Processeur reconfigurable pour applications multimédia

Projet ROMA

Reconfigurable Operators for Multimedia Applications

Partenaires du projet

- Industriel
  - Thomson Silicon Components (Rennes)
- Académiques
  - CEA - LIST (Saclay)
  - IRISA – Univ. Rennes1 (Rennes/Lannion)
  - LIRMM (Montpellier)

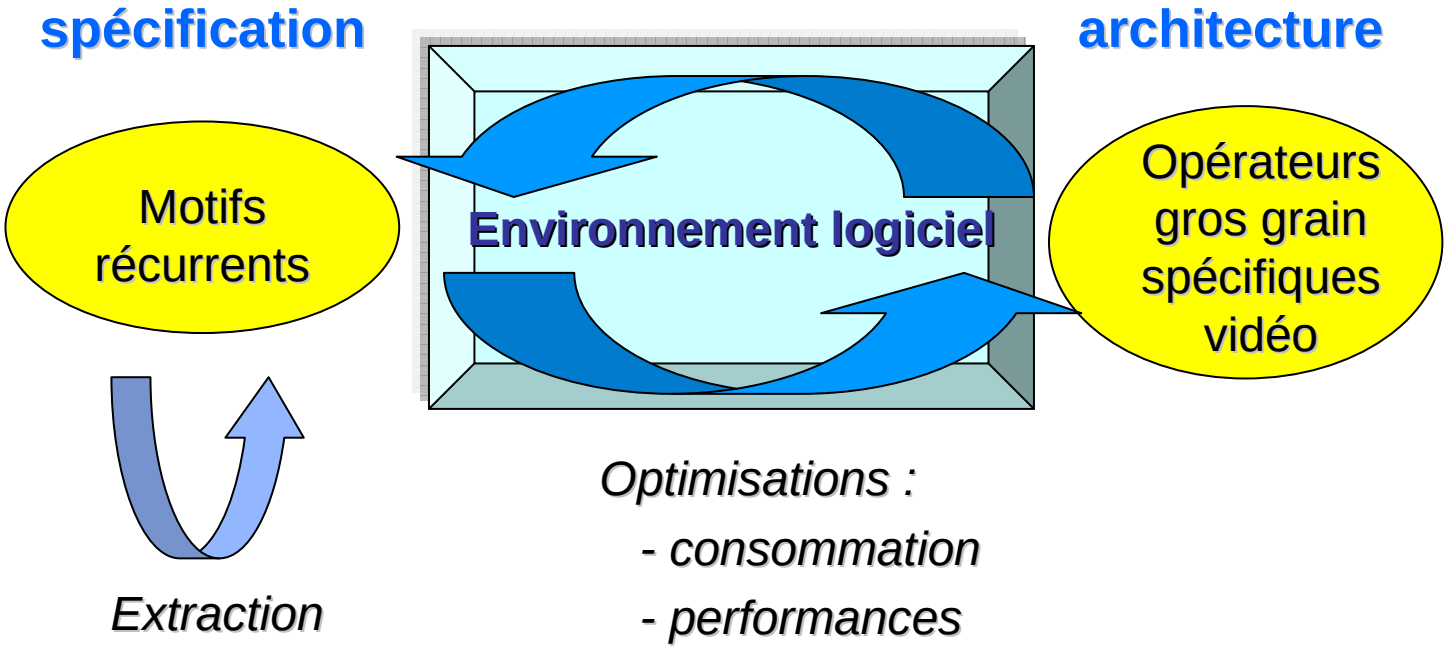
Partenaires du projet

- Labellisé Pole Images et Réseaux (octobre 2006)

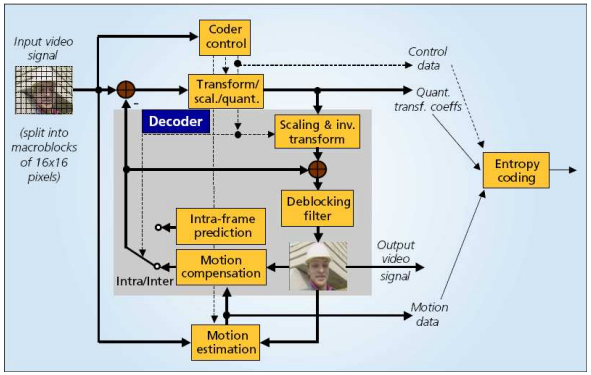
ROMA : objectif

- Conception d'un **processeur reconfigurable** capable **d'adapter sa structure** de traitement aux **motifs de calculs** pour lesquels un gain significatif en termes de performances et de consommation est attendu
  - Architecture à base d'opérateurs à gros grain
  - Utilisation d'un codage approprié pour la représentation des données + dimensionnement des données
  - Environnement de compilation/configuration approprié
  - Domaine d'applications spécifique : vidéo

