Logiciel pour les architectures reconfigurables

Bernard Pottier

Équipe Architectures & Systèmes
Université de Bretagne Occidentale

1. Description du domaine

3 niveaux de reconfiguration:

Logique, Chemin de Données, Système

2. Un scénario logiciel pour les systèmes reconfigurables

Description prospective d 'une chaine de compilation

3. Conclusion

1.1 Définitions

- ' Architectures ' reconfigurables

  possibilité matérielle de redéfinir des caractéristiques de la machine de manière statique ou dynamique, complètement, ou partiellement.

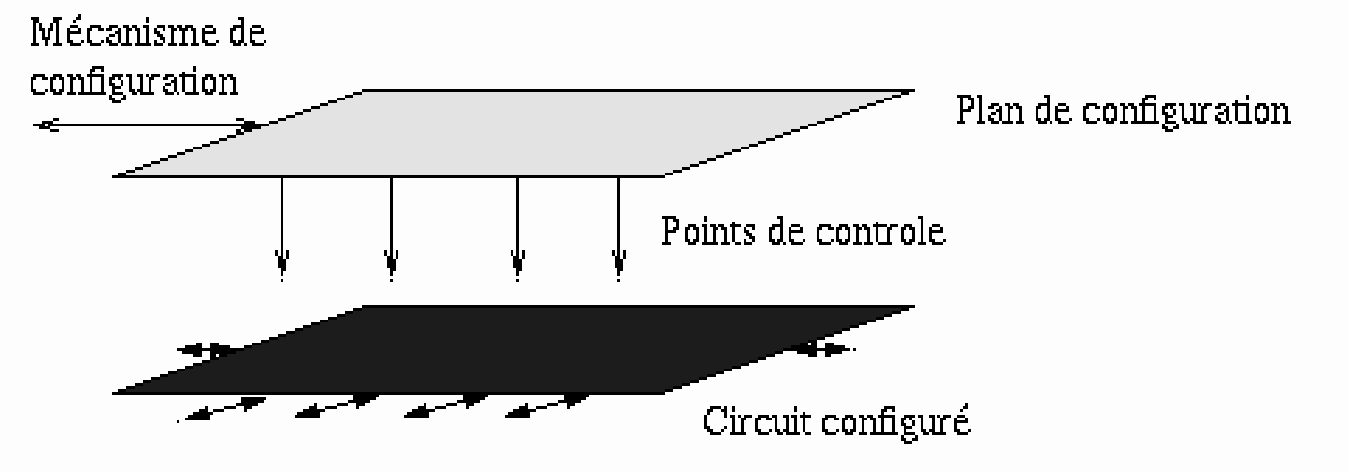
  - FPGA en tant que technologie d 'implémentation
  - Unité FPGA embarquée
  - Unité à chemins de données reconfigurable
  - Processeurs micro-programmables
  - Systèmes reconfigurables

Configuration

•Organisation de valeurs placées sur des points de contrôle

•mécanisme de chargement de ces valeurs dans la machine

•multiplicité possible des plans de configuration



Signification de la configuration (bottom up)

- **L** (Logique) Etablissement de connections de niveau logique, définition de comportements logiques
- **CD** (Chemin Données) Etablissement de connections de niveau chemins de données, contrôles d 'opérateurs
- **S** (Système) Définitions de séquences d 'opérations en liaison avec un jeu d 'instruction, ou un langage

Caractérisation du flot logiciel/matériel

- Le logiciel décrit une organisation de transformations à opérer sur des données, que l 'on peut dénommer `**application**'
- le ' matériel ' décrit une structure d 'exécution physique permettant de réaliser des traitements, que l 'on peut désigner par ' **architecture**'
- La dualité **application-architecture** se retrouve à plusieurs niveaux, avec des options alternatives dans le flot de conception
- Les **outils de conception** assurent le ' **mapping** ' de l 'application vers l 'architecture

Diagramme en Y - mapping

- Le modèle de référence présente l 'application et l 'architecture aux outils de mapping
- L 'architecture et l 'application sont conformes à des modèles abstraits connus et implicites dans les outils
- Les triangles peuvent s 'empiler et se composer
- modèle conforme à une vision générique ' Y-chart ' (TU-Leiden, LIACS)

