

# Méthode de conception rapide d'architecture massivement parallèle sur puce: de la modélisation à l'expérimentation sur FPGA

**Mouna Baklouti Kammoun**

18 Décembre 2010



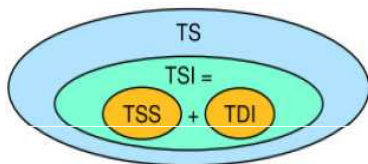
# Plan

- 1 Introduction
- 2 Système mppSoC proposé
- 3 Méthode de conception/programmation de mppSoC
- 4 Flot de génération de configurations mppSoC
- 5 Étude de cas
- 6 Conclusions et perspectives

# Plan

- 1 Introduction
  - Domaine d'application
  - Problématique et positionnement
  - Contributions
- 2 Système mppSoC proposé
- 3 Méthode de conception/programmation de mppSoC
- 4 Flot de génération de configurations mppSoC
- 5 Étude de cas
- 6 Conclusions et perspectives

# Traitement du signal systématique (TSS)



TS : Traitement de Signal  
TSI : Traitement de Signal Intensif  
TSS : Traitement de Signal Systématique  
TDI : Traitement de Données Intensif

- Caractéristiques du TSS :
  - traitements très réguliers
  - indépendants de la valeur des données



# Traitement du signal systématique (TSS)

- Présence du TSS dans :
  - systèmes de détection
  - multimédia
  - télécommunications
  - **Puissance de calcul** ↗
- Systèmes embarqués
  - SoC
    - ASIC
    - FPGA
  - **Capacité d'intégration** ↗



# Traitement du signal systématique (TSS)

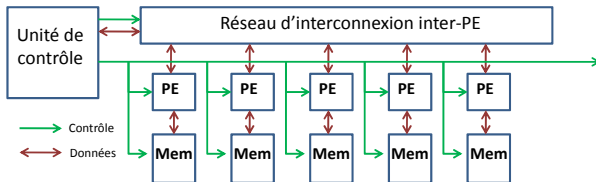
- Présence du TSS dans :
  - systèmes de détection
  - multimédia
  - télécommunications
  - **Puissance de calcul** ↗
- Systèmes embarqués
  - SoC
    - ASIC
    - FPGA
  - **Capacité d'intégration** ↗



- Architectures massivement parallèles sur puce
  - hautes performances
  - multiplicité des ressources de calcul
    - Flynn : SIMD, MIMD...

# Efficacité de SIMD pour le TSS

- Modèle SIMD

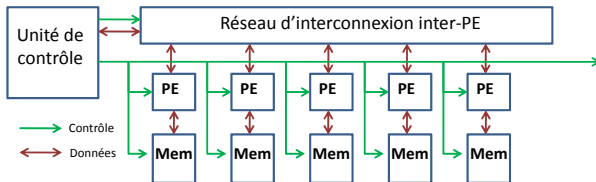


- Avantages :

- parallélisme de données
- synchronisme parfait
- intégration de simples unités d'exécution
- consommation réduite pour les traitements réguliers

# Efficacité de SIMD pour le TSS

- Modèle SIMD



- Avantages :

- parallélisme de données
- synchronisme parfait
- intégration de simples unités d'exécution
- consommation réduite pour les traitements réguliers

⇒ SIMD : adapté au TSS



# Problématique

- **Déclin** des architectures SIMD traditionnelles
  - MasPar, ILLIAC IV, Connection Machine...
  - complexité technologique
  - coût de conception

# Problématique

- **Déclin** des architectures SIMD traditionnelles
- Solutions SIMD existantes sur FPGA
  - Accélérations SIMD matérielles
    - R.L. Rosas et al. 2005 : SIMD pour la détection de contours
    - M. Sayed et al. 2008 : VBSME pour le calcul de l'estimation de mouvement
    - **coûts et délais de conception** 

# Problématique

- **Déclin** des architectures SIMD traditionnelles
- Solutions SIMD existantes sur FPGA
  - Accélérations SIMD matérielles
  - Extensions SIMD pour processeurs embarqués
    - spécifiques à un traitement
    - **non génériques**

# Problématique

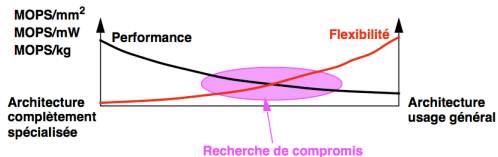
- **Déclin** des architectures SIMD traditionnelles
- Solutions SIMD existantes sur FPGA
  - Accélérations SIMD matérielles
  - Extensions SIMD pour processeurs embarqués
  - Solutions SIMD programmables
    - F. Schurz and D. Fey. 2007 : architecture SIMD à base de l'IP PicoBlaze avec des PE 1bit très réduits
    - Ph. Bonnot et al. 2008 : Ter@Core avec une conception du PE à base des blocs DSP de l'FPGA utilisé + un seul mode de communication en anneau
    - ...