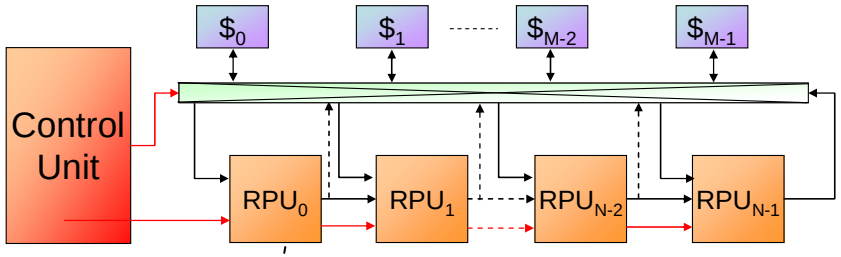
ROMA : principe



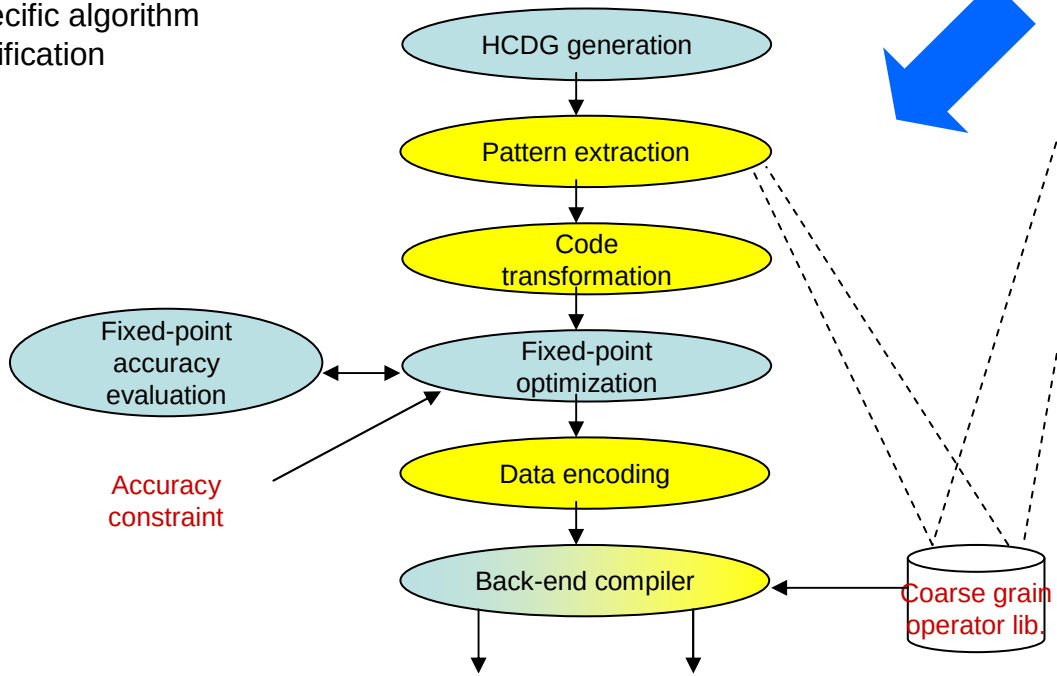
ROMA: principe



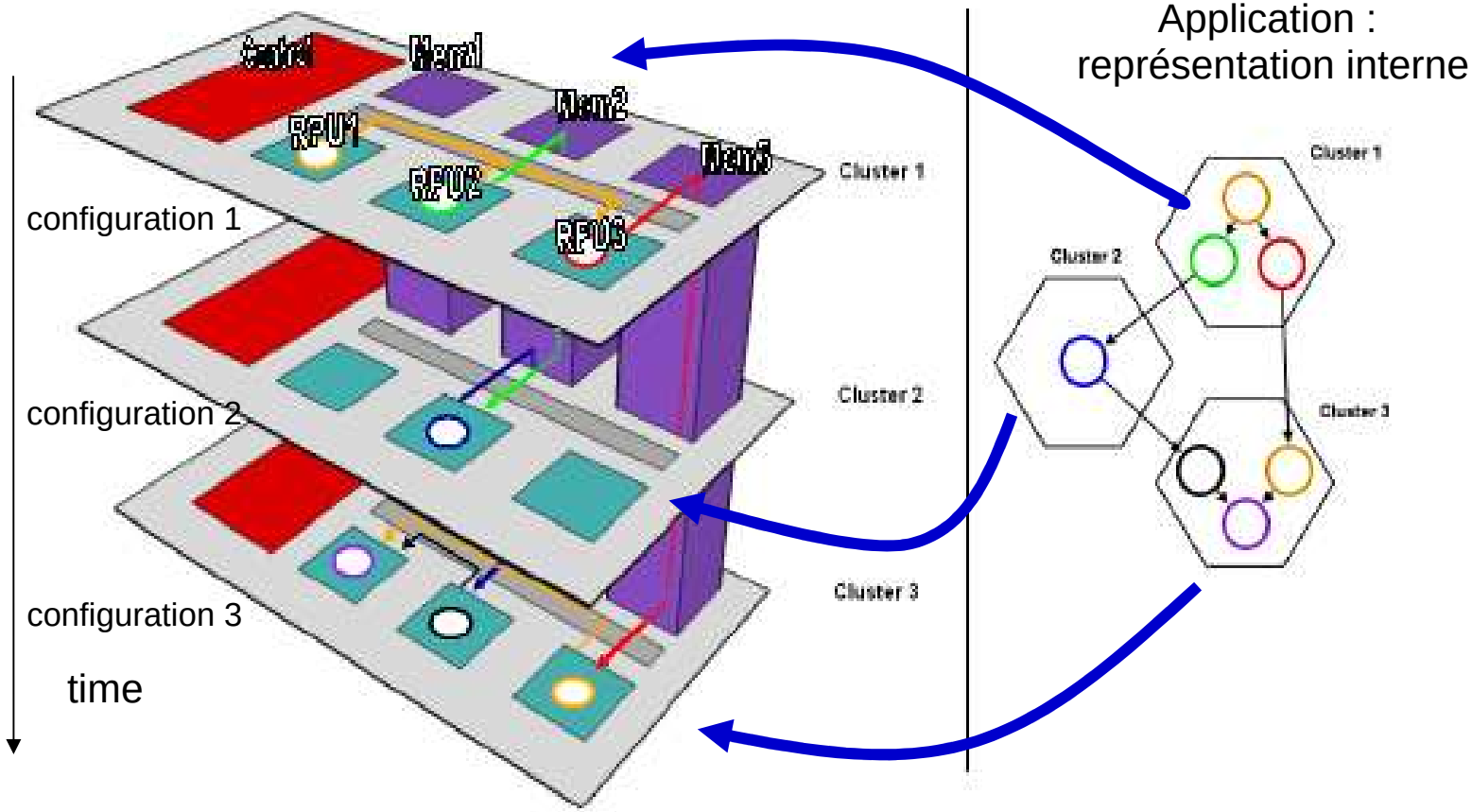
Domain specific algorithm specification



Reconfigurable architecture

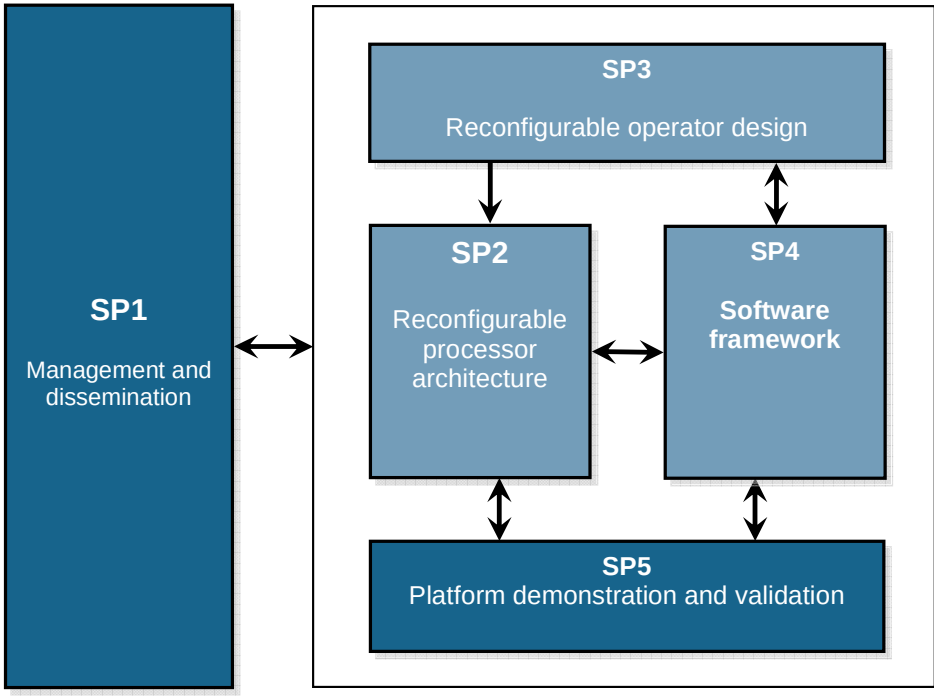


ROMA: principe



Organisation générale du projet

- SP1 : Spécification applications / Management du projet / Dissémination
- SP2 : Architecture du processeur reconfigurable
- SP3 : Architecture des opérateurs reconfigurables
- SP4 : Environnement logiciel
- SP5 : Plate-forme de démonstration, validation



Domaine d'application - spécification algorithmique

Thomson

❑ Domaine cible : vidéo

➢ Algorithme PRIME : **P**ixel **R**ecurs**I**ve **M**otion **E**stimator

Applications principales :

- Amélioration de l'affichage LCD & Plasma
- Conversion de fréquence d'affichage
- Réduction du bruit dans la compression MPEG


