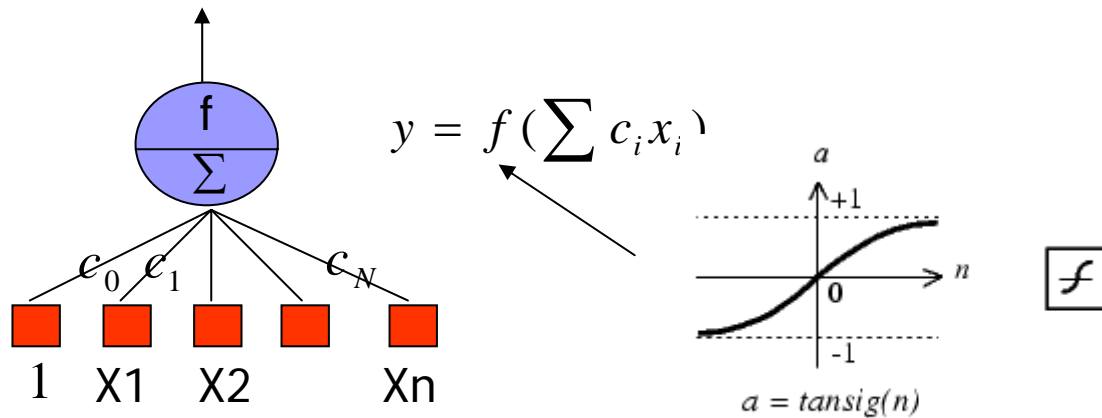


■ Classifieur

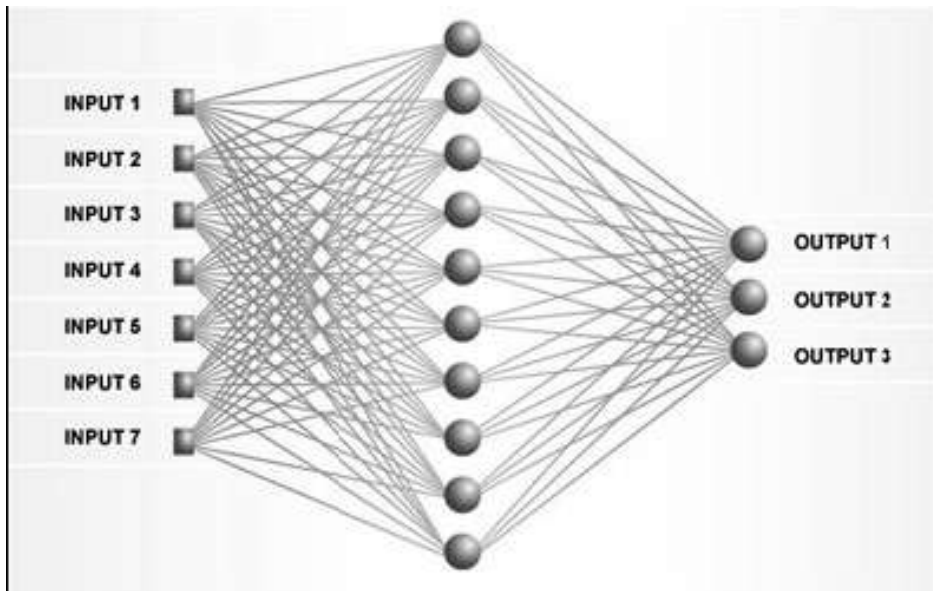
- basé sur les réseaux de neurones

- PMC à 3 sorties (gammas, muons, protons)
- Apprentissage hors-ligne en fonction du contexte
 - Détection de galaxies
 - Analyse de sources

Réseau de neurones



Tan-Sigmoid Transfer Function



Les paramètres (poids) du réseau de neurones sont modifiés dans l'étape d'apprentissage.

L'apprentissage est réalisé hors-ligne.



Architecture proposée et implantation matérielle

- Architecture sur cible FPGA
 - Facilite le parallélisme
 - Possibilité de traitement en 10 μ s
 - pré-traitement + réseau de neurones
- Gestion de la reconfiguration **statique**
 - effectuée par un contrôleur en fonction des mesures à effectuer
 - Détection de gammas
 - Rejet des autres particules



Où intervient la reconfiguration ?