# Problématique

- **Déclin** des architectures SIMD traditionnelles
- Solutions SIMD existantes sur FPGA
  - Accélérations SIMD matérielles
  - Extensions SIMD pour processeurs embarqués
  - Solutions SIMD programmables
- **Limites** des solutions existantes :
  - conçues spécifiquement pour une application déterminée : coûts de conception élevés
  - peu évolutives : difficulté de satisfaire différents besoins

# Problématique

- **Déclin** des architectures SIMD traditionnelles
- Solutions SIMD existantes sur FPGA
  - Accélérations SIMD matérielles
  - Extensions SIMD pour processeurs embarqués
  - Solutions SIMD programmables
- Défis de conception
  - technologiques
  - délais (commercial)



# Objectifs

Rendre le système SIMD flexible et pouvant s'adapter aux besoins applicatifs

- choix d'une solution architecturale adéquate à l'application
- souplesse de l'architecture
- satisfaire les besoins applicatifs

# Objectifs

## Rendre le système SIMD flexible et pouvant s'adapter aux besoins applicatifs

- choix d'une solution architecturale adéquate à l'application
- souplesse de l'architecture
- satisfaire les besoins applicatifs

## Proposer une méthode de conception simple et rapide

- accélérer le temps de conception
- réduire le temps de mise sur le marché



# Contributions

Définir un système massivement parallèle sur puce à architecture SIMD pouvant être adapté à l'application

- paramétrique
- programmable

Proposer une méthode de conception par assemblage d'IP

- réduire le temps de conception

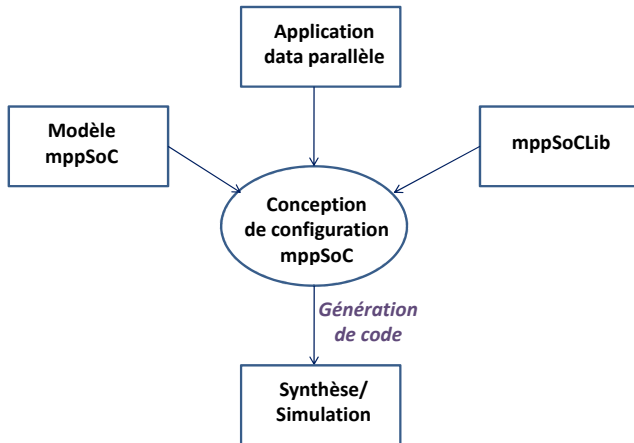
Accélérer la conception par génération automatique de configurations SIMD

- abstraire les détails d'implémentation

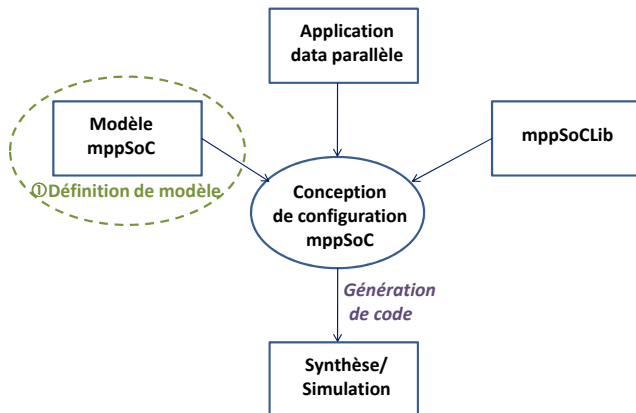
Réaliser une expérimentation sur FPGA

- tester différentes configurations mppSoC
- choisir la configuration la plus adéquate