Zone de mouvement

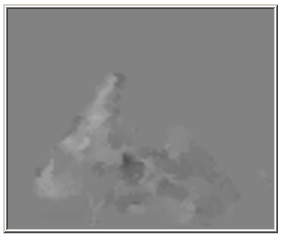

Src YUV à T-1


Src YUV à T


Src Y à T-1


Src Y à T


X


Y

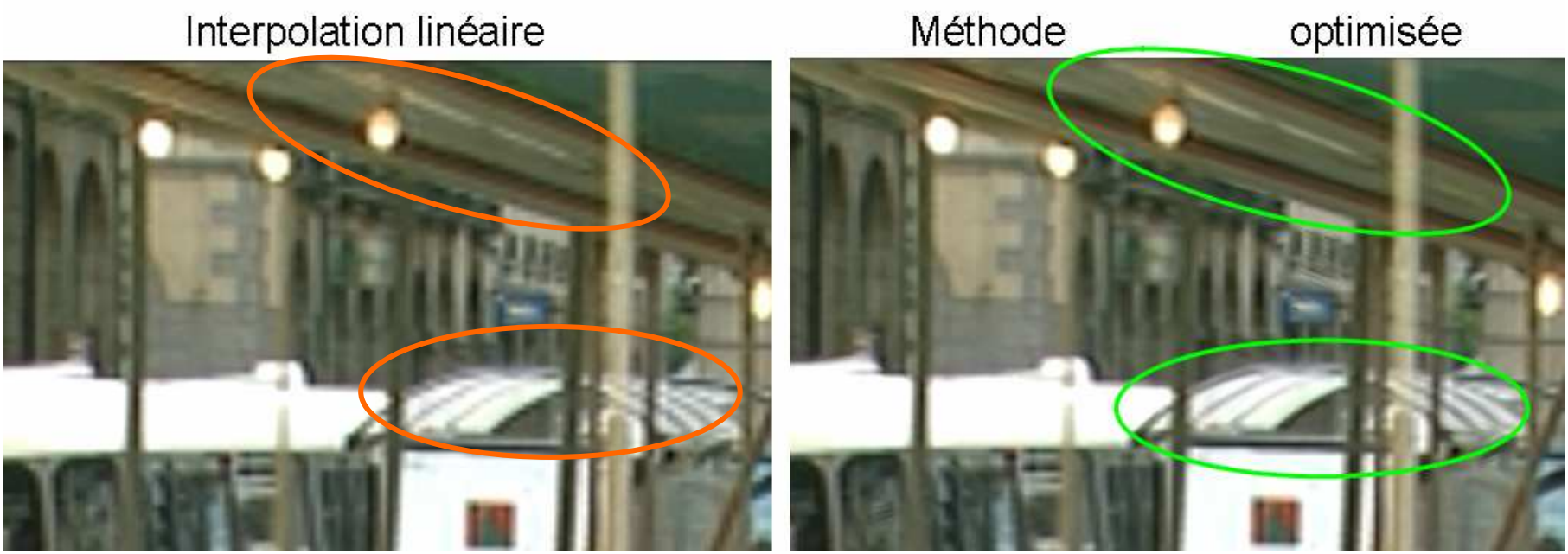
Visualisation des vecteurs de mouvements suivant les 2 axes

Domaine d'application - spécification algorithmique

Thomson

- Algorithme Upscaler SD => HD

Application principale : • Affichage sur écran HD d'un flux vidéo diffusé en SD



Domaine d'application - spécification algorithmique

Thomson

❑ PRIME

- Minimisation de la différence de déplacement entre images (DFD)
- Prédiction spatiale et temporelle
- Correction par algorithme de calcul de gradients

❖ <u>Caractéristique & difficulté</u> :

- ➢ Algorithme récursif
- ➢ Effectue énormément de DFDs

❑ Upscaler SD => HD

- Détection des contours (suivant la luminance)
- Calcul des gradients spécifique au traitement 2D
- Filtrage des distances par médian conditionnel optimisé
- Interpolation 2D

❖ <u>Caractéristique & difficulté</u> :

- ➢ Algorithme adapté à l'embarqué
- ➢ Performances temps réel non atteintes sur accélérateur vidéo

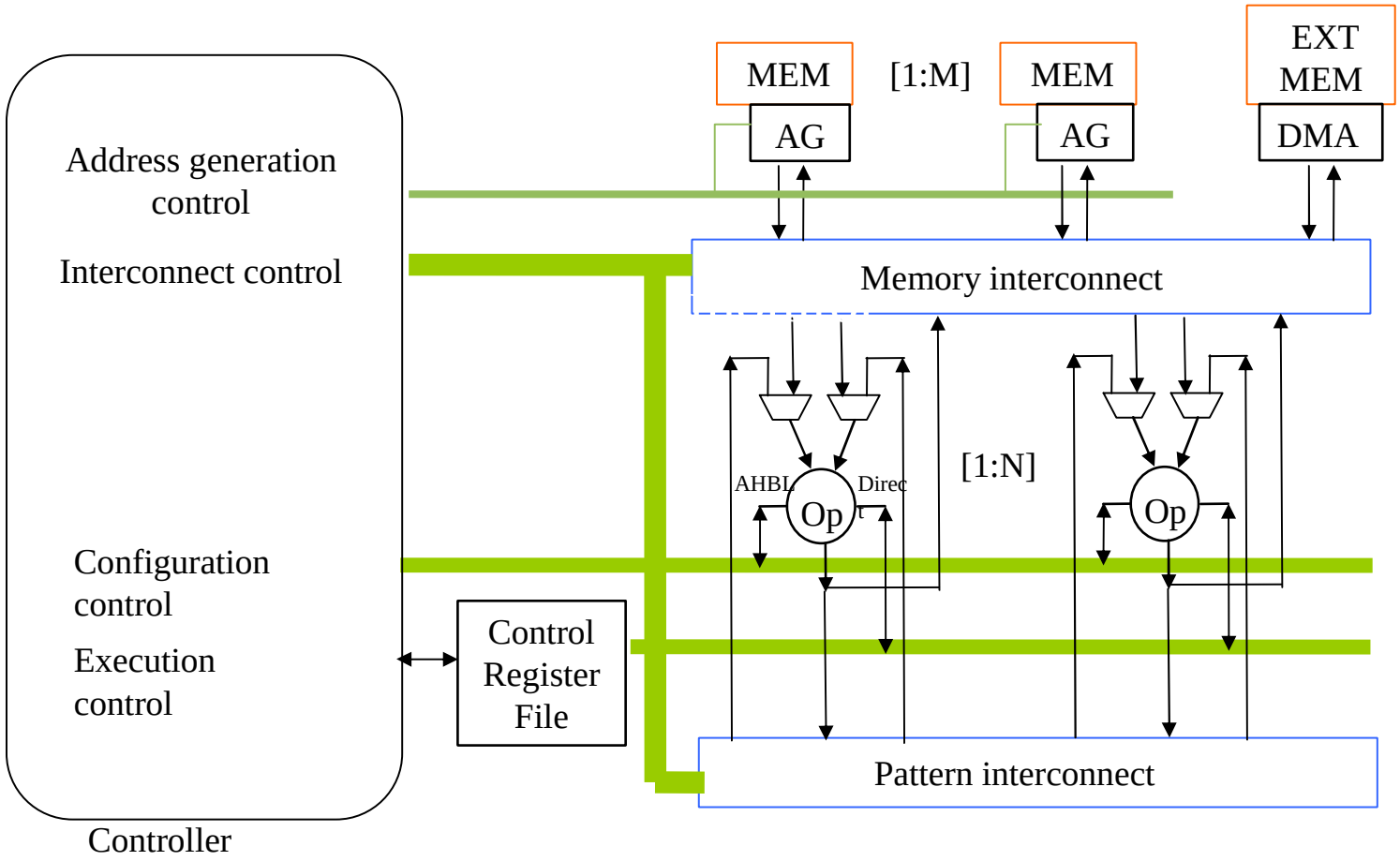
Architecture du processeur reconfigurable