■ Pré-traitement

- modification du nombre de moments géométriques pris en compte
- modification du type de pré-traitement
 - pré-traitement plus efficace ?

■ Réseau neuronal

- Modification de la structure du réseau
 - tailles des couches (entrée, cachée, sortie)
- Modification des paramètres
 - Poids spécifiques en fonction du contexte



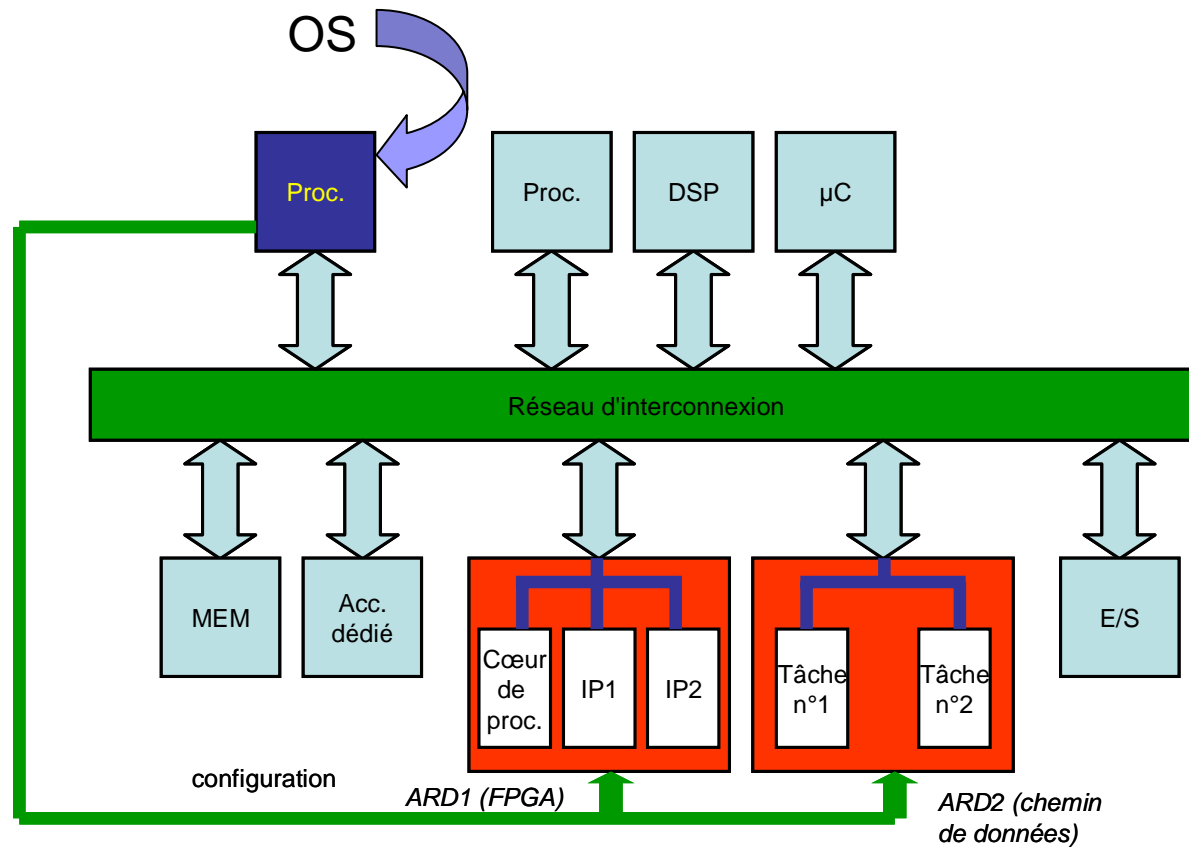
Résumé

- **Implantation matérielle reconfigurable**
 - Possibilité de calculer le pré-traitement + réseaux de neurones en $10\mu\text{s}$ (impossible en logiciel...)
 - Architecture parallèle
 - Reconfiguration des paramètres du pré-traitement
 - Modification des poids du réseau suivant le contexte



Reconfiguration **dynamique** Gestion par un OS

Contexte : RSoC





Gestion du reconfigurable

■ Problématique

- Notion de tâche matérielle
- Les tâches doivent s'échanger des données
 - SW/SW, SW/HW, HW/HW
- Elles doivent être créées, détruites dynamiquement
- L'OS doit pouvoir gérer les tâches logicielles et matérielles
- Il doit posséder les services appropriés
 - Séquenceur, communication...