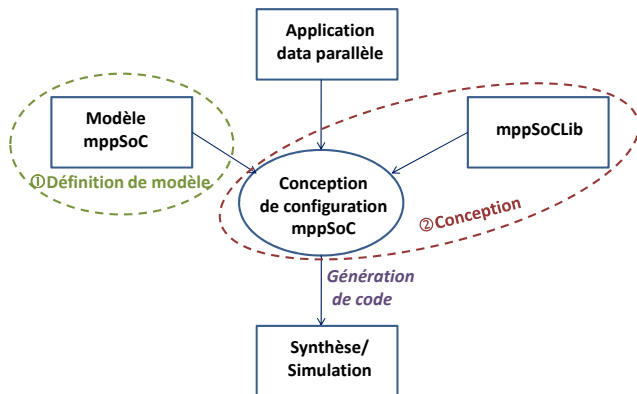
# Approche proposée



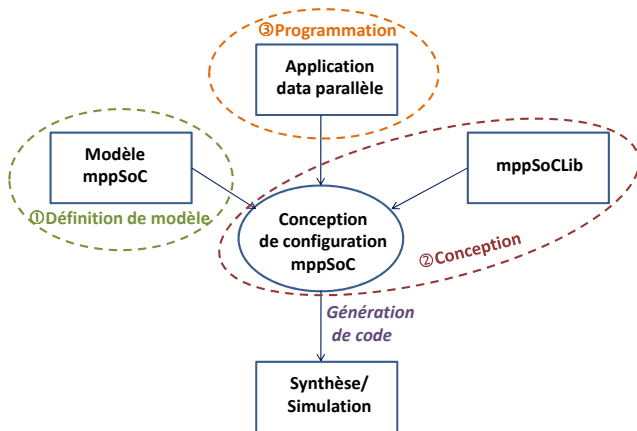
# Approche proposée



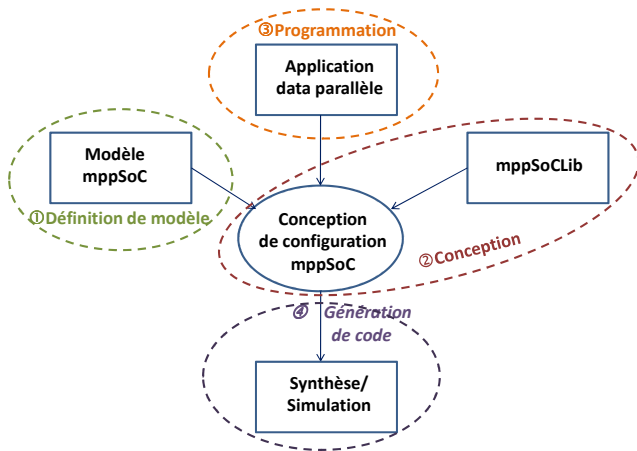
# Approche proposée



# Approche proposée



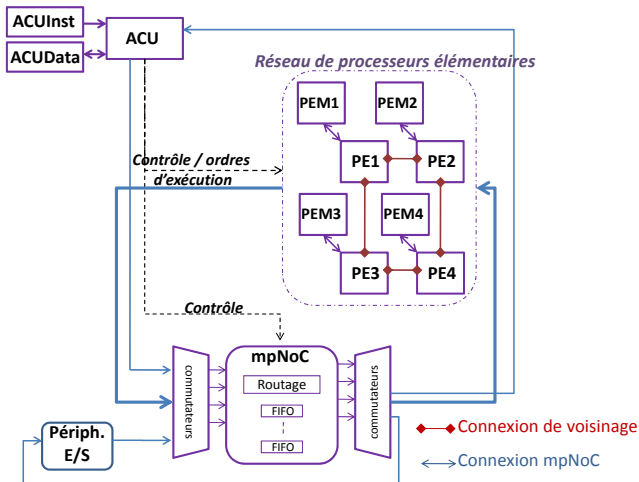
# Approche proposée



# Plan

- 1 Introduction
- 2 **Système mppSoC proposé**
  - Modèle mppSoC proposé
  - Caractéristiques du modèle mppSoC
- 3 Méthode de conception/programmation de mppSoC
- 4 Flot de génération de configurations mppSoC
- 5 Étude de cas
- 6 Conclusions et perspectives

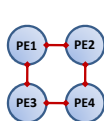
# Modèle mppSoC proposé



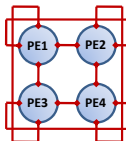


# Caractéristiques du modèle mppSoC

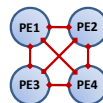
- Paramétrique et flexible
  - nombre de PEs variable : assurer l'extensibilité
  - taille mémoire variable : répondre aux besoins applicatifs
  - support de différents modèles d'interconnexions de voisinage : répondre à la variété des communications inter-PE



*Maillage*



*Tore*



*Xnet*



*Linéaire*



*Anneau*

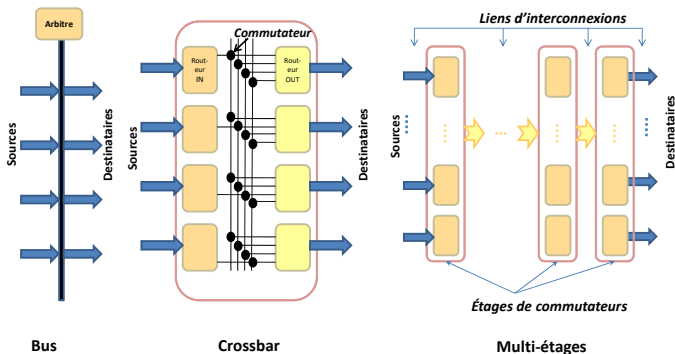
- mpNoC : réseau de communication point à point

# Caractéristiques du modèle mppSoC

- Paramétrique et flexible
  - nombre de PEs variable : assurer l'extensibilité
  - taille mémoire variable : répondre aux besoins applicatifs
  - support de différents modèles d'interconnexions de voisinage : répondre à la variété des communications inter-PE
  - mpNoC : réseau de communication point à point
    - connecter tout PE avec un autre
    - connecter les PEs aux périphériques
    - connecter l'ACU avec les PEs

# Caractéristiques du modèle mppSoC

- Paramétrique et flexible
  - nombre de PEs variable : assurer l'extensibilité
  - taille mémoire variable : répondre aux besoins applicatifs
  - support de différents modèles d'interconnexions de voisinage : répondre à la variété des communications inter-PE
  - mpNoC : réseau de communication point à point



# Caractéristiques du modèle mppSoC

- Paramétrique et flexible
  - nombre de PEs variable : assurer l'extensibilité
  - taille mémoire variable : répondre aux besoins applicatifs
  - support de différents modèles d'interconnexions de voisinage : répondre à la variété des communications inter-PE
  - mpNoC : réseau de communication point à point

## Avantages du réseau mpNoC proposé

- Divers modes de communication
  - tous-vers-un
  - un-vers-tous
  - un-vers-un
- Entrées/Sorties parallèles

# Caractéristiques du modèle mppSoC

- Paramétrique et flexible
- Modulaire
  - assembler les composants nécessaires
  - choisir la configuration optimale
  - maîtriser la complexité
  - réduire le temps de conception