- Caractéristiques de l'architecture du processeur reconfigurable
  - Opérateurs configurables
  - Structure de contrôle adaptée
    - ✓ jeu d'instructions spécifique
    - ✓ gestion rapide des interruptions
  - Définition de la structure mémoire
    - ✓ adaptée pour traitements par flux
  - Réseau de communication optimisé
    - ✓ communication mémoire-opérateur-mémoire
      - **latence permise / bande passante élevée**
    - ✓ communication opérateur-opérateur
      - **pas de latence / bande passante moyenne**

Architecture du processeur reconfigurable

CEA-LIST

❑ Structure du processeur reconfigurable



Architecture du processeur reconfigurable

CEA-LIST

❑ Objectifs atteints

- ➢ Définition de l'architecture
  - ✓ interfaces opérateurs
  - ✓ schémas d'interconnections
    - ▪ **spécialisation des liens inter-opérateurs et des liens vers les mémoires**
    - ▪ **séparation des chemins de données et de configuration**
- ➢ Définition du modèle d'exécution
  - ✓ contrôle centralisé de l'exécution avec couplage fort des opérateurs
  - ✓ reconfiguration dynamique par appel explicite avec prefetch possible
  - ✓ granularité des accès aux données, avec générateurs d'adresse
- ➢ Analyse statique de codes de contrôle
  - ✓ optimisation du jeu d'instructions

Opérateur reconfigurable

LIRMM & IRISA

❑ Systèmes de représentation des nombres (LIRMM)

- Entiers et virgule fixe classiques pour les E/S
  - ✓ non signé
  - ✓ complément à 2
  - ✓ signe/valeur absolue
- Systèmes redondants en interne (*carry-save, borrow-save*)
  - ✓ Temps d'addition indépendant de la taille des nombres
  - ✓ Schéma de propagation des retenues simple à découper (pour la configuration et le SWP)
  - ✓ Nécessite une conversion en sortie
- Futur: bases 4/8, système logarithmique

Opérateur reconfigurable

LIRMM & IRISA

- Algorithmes pour les opérations arithmétiques
  - Addition, soustraction (comparaison, val. absolue...)
    - ✓ Séquentielle, retenue bondissante, sélection de retenue, préfixe parallèle, redondant
    - ✓ Optimisation d'un opérateur pour $\Sigma|a_i \pm b_i|$
  - Multiplication (MAC, carré...)
    - ✓ Recodages
    - ✓ Schéma de décomposition: régulier, Karatsuba, hybride
    - ✓ Arbres de réduction (rapide vs. régulier)
    - ✓ Addition(/accumulation) finale fusionnée ($\Sigma a_i * b_i$)
  - Division et racine carrée
    - ✓ Additions décalages SRT (simple à découper)
    - ✓ Itérations de fonction Newton (réutilise multiplieurs)
  - Points critiques
    - ✓ Gestion du signe (compl. 2) et des cas spéciaux (débordements)
    - ✓ Découpages pour des tailles non puissances de 2

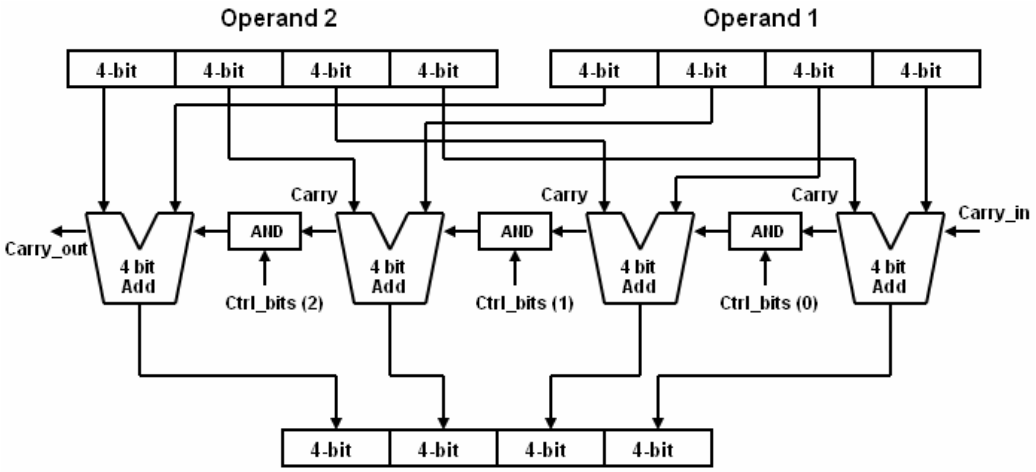
Opérateur reconfigurable

LIRMM & IRISA

❑ Dimensionnement spécifique des opérateurs (IRISA)

- ➢ Objectif : efficacité vitesse/consommation
- ➢ Principe : opérateur de taille spécifique selon la taille des opérandes
- ➢ Moyen : SWP (sub-word parallelism)

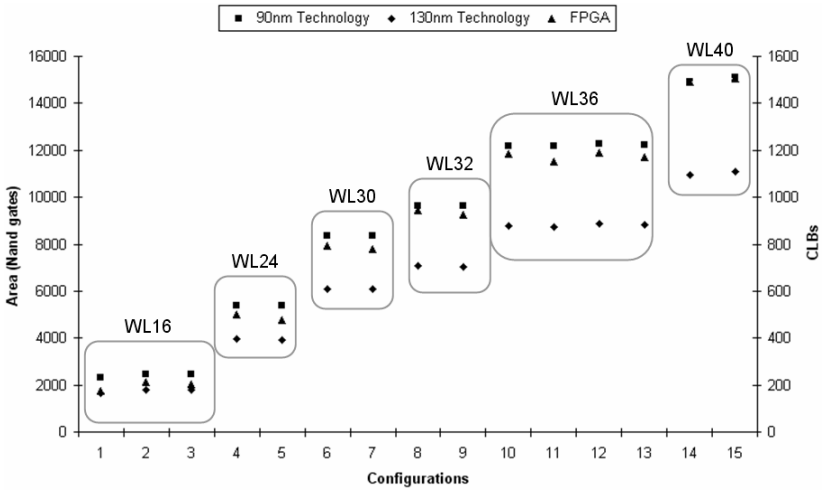
Additionneur 16 bits
ou 4 fois 4 bits
ou 2 fois 8 bits
…

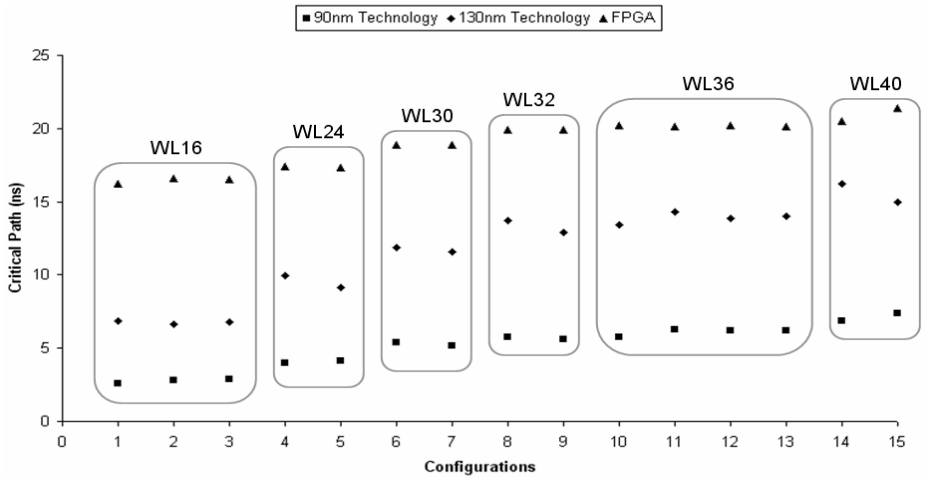


Opérateur reconfigurable

LIRMM & **IRISA**

- Dimensionnement spécifique des opérateurs
  - Objectifs atteints :
    - Etat de l'art des opérateurs SWP de base
      - ✓ +, *, MAC
      - ✓ 4, 8, 16 … (L_{i+1}=2. L_{i+1})
    - Conception d'opérateurs +, *, MAC 40 bits (opérandes SWP typées vidéo 8, 10, 12, 16 bits) VHDL RTL