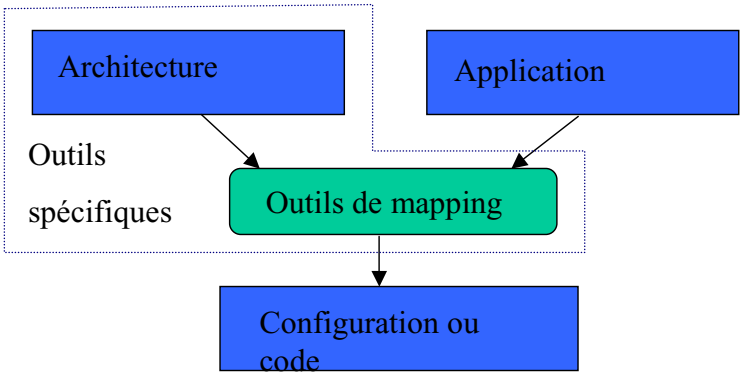
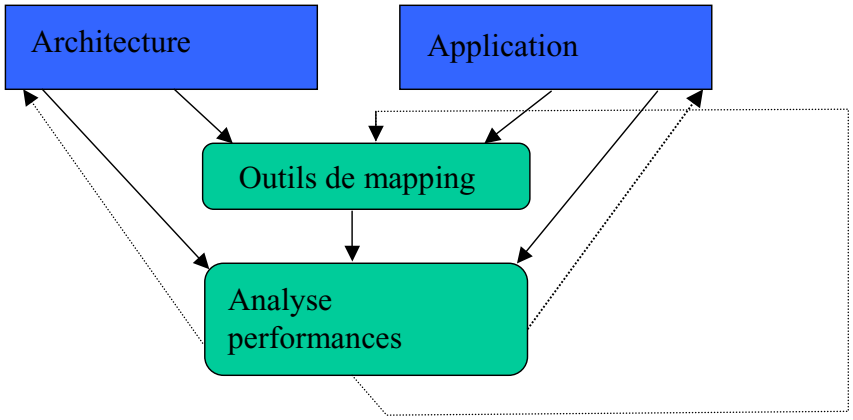


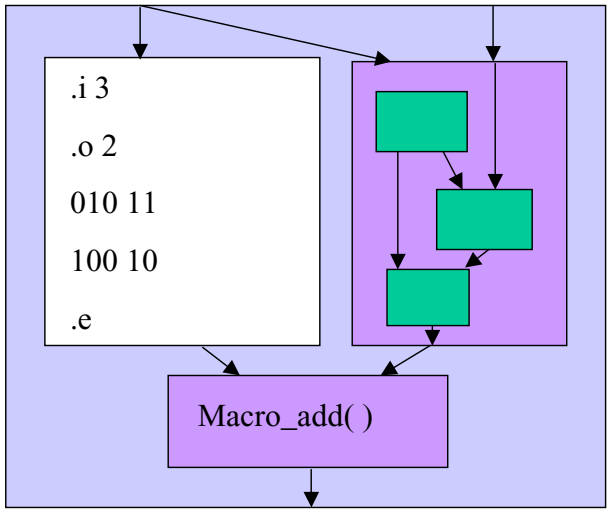
Diagramme en Y - exploration

- Possibilité de recibler l 'application et d 'effectuer de l 'exploration architecturale
- Possibilité de rechercher des implantations optimales pour une architecture donnée
- Concept empilable: système -> chemin de données -> logique ...



1.2 (L) Niveau logique: application

- Input : Fichier d 'interchange (netlist)
- tables logiques
- macros d'édition
- compositions hiérarchiques
- primitives matérielles
- Output : configuration