# Caractéristiques du modèle mppSoC

- Paramétrique et flexible
- Modulaire
- Programmable
  - s'adapter aux évolutions applicatives
  - langage data parallèle ⇒ adapté au modèle d'exécution SIMD

# Caractéristiques du modèle mppSoC

- Paramétrique et flexible
- Modulaire
- Programmable

## Mise en place de configurations (instances) mppSoC

- définir les paramètres
- choisir les composants nécessaires
- adapter selon les besoins

- M. Baklouti et al. *IP based configurable SIMD massively parallel SoC* . *FPL, PhD Forum*. Italy. 2010

# Plan

- 1 Introduction
- 2 Système mppSoC proposé
- 3 Méthode de conception/programmation de mppSoC**
  - Conception de mppSoC
  - mppSoCLib
  - Programmation de mppSoC
- 4 Flot de génération de configurations mppSoC
- 5 Étude de cas
- 6 Conclusions et perspectives

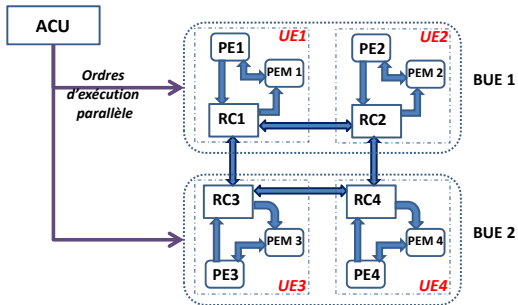


# Principes de conception

- Assemblage d'IPs
  - processeurs, mémoires, réseaux...
  - méthodologie de réutilisation

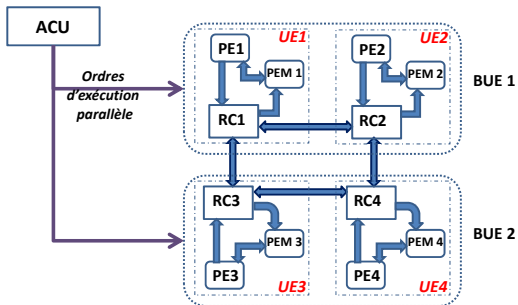
# Principes de conception

- Assemblage d'IPs
- Conception hiérarchique



# Principes de conception

- Assemblage d'IPs
- Conception hiérarchique



- **Avantages**
  - Alléger les coûts de conception
  - Permettre une implémentation générique 2D/1D
  - Faciliter le placement du réseau de PEs sur la puce

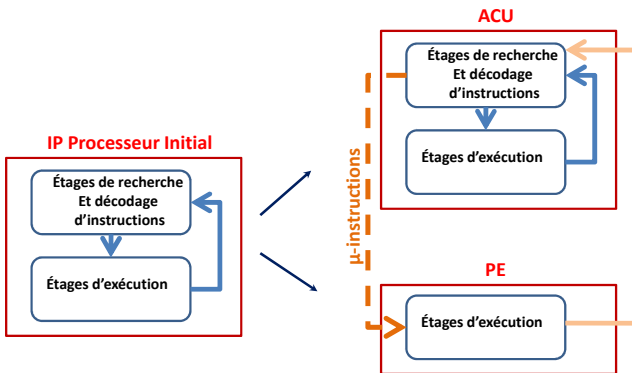
# Principes de conception

- Assemblage d'IPs
- Conception hiérarchique
- Conception des processeurs
  - ACU + PE : même IP processeur
    - minimiser les développements matériels et logiciels
    - réduire le haut coût de conception lié aux machines SIMD

# Méthodes de conception des processeurs

## Réduction

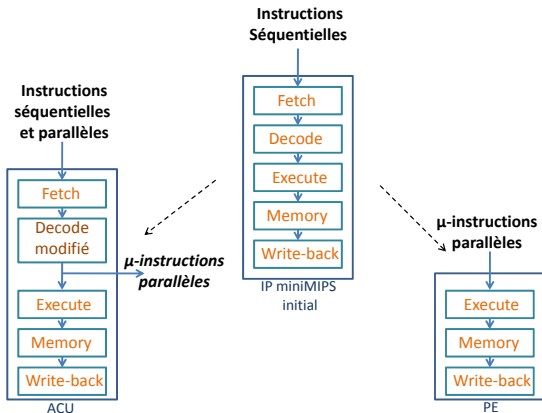
- utiliser un IP open-source
- réduire le processeur  $\Rightarrow$  PE simple et réduit



# Méthodes de conception des processeurs

## Réduction

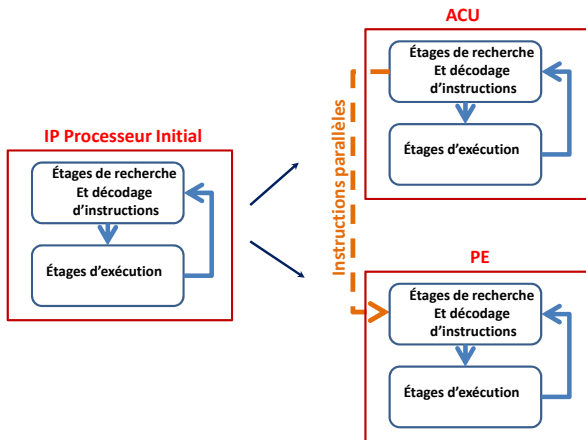
- utiliser un IP open-source
- réduire le processeur  $\Rightarrow$  PE simple et réduit