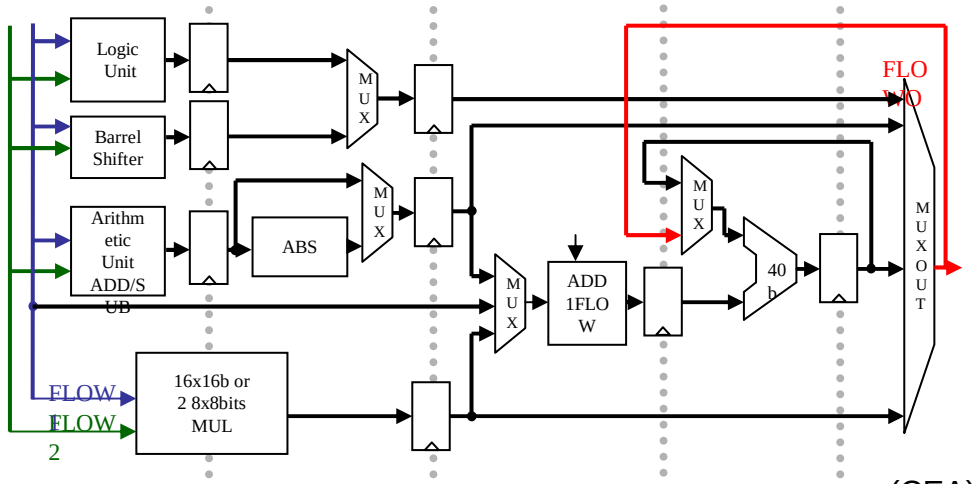




Opérateur reconfigurable

LIRMM & **IRISA**

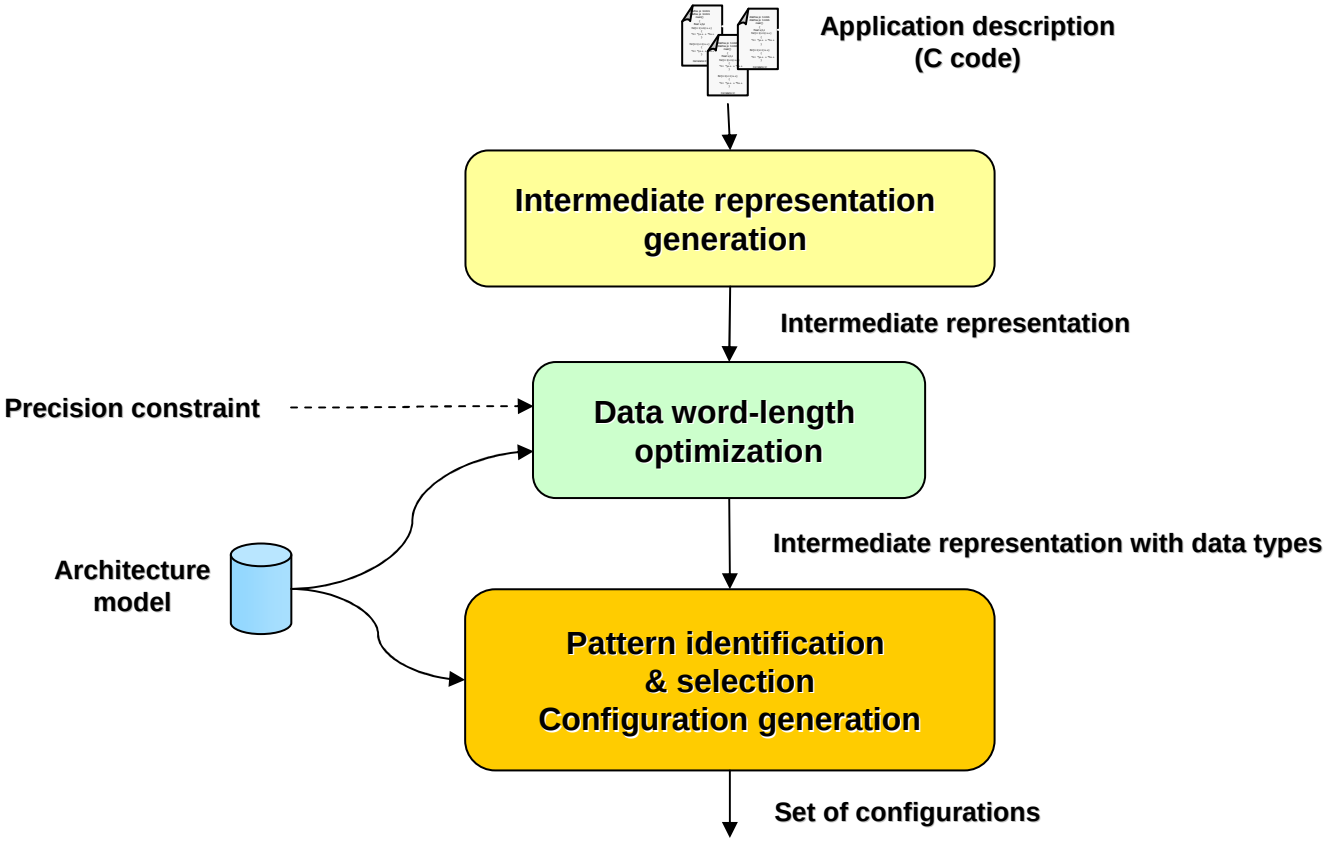
- Dimensionnement spécifique des opérateurs
  - Travaux en cours :
    - Opérateur SWP reconfigurable pour la vidéo
      - ✓ Opérations ciblées (+, *, MAC, ABS, SAD, …)
      - ✓ Architecture
      - ✓ Configurations (opération(s), taille input SWP, taille data internes)



(CEA)