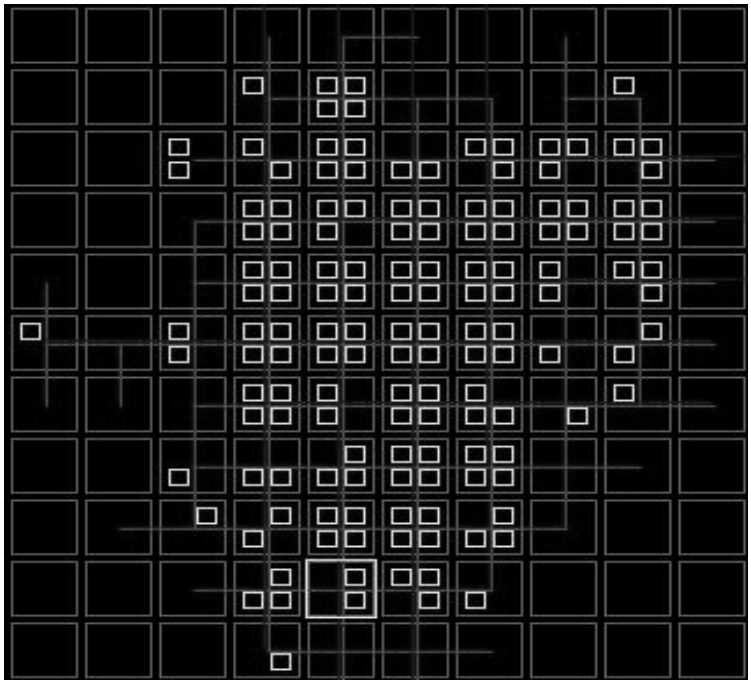
(L) Niveau logique: architecture

Description

- Points d 'interconnexion
- aiguillages
- structure d'une cellule
- comportement logique

Modèle

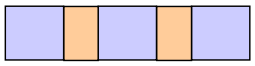
- langage de description structurel paramétrique
- madeo (UBO), vpr (Toronto)



Niveau logique:outils

- Placement:
  - allocation de ressources dans l 'architecture, à grain fin, on alloue des cellules que l 'on caractérise
  - le grain logique est adapté aux ressources de synthèse
- Routage :
  - recherche de chemins pour les signaux, allocation des ressources de routage
- floor-planning
  - placement des composants les uns par rapport aux autres

- Edition:
  - commande des outils de base afin de calculer les architectures
  - place-route
  - copie du module
  - réplication du module
  - routage du canal
  - copie du canal
  - etc...



Processeur - canal - processeur-canal...

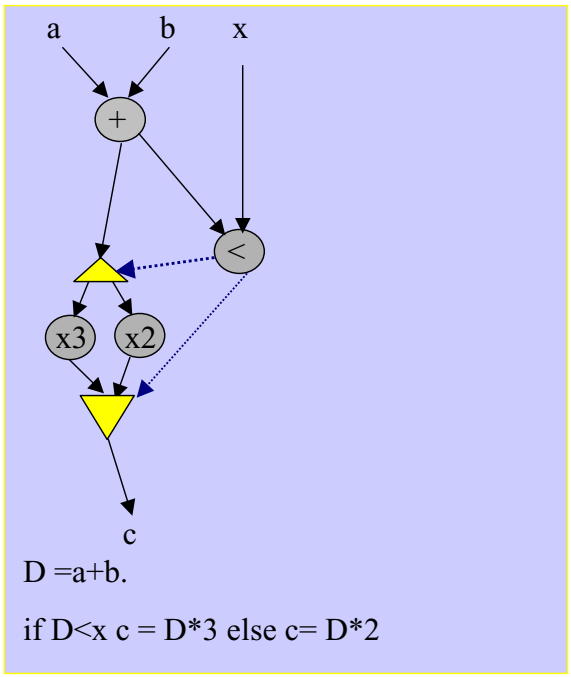
Niveau logique:références

- Architectures commerciales
  - Virtex, virtex2
  - Altera
  - FPGA embarqués et recherche

- Outils de recherche
  - Virtual P&R (J.Rose, Toronto)
  - Jbits (Xilinx)
  - Madeo

1.3 (CD) Niveau Chemin de données : application

- Graphes, plats ou hiérarchiques
  - flots d 'opérations (DFG)
  - flots d 'opération + contrôle (CDFG)
- On s 'appuie sur des opérations primitives
  - cablées (cas des data-paths)
  - configurées (on s 'appuie sur le niveau logique)
  - les primitives peuvent être spécifiques