Quels sont les services à mettre en œuvre ?



Le projet OveRSoC

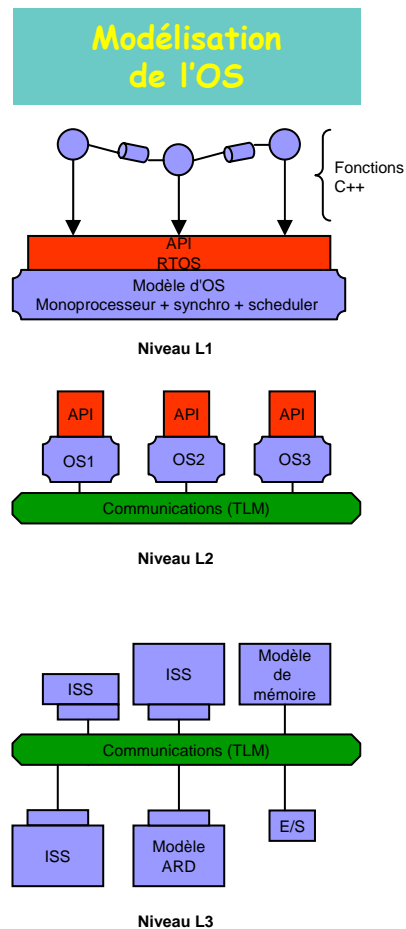
■ Projet ANR

- Equipe R2D2, ETIS (Cergy Pontoise), LISIF (Paris VI)

■ Objectifs

- Proposer une méthodologie d'exploration des services d'un OS pour la gestion du reconfigurable dynamique (niveau système)
 - Modélisation en SystemC
- Proposer un outil associé à cette méthodologie

Modèle d'OS



- Modèle exécutif (L1) :
 - 1 tâche principale
 - lance le scheduler
 - crée les autres tâches.
 - L'objectif principal est l'exploration du séquençement global des opérations.
- Modèle de distribution (L2) :
 - Prise en compte d'un OS distribué.
- Modèle d'implantation (L3) :
 - Les différents modules de la plateforme peuvent être décrits à plusieurs niveaux
 - L'OS tourne sur un modèle de processeur ou d'ARD



Modèle d'archi. reconfigurable

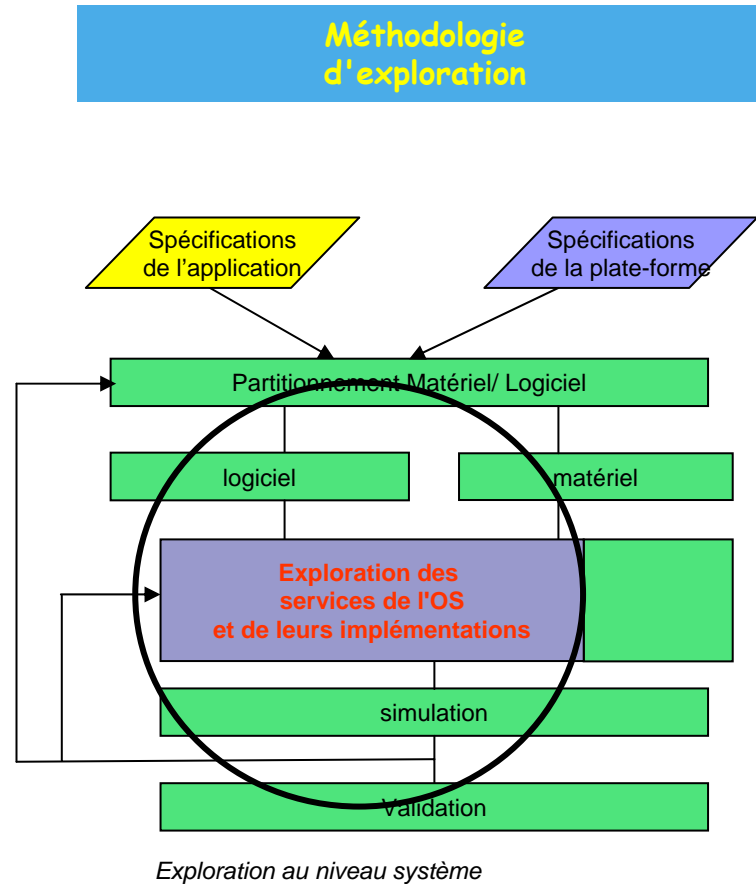
Modélisation
des unités de calcul
reconfigurables