Environnement logiciel

❑ Flot global de compilation



Environnement logiciel

IRISA

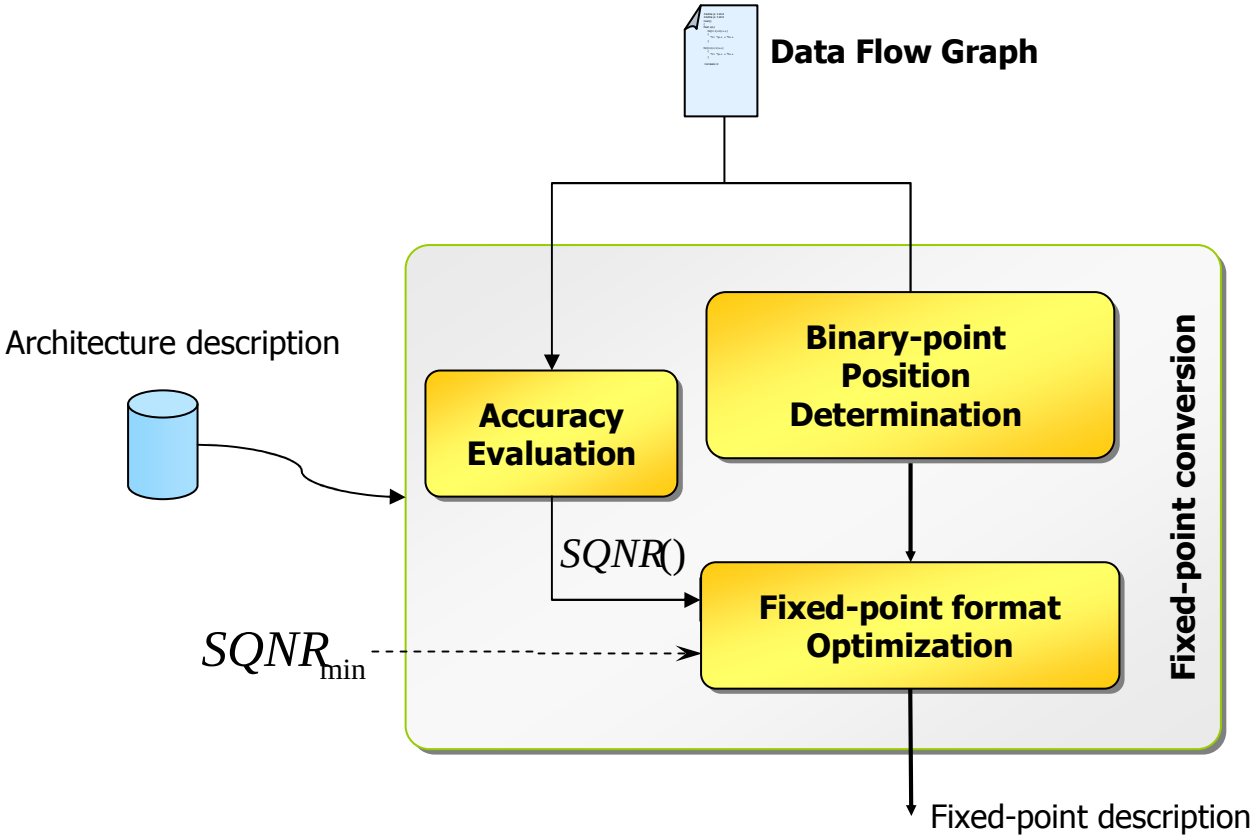
❑ Optimisation de la largeur des données

- ➢ Objectifs
  - ✓ Réaliser la conversion en virgule fixe de l'application
  - ✓ Optimiser la largeur des données sous contrainte de précision
- ➢ Fondements de l'approche
  - ✓ **Méthode analytique** de détermination de la dynamique des données et d'évaluation de la précision
    - ▪ Réduction des temps d'optimisation par rapport aux méthodes basées sur des simulations en virgule fixe

Environnement logiciel

IRISA

❑ Flot de conversion en virgule fixe



Environnement logiciel

IRISA

❑ Flot virgule fixe : Objectifs atteints

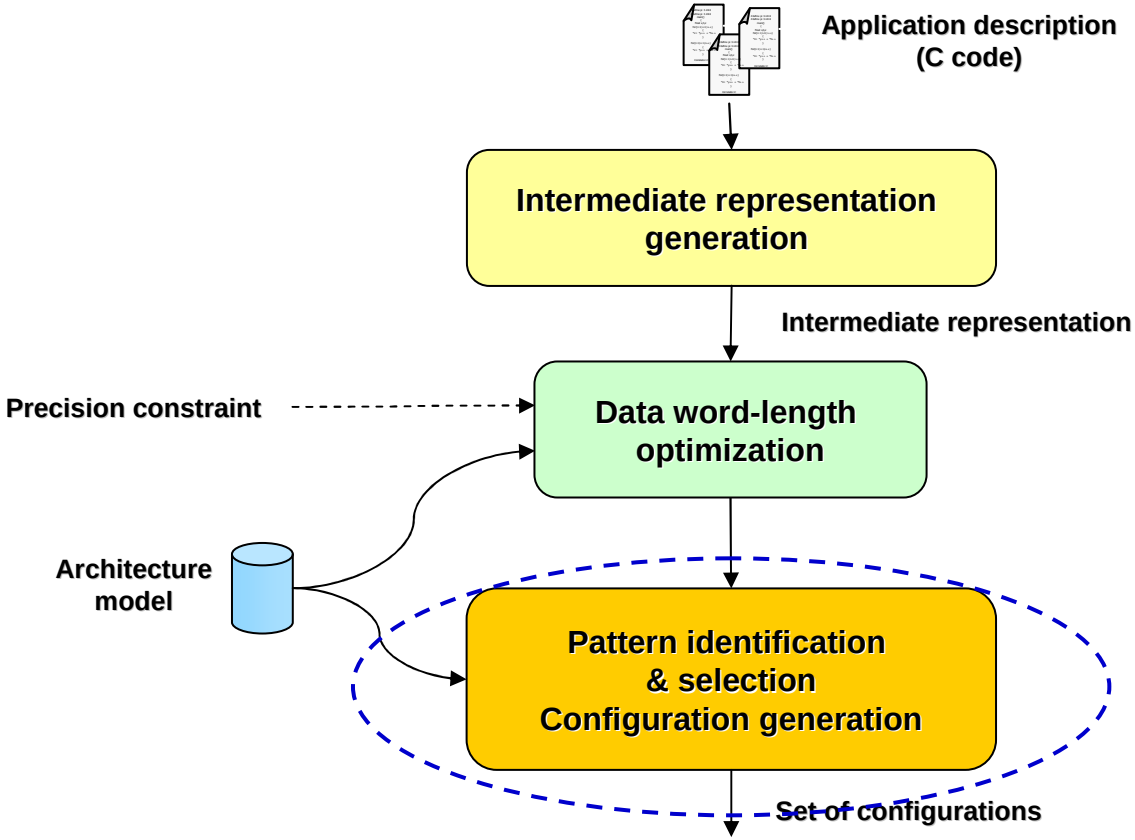
- **Outil d'évaluation de la précision des calculs**
  - ✓ **Modélisation au niveau bruit de calcul**
  - ✓ **Extraction des équations régissant le système**
    - Traitement des structures non récursives
  - ✓ **Génération de l'expression de la puissance du bruit**
  - ✓ **Interfaces entre les 2 modules à réaliser**

- **Outil de conversion en virgule fixe**
  - ✓ **Modélisation du problème d'optimisation de la largeur des données**
  - ✓ **Etat de l'art sur les techniques d'optimisation en cours**