# Méthodes de conception des processeurs

## Réplication

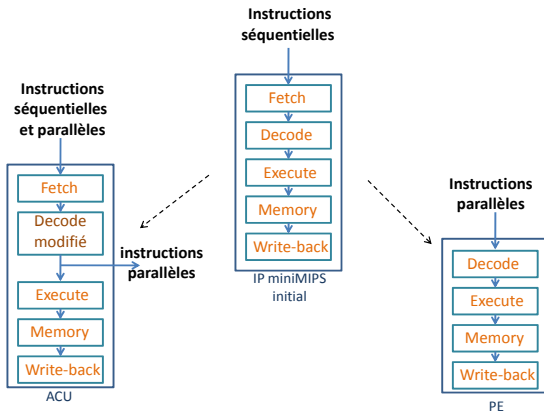
- concevoir le PE par le même IP processeur
- répliquer le PE



# Méthodes de conception des processeurs

## Réplication

- concevoir le PE par le même IP processeur
- répliquer le PE





# Méthodes de conception des processeurs

## Réduction/Réplication

FPGA	Conception (miniMIPS)	Nbre max de PE	taille mém/PE
Stratix 2S180	réduction	96	0.5 Ko
	réplication	64	0.5 Ko

# Méthodes de conception des processeurs

## Réduction/Réplication

	réduction	réplication
Temps de conception	long	réduit
Intégration de PE	+++	+
Efficacité performance/surface	++	+

# Méthodes de conception des processeurs

## Réduction/Réplication

	réduction	réplication
Temps de conception	long	réduit
Intégration de PE	+++	+
Efficacité performance/surface	++	+

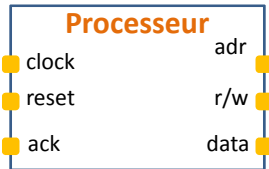
## Solution pour accélérer la conception des processeurs

- Utiliser un IP processeur
  - réduire le temps de conception
  - faciliter la programmation
- Choisir entre deux méthodes de conception : réduction/réplication
  - satisfaire les besoins

- M. Baklouti et al. *A design and an implementation of a parallel based SIMD architecture for SoC on FPGA*. **DASIP**. France. 2008

# Bibliothèque : mppSoCLib

- Ensemble d'IPs dédiés à mppSoC
- Interface propriétaire
  - Processeurs
    - miniMIPS : réduction + réplication
    - OpenRisc 1200 : réduction + réplication
    - NIOS II : réplication

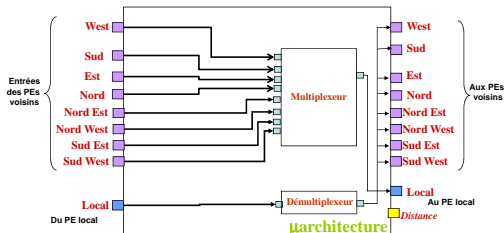


# Bibliothèque : mppSoCLib

- Ensemble d'IPs dédiés à mppSoC
- Interface propriétaire
  - Processeurs
  - Mémoires

# Bibliothèque : mppSoCLib

- Ensemble d'IPs dédiés à mppSoC
- Interface propriétaire
  - Processeurs
  - Mémoires
  - Routeur de voisinage

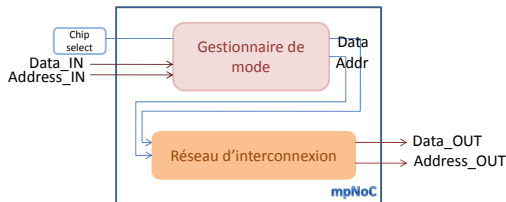


⇒ simplicité du routeur

- M. Baklouti et al. *Study and Integration of a Parametric Neighbouring Interconnection Network in a Massively Parallel Architecture on FPGA*. AICCSA. Morocco. 2009

# Bibliothèque : mppSoCLib

- Ensemble d'IPs dédiés à mppSoC
- Interface propriétaire
  - Processeurs
  - Mémoires
  - Routeur de voisinage
  - Réseau mpNoC



- mode configurable : PE-PE, PE-ACU, PE-périphérique
- IP réseau d'interconnexion paramétrique

- M. Baklouti et al. *Reconfigurable Communication Networks in a Parametric SIMD Parallel System on Chip*. **International Symposium on Applied reconfigurable Computing**. Thailand. 2010
- M. Baklouti et al. *Scalable mpNoC for Massively Parallel Systems - Design and Implementation on FPGA*. **Journal of Systems Architecture**. 2010

# Programmation de mppSoC

- dépend du processeur utilisé : Assembleur/C
- dépend de la méthodologie de conception des processeurs

## Méthode de programmation proposée

- jeu d'instructions dérivé de celui du processeur utilisé
- macros définies pour les instructions spécifiques
  - Instructions de contrôle
  - Instructions de communications
  - codées à base des instructions d'accès mémoire : *Load/Store*



# Programmation de mppSoC

- dépend du processeur utilisé : Assembleur/C
- dépend de la méthodologie de conception des processeurs