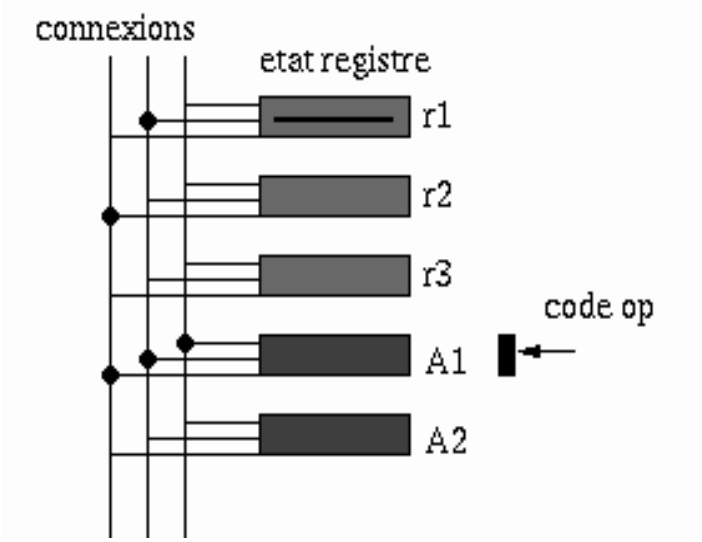
Niveau Chemin de données: architecture

Points de contrôle:

- multiplexeurs
- segments de bus
- contrôle des opérateurs
- sélections de registres

Implantation du contrôle:

- microséquenceurs
- synthèse des controleurs



Niveau Chemin de données: outils

ordonnancement
- segments de bus
- contrôle des opérateurs
- sélections de registres

allocation de ressources
- registres
- opérateurs

synthèse de la configuration
- établissement des connexions
- définition du contrôle

Contraintes
- fréquence, latence
- ressources disponibles
- disponibilité des données
- consommation

- Plus ou moins: le savoir faire en synthèse d 'architecture explicité par Gajski/DeMicheli,
- appliqué à des structures fixes
- ou synthétisées

Niveau Chemin de données: exemples

- **DART de l 'ENSSAT**
- **XPP de PACT**
  - réseau d 'opérateurs
  - graphe d 'opérations pipelinées asynchrone.
  - communication paquet
  - reconfiguration dynamique
  - programmation vectorielle extraite de programmes C
  - langage de description des configurations (NML)