Environnement logiciel

IRISA

- Flot de compilation



Environnement logiciel

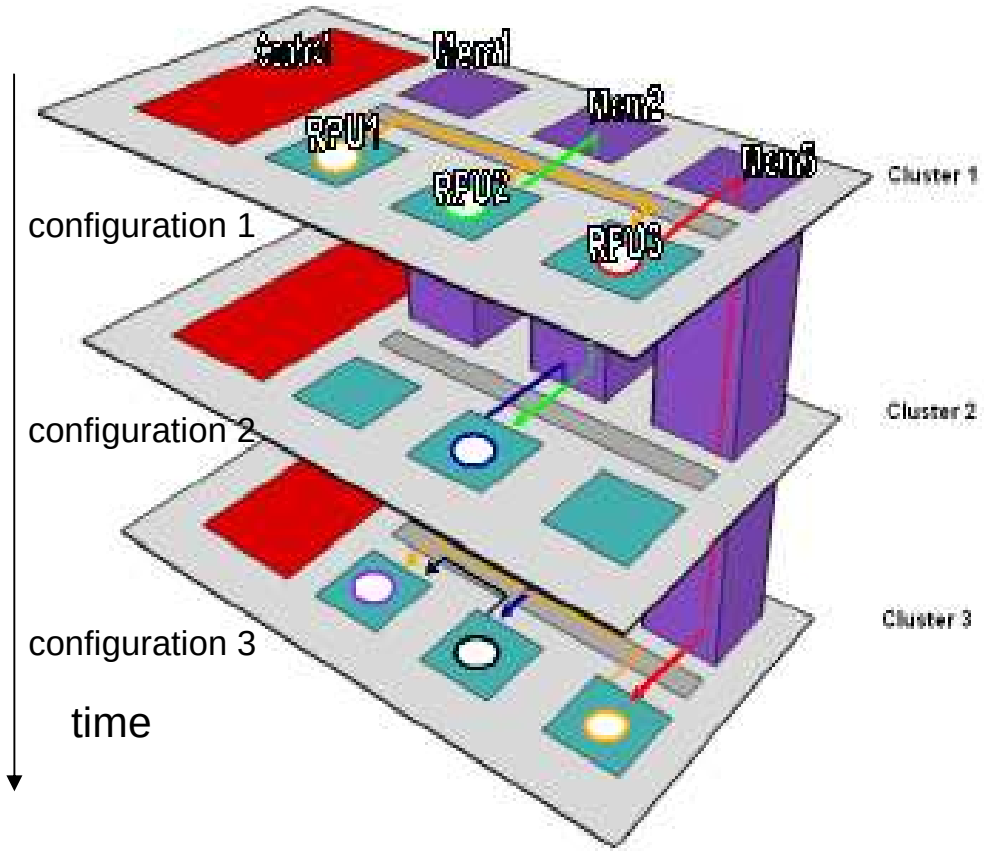
IRISA

❑ Flot de compilation

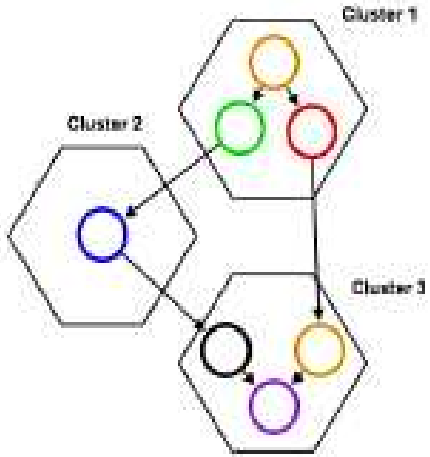
- ➢ Objectifs
  - ✓ Sélectionner les parties critiques de l'application et réaliser les transformations nécessaires à l'optimisation du placement sur l'architecture
  - ✓ Générer les configurations de l'architecture
- ➢ Fondements de l'approche
  - ✓ Représentation intermédiaire basée sur les **graphes hiérarchiques aux dépendances conditionnées** (HCDG)
  - ✓ **Programmation par contraintes** pour les problèmes d'optimisation