## Méthode de programmation proposée

- jeu d'instructions dérivé de celui du processeur utilisé
- macros définies pour les instructions spécifiques
  - Instructions de contrôle
  - Instructions de communications
  - codées à base des instructions d'accès mémoire : *Load/Store*

TABLE: Codage du macro de lecture d'identité

Macro	Définition	Codage		
		miniMIPS	OpenRisc	NIOS II
<b>P_GET_IDENT (reg)</b>	lire l'identité	p_lui r1,0x2 p_ori r1,r1,0 p_LW reg,0(r1)	l.movhi r1,0x2 l.lwz reg,0x0(r1)	NIOS2_READ_ CPUID(id)

# Programmation de mppSoC

- dépend du processeur utilisé : Assembleur/C
- dépend de la méthodologie de conception des processeurs

## Méthode de programmation proposée

- jeu d'instructions dérivé de celui du processeur utilisé
- macros définies pour les instructions spécifiques

## Conception/programmation proposée pour mppSoC

- Utiliser l'existant
- Accélérer la conception
- Faciliter la programmation

# Plan

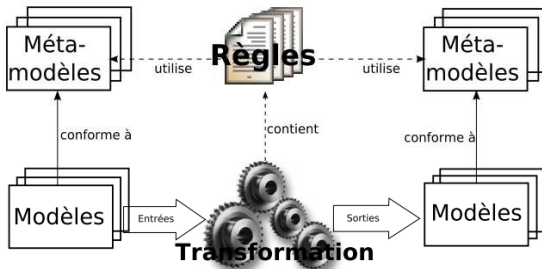
- 1 Introduction
- 2 Système mppSoC proposé
- 3 Méthode de conception/programmation de mppSoC
- 4 Flot de génération de configurations mppSoC
  - IDM
  - Gaspard
  - Flot de conception
- 5 Étude de cas
- 6 Conclusions et perspectives

# Ingénierie Dirigée par les Modèles (IDM)

- Apports de l'IDM
  - répondre à la complexité croissante des systèmes
  - abstraire les détails et techniques d'implémentation
  - réduire le temps de conception

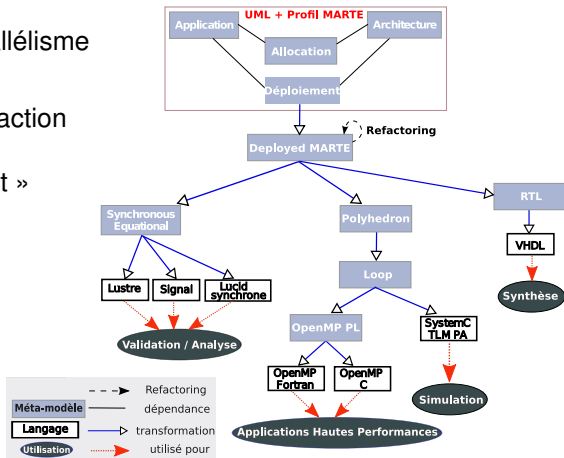
# Ingénierie Dirigée par les Modèles (IDM)

- Apports de l'IDM
- Concepts de l'IDM
  - modèle : abstraction de la réalité
  - méta-modèle : ensemble de concepts et de relations permettant de spécifier des modèles
  - transformation de modèles : passage d'un modèle source décrit à un niveau d'abstraction à un autre modèle destination décrit à un autre niveau d'abstraction