Définition des services spécifiques :

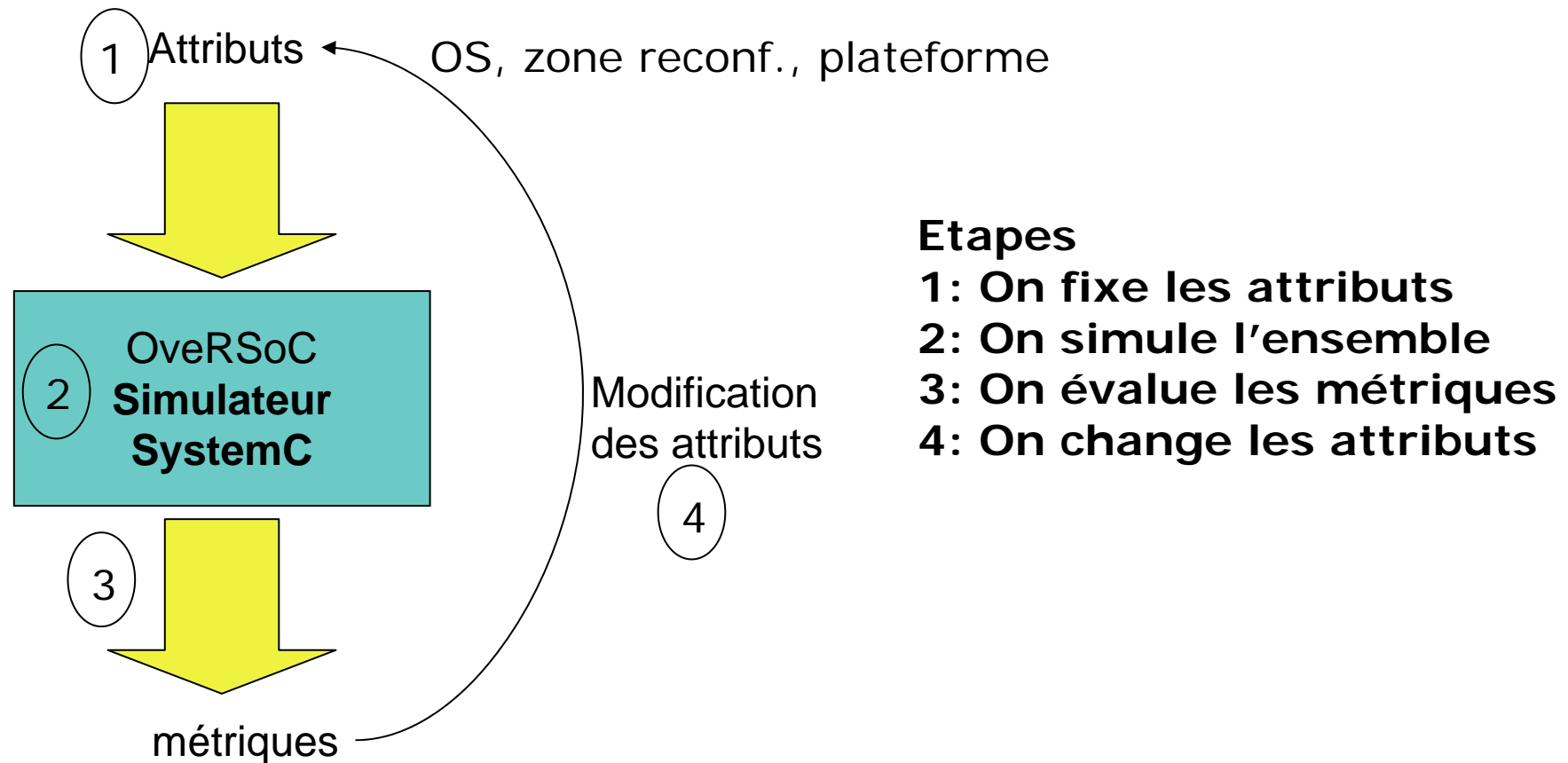
- Gestion des communications reconfigurables
- Gestion du placement des tâches
- Gestion de la migration
- Gestion de la préemption