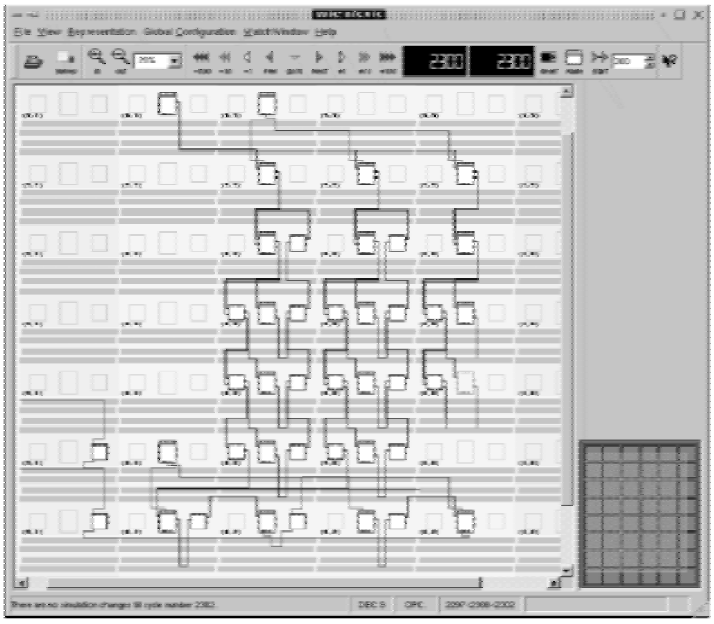


Figure 8. xvis Screen Shot

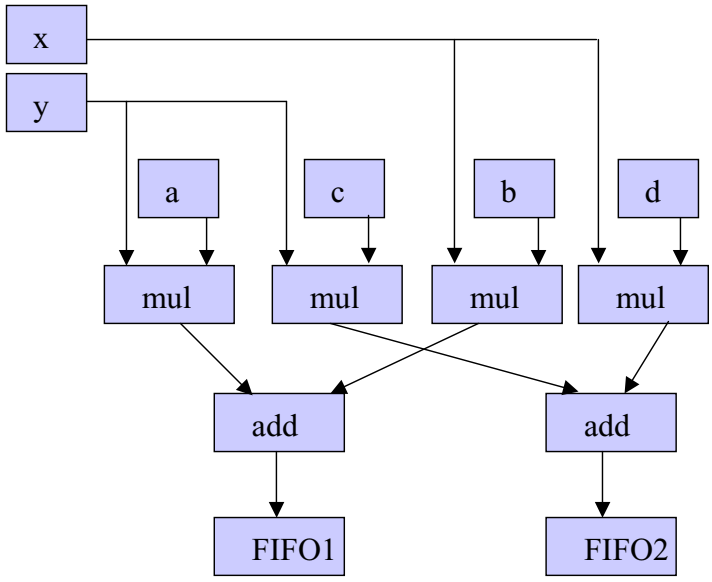
Niveau Chemin de données: exemples

XPP de PACT

- Le graphe d 'opérations est placé et routé sur le circuit
- les PAE groupent des ALU, registres d 'entrées et de sorties
- les PAE sont connectés à un réseau qui fait circuler données et évènements

Physiquement

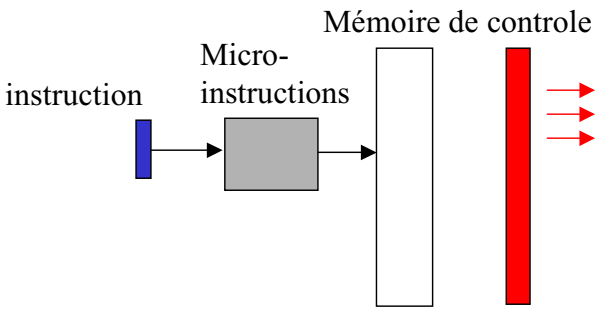
- 64 ALU de 32 bits
- 16 mémoires 1Ko
- 4 ports de 32 bits

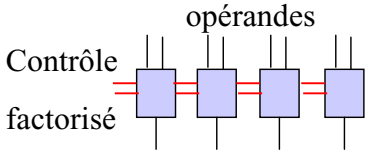


1.4 (S1) Niveau Système: architecture

Microprogrammation

- **Présence de mots de contrôle permettant de définir des transferts et des opérations dans le processeur**
  - **conceptuellement on peut assimiler ce contrôle à de la reconfiguration dynamique (opérateurs, transferts)**
  - **les opérateurs sont configurés ' par lots ', en parallèle**
- **Possibilité de construire des séquences de micro instructions associées à des instructions**
  - **le contrôle peut être redéfini pour construire des instructions spécifiques**





(S1) exemples d 'architectures

Architecture MOLEN (DELFT)

- **Jeu d 'instruction englobant la configuration et l 'exécution de modules**
- **détails de contrôle rejetés dans le reconfigurable**
- **registres de statut et registres partagés**

Symptome (CEA)

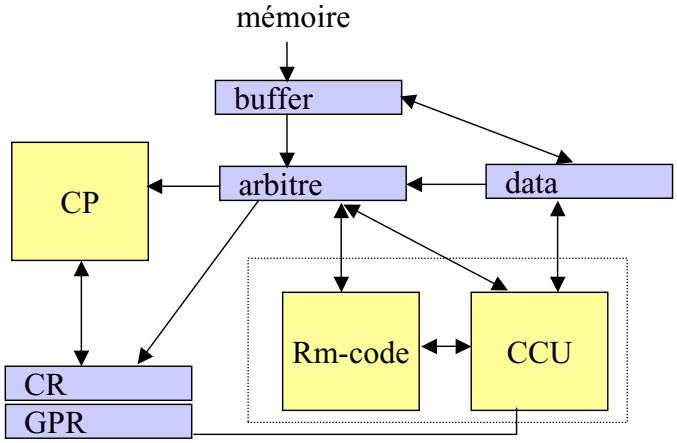
- **Multi threading dans le micro code**
- **asynchronisme**
- **réseaux de Petri + opérateurs synthétisés**

•Future direction of programmable and reconfigurable embedded processors (SAMOS02)

•Thèse de F.Blanc

Architecture de contrôle (MOLEN)

- Architecture de MOLEN
  - CP: Core processing
  - CR : Control register
  - GPR : registres partagés
  - rm-code : configurations
  - CCU : custom configured unit
- Support d 'instructions de configuration exécution
  - S/X set execute
  - R/P resident/pageable
  - adresse