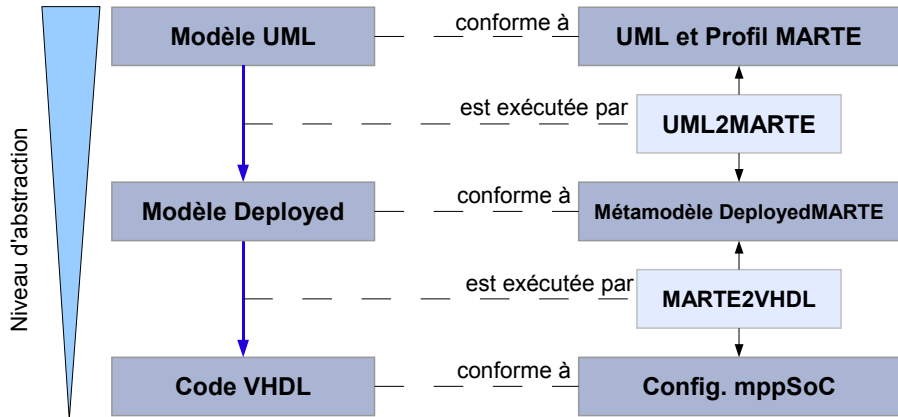


# Gaspard : environnement basé sur l'IDM

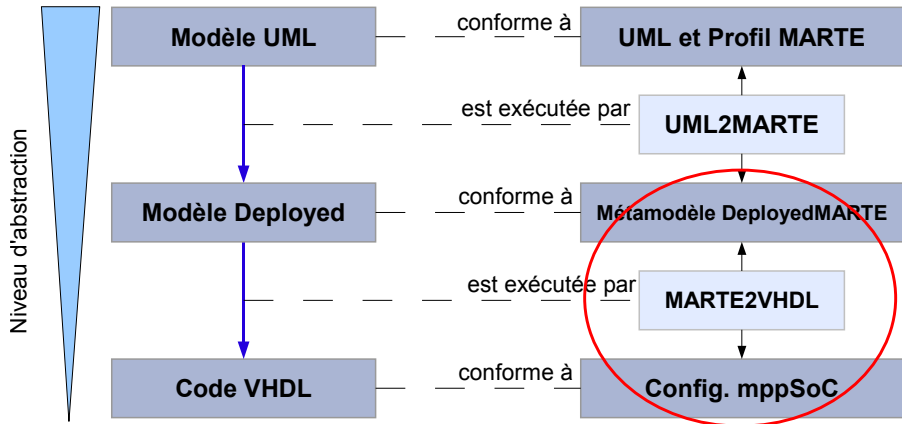
- Environnement de co-conception pour SoC
  - TSS
  - Expression du parallélisme
- Spécification
  - haut niveau d'abstraction
  - UML
  - « MARTE compliant »
  - standard
- Cibles
  - simulation
  - exécution
  - vérification
  - ...



# Flot de conception pour la génération de configurations mppSoC

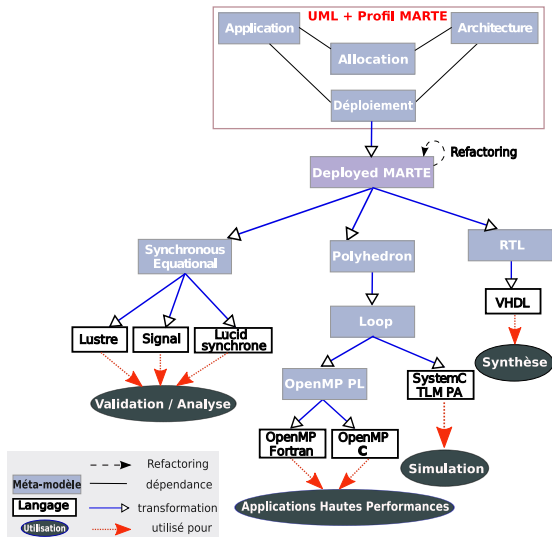


# Flot de conception pour la génération de configurations mppSoC

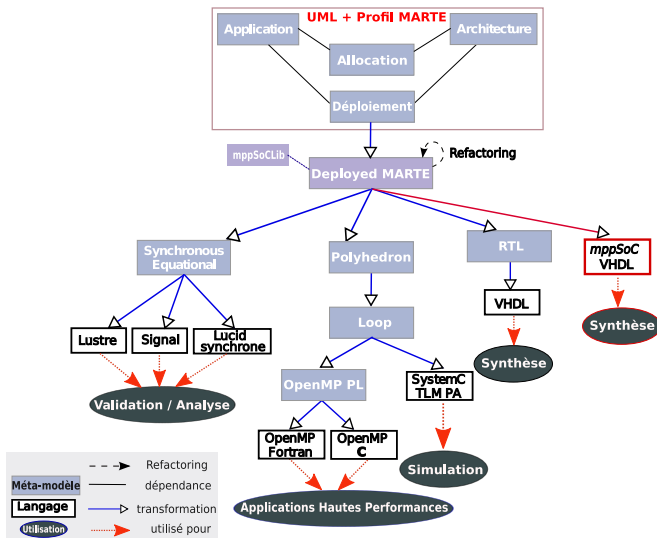




# Intégration dans Gaspard



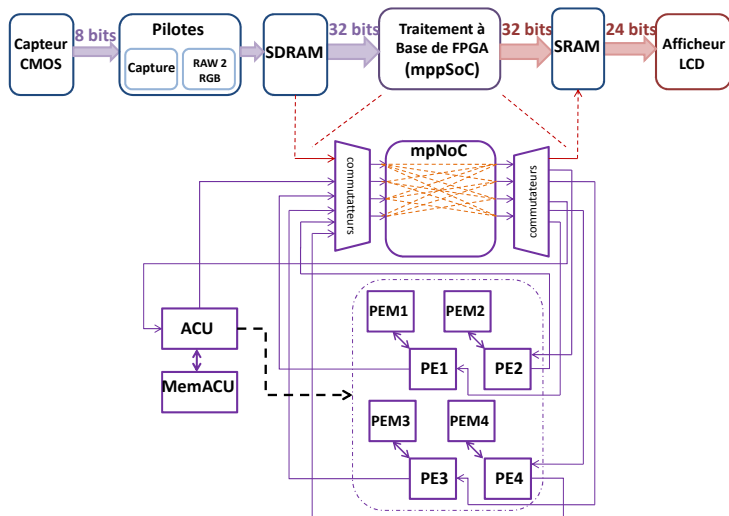
# Intégration dans Gaspard



# Plan

- 1 Introduction
- 2 Système mppSoC proposé
- 3 Méthode de conception/programmation de mppSoC
- 4 Flot de génération de configurations mppSoC
- 5 Étude de cas
  - Chaîne de traitement vidéo à base de mppSoC
  - Choix de configuration mppSoC
- 6 Conclusions et perspectives

# Configuration mppSoC



# Étapes de conception

- Modélisation
  - composants
  - communications
- Déploiement
  - Choisir les IPs à utiliser ⇒ notion de "codeFile"
- Génération de code
  - Transformation modèle vers texte
    - modèle Deployed ⇒ code VHDL
  - Dédire les paramètres à partir des modèles
  - Dédire les IPs à partir du déploiement