- Modélisation à plusieurs niveaux :
 - Du placement des tâches dans une ARD
 - De la migration de tâches
 - SW/HW
 - HW/HW
 - HW/SW
 - De la préemption
 - Des communications au sein de la zone reconfigurable

Méthodologie d'exploration



Méthodologie d'exploration (suite)



L'outil OverSoC

The image displays the OverSoC tool interface, which is used for SoC design. It features a main workspace with a Navigator, Outline, and Platform View. The workspace shows a hardware diagram with components like 'mips1', 'mips1_1', 'bus', and 'memory'. Three dialog boxes are overlaid on the workspace, each providing configuration options for different components.

Attributs de la mé

ATT_Mem_CLK	Horloge
ATT_Mem_LENGTH	Taille de la mémoire
ATT_Mem_WIDTH	Taille de la mémoire
ATT_Mem_TYPE	

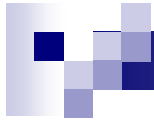
Attributs processeur

ATT_Proc_ClkFreq	Fréquence du processeur
ATT_Proc_Name	Nom du processeur
ATT_Proc_Type	Type de processeur

Attributs APD

ATT_DRA_TYPE	
ATT_DRA_Available_Resources	Ressources disponibles
ATT_DRA_Reconfiguration_Time	Temps de reconfiguration de la DRA
ATT_DRA_IOs	Description des E/S
ATT_DRA_Number_of_IOs	Nombre d'E/S

The hardware diagram shows a central 'Réseau d'interconnexion' (Interconnection Network) connecting various components: Proc., Proc., DSP, µC, Mem, Acc. Gedé, ARD1 (Core, IP1, IP2), ARD2 (TACH, TACH), and E/S.



vers une reconfiguration
"intelligente"...