Environnement logiciel

- Flot de compilation : séquencement de configurations IRISA

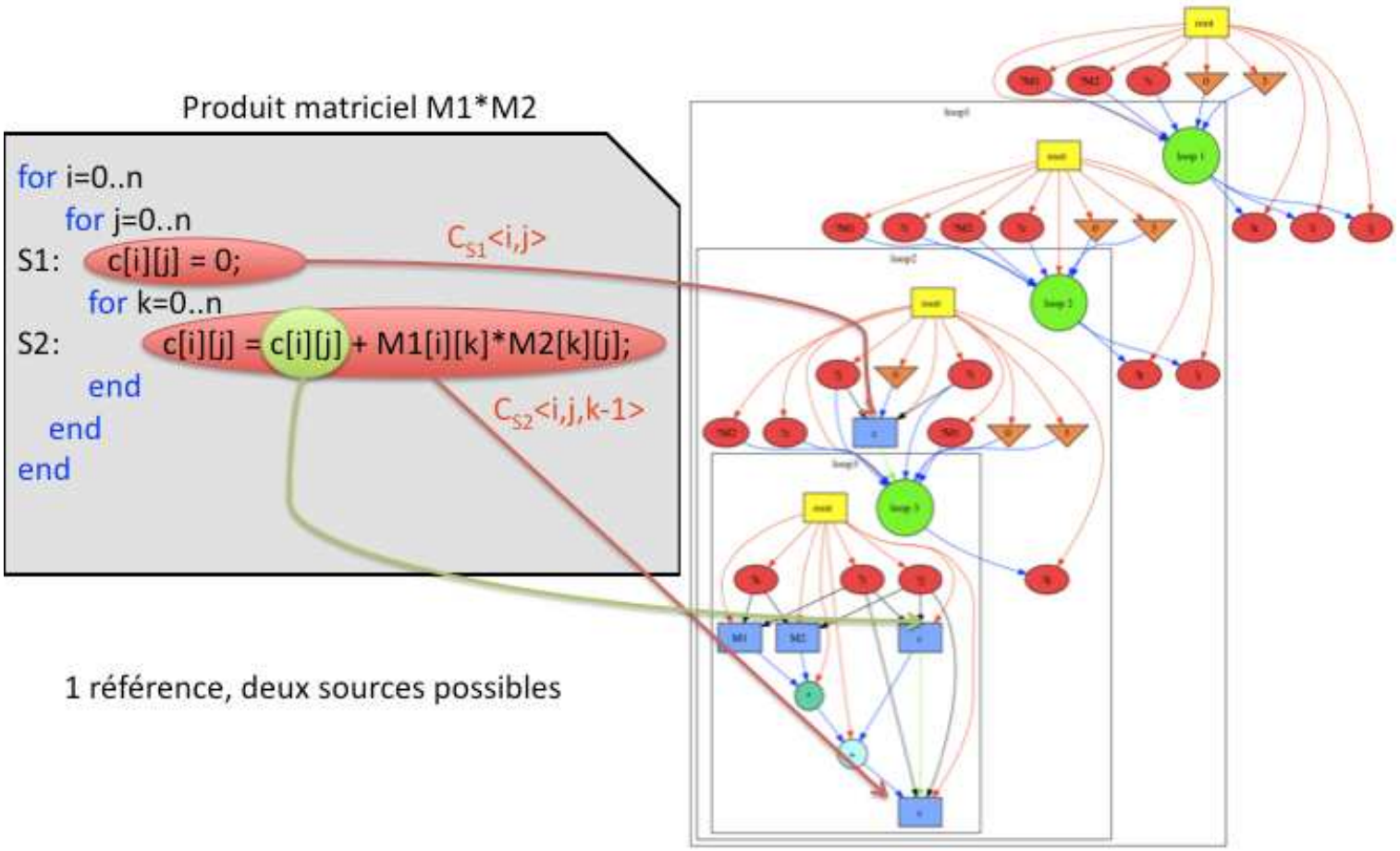


Application :
représentation interne



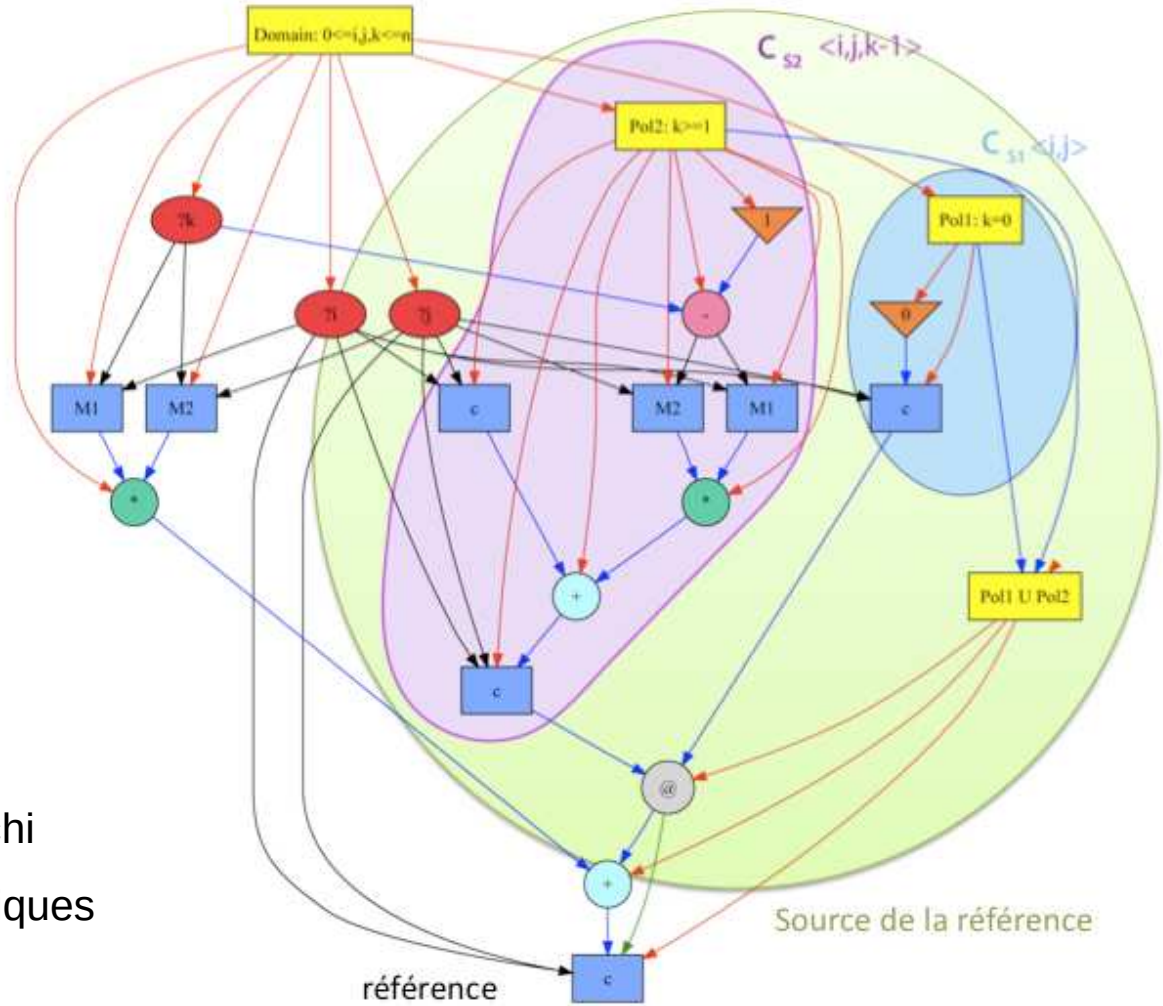
Environnement logiciel

- Flot de compilation : représentation intermédiaire HCDG IRISA



Environnement logiciel

- Flot de compilation : représentation intermédiaire HCDG IRISA



HCGD enrichi

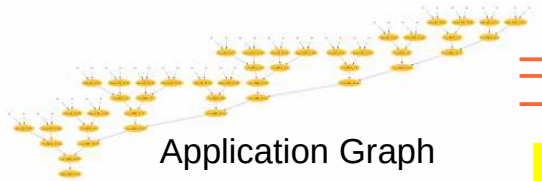
gardes polyédriques

Environnement logiciel

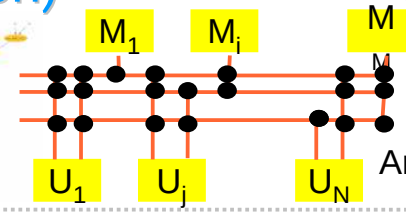
- Flot de compilation : prototype (démonstration)

IRISA

Application Graph + Architecture

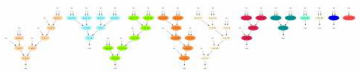


Application Graph



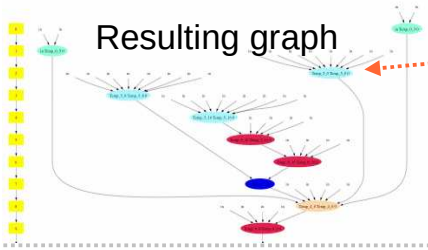
Architecture

Pattern Identification

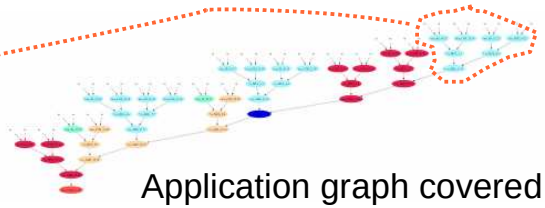


Set of patterns

Pattern Selection :
- scheduling,
- non-convex match removing

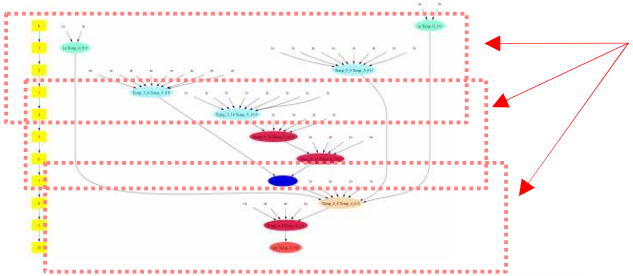


Resulting graph



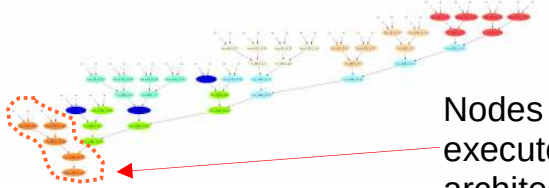
Application graph covered by the selected patterns.

Configuration generation
- match expansion
- placement,
- routing



Sliding window

Set of Configurations



Resulting application graph

Nodes with the same color are executed by the reconfigurable architecture during the same execution cycle.

Environnement logiciel

❑ Flot de compilation

- ➢ Objectifs atteints :
  - ✓ Représentation intermédiaire
    - **Modèle de représentation des boucles dans les graphes HCDG**
    - **Intégration dans l'environnement de compilation Gecos développé à l'Irisa**
  - ✓ Définition d'un flot préliminaire
    - **Sélection de motifs, couverture du graphe, génération de configurations**
    - **Réalisation d'un premier prototype**

Plateforme de démonstration

Thomson & CEA

- Mise en place
  - Préparation du code
    - ✓ Détourage des fonctions critiques
    - ✓ Extraction des motifs de test : entrées et sorties de références
  - Plateforme d'émulation matérielle
    - ✓ Mise en place des interfaces
    - ✓ Portage du processeur ROMA sur FPGA
- Scénario de démonstration
  - Profiling de l'applicatif complet retenu sur PC
  - Compilation des fonctions critiques utilisant le flot de conception ROMA
  - Chargement sur le processeur ROMA
  - Exécution
  - Comparaison des exécutions

Plateforme de démonstration

Thomson & CEA