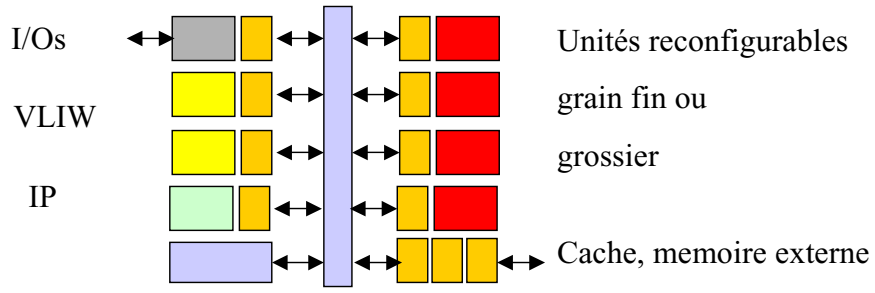
Format d 'une instruction set/execute

S/C	R/P	Adresse mémoire configuration

(S2) Niveau Système: architecture

Systèmes reconfigurables

- **Composition**
  - **présence d 'une ou plusieurs unités reconfigurables**
  - **Ips, processeurs de calcul**
  - **réseau intégré**
  - **processeur de contrôle, mémoire distribuée**



(S2) Niveau Système: application

Processus et tâches

- **Constructions hiérarchiques à la CSP**
  - **orientation contrôle**
  - **horloge implicite, canaux, description structurelle statique**
- **Processus de Kahn**
  - **orientation flôts multimedia**
  - **graphes de tâches produit par compilation, ou décomposition programmée**
  - **adapté à la reconfiguration dynamique**

•Handel-C (celoxica.com)

•SCORE (Berkeley)

•Compaan (TU Leiden)

(S2) Niveau Système: outils

Processus communicants/petites tâches

- **Constructions hiérarchiques à la CSP**
  - **traduction HDL/netlist directe, peu de transformations**
  - **floor planning, placement, routage (outils xilinx/altera)**
- **Processus de Kahn, par exemple (TU Leiden),**
  - **Matlab vers affectation unique**
  - **construction d 'un graphe de dependances polyhedral reduit**
  - **construction d 'un réseau de taches**
  - **synthèse ou compilation sur 1 ou plusieurs unités (?)**
- **Méthodes (SynDeX, ..)**
- **Compilation de code orienté objet modulaire (prospectif)**

•Compilation from matlab to process networks (Cases99)

1. Description du domaine

3 niveaux de reconfiguration:

Logique, Chemin de Données, Système

2. Un scénario logiciel pour les systèmes reconfigurables

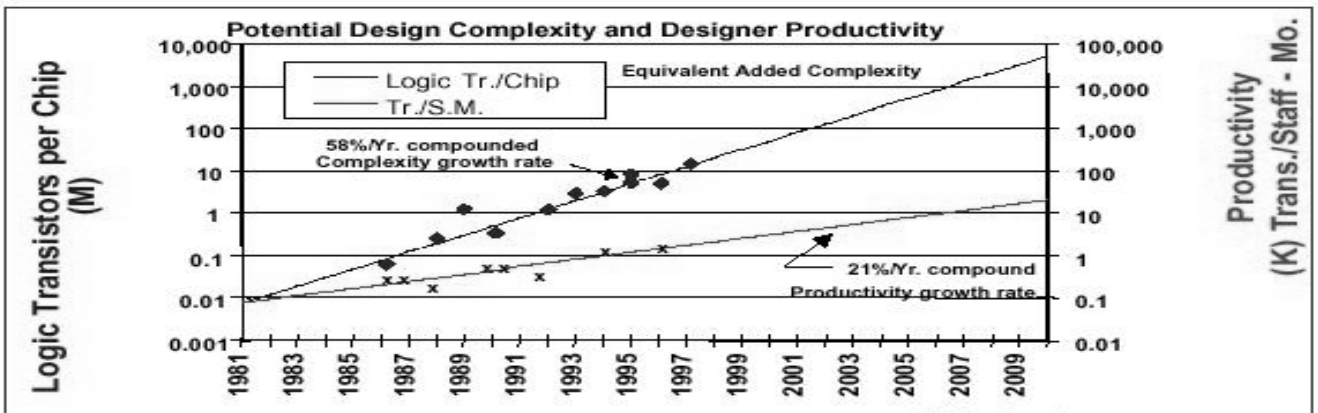
Description prospective d 'une chaine de compilation