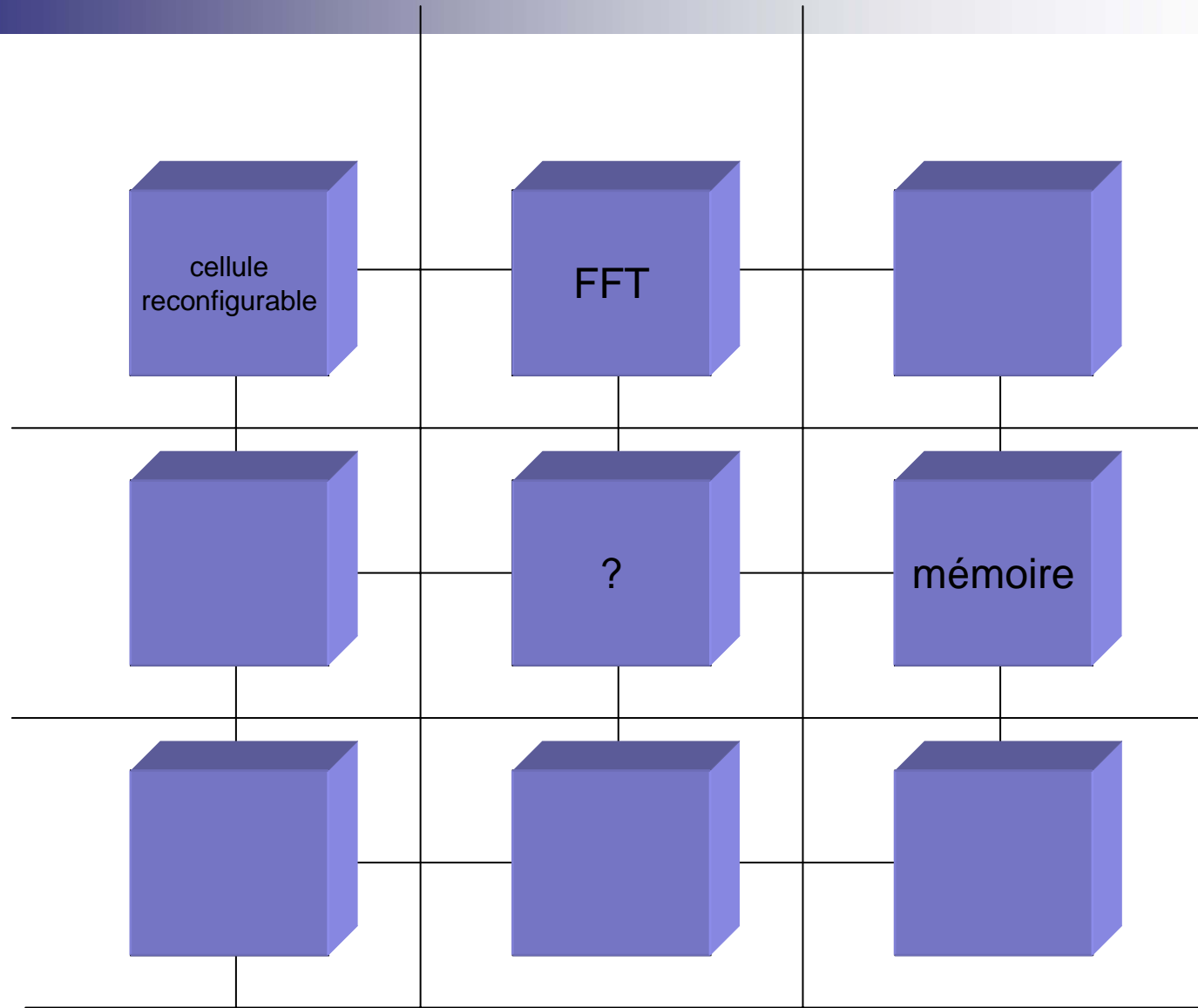
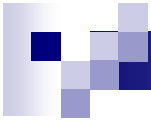
Contexte

- Les circuits deviennent de plus en plus complexes (millions de portes à gérer)
- La gestion de ces circuits est devenu un enjeu majeur
 - Hors ligne :
 - manque d'outils performants pour optimiser l'archi
 - Pas assez de dynamicité
 - En-ligne :
 - Souvent trop complexe