Demo `./config1.avi`

- M. Ammar, M. Baklouti et al. *A Model Driven Engineering design approach to generate VHDL for MPPSoC*. **RAPIDO workshop at HIPEAC**. Greece. 2011 (Accepted)

# Flot de conception

- Spécifier une configuration mppSoC à un haut niveau d'abstraction
- Faciliter/accélérer la conception d'une configuration mppSoC

## Flot de conception

- Spécifier une configuration mppSoC à un haut niveau d'abstraction
- Faciliter/accélérer la conception d'une configuration mppSoC

### Chaîne mppSoC à base d'IDM

- réduire la complexité de conception de mppSoC
- modèle UML-MARTE  $\Rightarrow$  code VHDL  $\Rightarrow$  simulation/synthèse

# Choix de la configuration adéquate

- Simulation/Synthèse sur l'FPGA Cyclone II EP2C70F896C6

Méth. conception	Ressources logiques			Mémoire totale			Conso. de puiss. (mWatts)	Tpixel (Ns)
	Fonctions combinatoires	Registres	%	ACU (bytes)	PE (bytes)	%		
réplication	21971	9243	37	4096	1024	11	1207.98	44.95



# Choix de la configuration adéquate

- Simulation/Synthèse sur l'FPGA Cyclone II EP2C70F896C6

Méth. conception	Ressources logiques			Mémoire totale			Conso. de puiss. (mWatts)	Tpixel (Ns)
	Fonctions combinatoires	Registres	%	ACU (bytes)	PE (bytes)	%		
réplication	21971	9243	37	4096	1024	11	1207.98	44.95

- Exécution

Demo `./Exec-config1.AVI`

# Choix de la configuration adéquate

- Simulation/Synthèse sur l’FPGA Cyclone II EP2C70F896C6

Méth. conception	Ressources logiques			Mémoire totale			Conso. de puiss. (mWatts)	Tpixel (Ns)
	Fonctions combinatoires	Registres	%	ACU (bytes)	PE (bytes)	%		
réplication	21971	9243	37	4096	1024	11	1207.98	44.95

- Modification du modèle
- Exécution

Demo ./Model2.m3u

# Choix de la configuration adéquate

- Simulation/Synthèse sur l'FPGA Cyclone II EP2C70F896C6

Méth. conception	Ressources logiques			Mémoire totale			Conso. de puiss. (mWatts)	Tpixel (Ns)
	Fonctions combinatoires	Registres	%	ACU (bytes)	PE (bytes)	%		
réplication	21971	9243	37	4096	1024	11	1207.98	44.95
réduction	14176	4762	23	4096	1024	11	852.65	30

# Choix de la configuration adéquate

- Simulation/Synthèse sur l'FPGA Cyclone II EP2C70F896C6

Méth. conception	Ressources logiques			Mémoire totale			Conso. de puiss. (mWatts)	Tpixel (Ns)
	Fonctions combinatoires	Registres	%	ACU (bytes)	PE (bytes)	%		
réplication	21971	9243	37	4096	1024	11	1207.98	44.95
réduction	14176	4762	23	4096	1024	11	852.65	30

- ⇒ Choisir la configuration adéquate
- ⇒ Adapter le modèle à l'application

## Autres expérimentations

- Benchmarks de traitements d'image (EEMBC)
    - RGB to YIQ
    - Convolution
  - Génération de configurations mppSoC
  - Simulation et synthèse
    - fonctionnalité du système
    - performances : temps d'exécution, surface, consommation de puissance...
- ⇒ Choisir la configuration la plus adéquate
- ⇒ Faciliter l'exploration

# Plan

- 1 Introduction
- 2 Système mppSoC proposé
- 3 Méthode de conception/programmation de mppSoC
- 4 Flot de génération de configurations mppSoC
- 5 Étude de cas
- 6 Conclusions et perspectives**
  - Conclusions
  - Perspectives

# Conclusions

- mppSoC adapté pour le TSS
  - paramétrique
  - programmable
  - conçu à base d'IPs
  
- Flot de conception UML vers VHDL pour mppSoC
  - modélisation haut niveau à base de MARTE
  - génération automatique de code
  - intégration dans Gaspard
  
- Validation expérimentale à base d'FPGA

# Perspectives

- Exploration de configurations mppSoC
  - intégration d'une phase d'exploration
  - génération automatique de la meilleure configuration
  
- Extension du profil MARTE
  - Application data parallèle
  - Architecture SIMD :
    - interface d'envoi d'instructions
    - raffinement d'un IP processeur
    - spécificités SIMD (OR Tree, identité...)
  
- Modèle d'exécution multi-SIMD/SPMD
  - étendre le modèle d'exécution de mppSoC
  - viser un large spectre d'applications de TS



# Publications

## 13 publications durant la thèse :

- 1 chapitre d'ouvrage (2<sup>eme</sup> auteur) (*IGI-Global*)
- 1 journal (*JSA*, Elsevier)
- 9 conférences et workshops internationaux (*IDT'08*, *AICCSA'09*, *ARC'10*, *RAPIDO'11...*)
- 1 PhD Forum (à *FPL'10*)
- 1 participation aux journées doctorales (*EuroDoc-Info'10*)

# MERCI