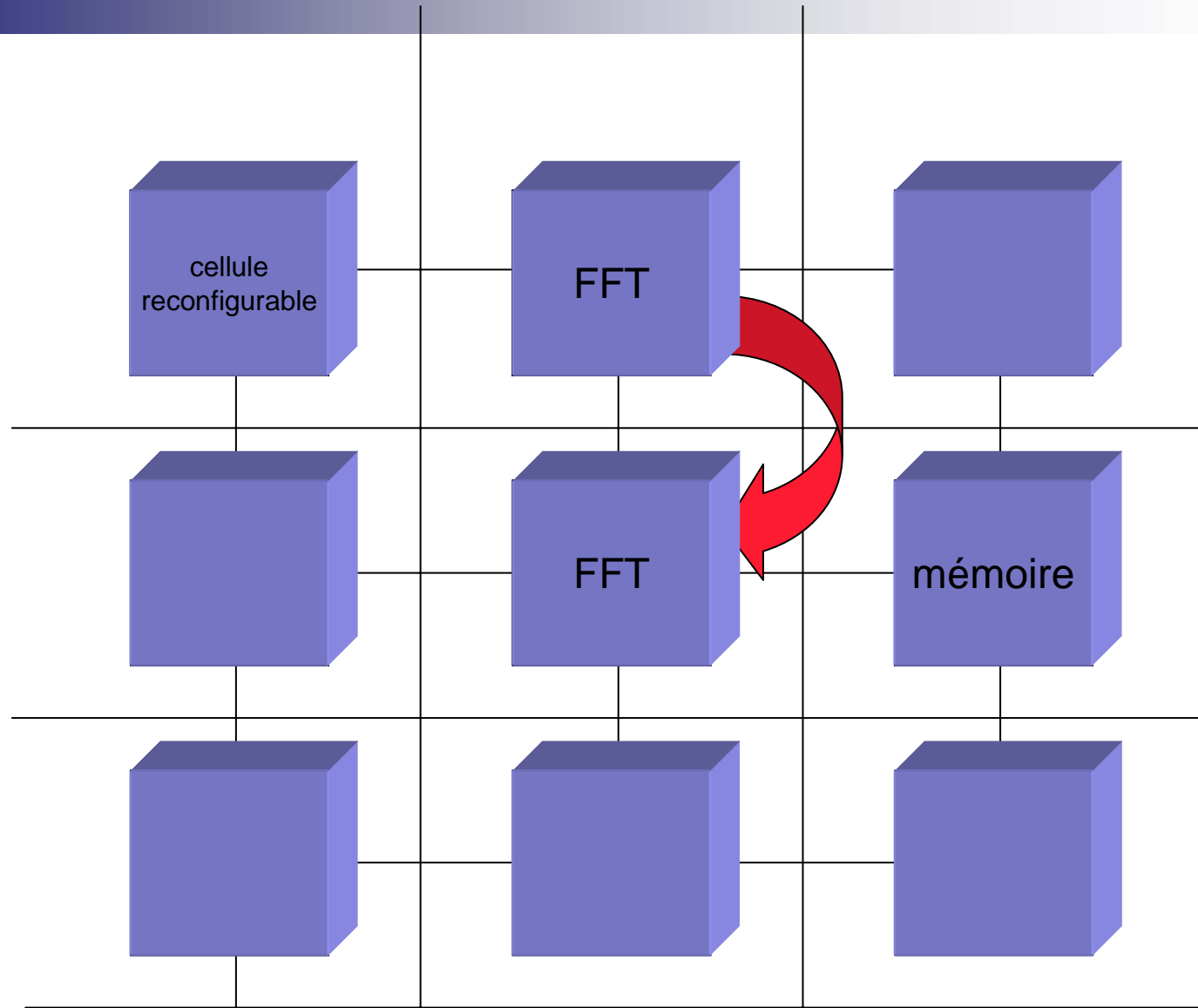
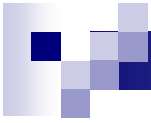


Idée

- Concevoir des architectures pouvant s'auto-gérer
 - capable de modifier leur architecture interne en fonction de leur environnement et au cours du temps
 - limiter voire éliminer le contrôle global
 - La gestion est localisée
 - Architecture multi-cellulaire ou chaque cellule est indépendante
 - Une cellule est capable d'interagir avec les autres cellules
 - création, destruction, etc.



Réseau d'opérateurs reconfigurables



le traitement a la possibilité de se paralléliser : création d'une nouvelle cellule



Problèmes à résoudre (en vrac)

- Flot de données à travers l'architecture
- Définition d'un opérateur générique
- Mise en œuvre de la reconfiguration
 - mécanismes de gestion
- Contraintes à l'exécution (convergence ?)
- Métriques
- et tant d'autres...



Conclusion

- Le reconfigurable constitue une voie intéressante pour les circuits
- La gestion statique des circuits a fait ses preuves 😊
- La gestion dynamique n'est pas encore pour demain ☹️
 - Les technologies ne le permettent pas encore...
 - Les outils non plus
 - C'est à la communauté de présenter des applications 😊
- La complexité des circuits tend à les rendre de plus en plus difficile à gérer dynamiquement ☹️
 - Besoin d'OS
 - Il reste à inventer d'autres moyens de gestion ...😊