3. Conclusion

Objectifs

- Réutilisabilité des composants matériels pour plusieurs applications
  - portée économique similaire à celle de circuits de type VirtexII,
  - architecture système scalable
  - outils de haut/très haut niveau
- Développement rapide
  - pas de HDL
  - compilateurs ciblant des domaines d 'application (flots, contrôle)
- Composition des outils/architectures structurée verticalement
  - piles de diagramme Y
  - modèles formalisant à tous les niveaux de la pile:
    - les architectures,
    - la structure des applications et,
    - les outils

Nécessité de la productivité en développement

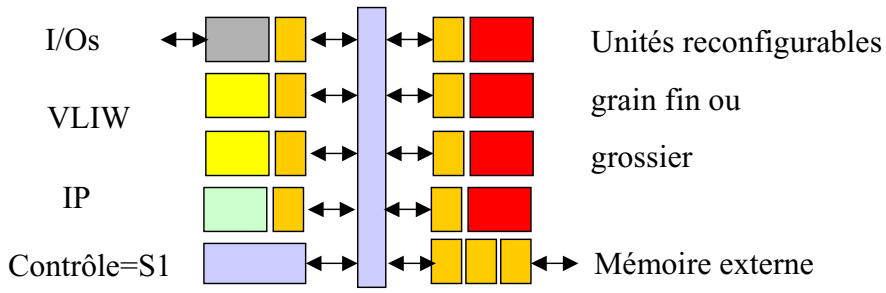


Year	Technology	Chip Complexity	Frequency	3 Yr. Design Staff	Staff Cost*
1997	250 nm	13 M Tr.	400	210	90 M
1998	250 nm	20 M Tr.	500	270	120 M
1999	180 nm	32 M Tr.	600	360	160 M
2002	130 nm	130 M Tr.	800	800	360 M

* @ $150K / StaffYr. (In 1997 Dollars)

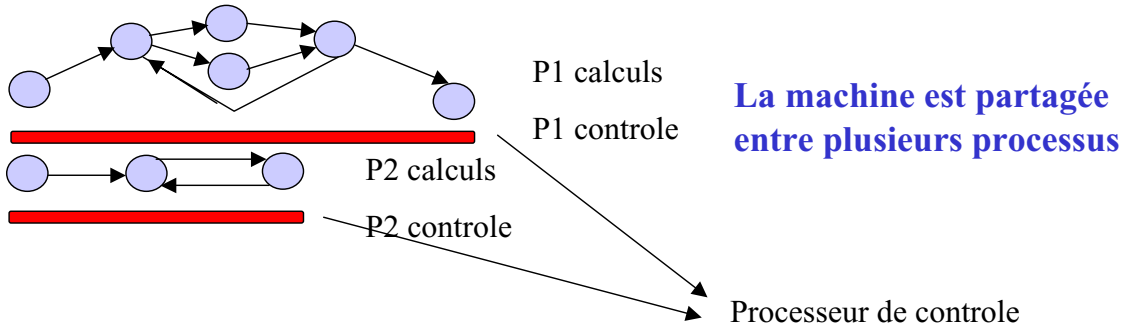
2.1 Modèle d 'architecture (S1+S2)

- Modèle système de l 'architecture (rappel)
  - ensemble d 'unités interagissant autour d 'un bus
  - communication tamponnée par paquets
  - mémoires distribuées, et/ou DMA pour l 'intégration des accélérateurs reconfigurables
  - présence d 'un processeur maître contrôlant le réseau, et les unités via des canaux d 'évènements
  - présence d 'un système de pagination anticipant ou tamponnant les mouvements de données pour les streams, et jouant le rôle de cache



2.2 Schéma d 'exécution (S1+S2)

- Restructuration des programmes d 'application
  - les noyaux de calcul, ou les blocs exigeant une forte disponibilité sont groupées dans des taches (partitionnement horizontal)
  - le programme est partitionné verticalement en un flot de contrôle et un flot d 'opérations taches de niveau (CD)
  - les taches sont ensuite synthétisées et/ou compilées pour les unités, reconfigurables et/ou programmables
  - le système (OS) choisit à l 'exécution l 'unité appropriée et alloue cette ressource (chargement de code et/ou configuration dynamique)

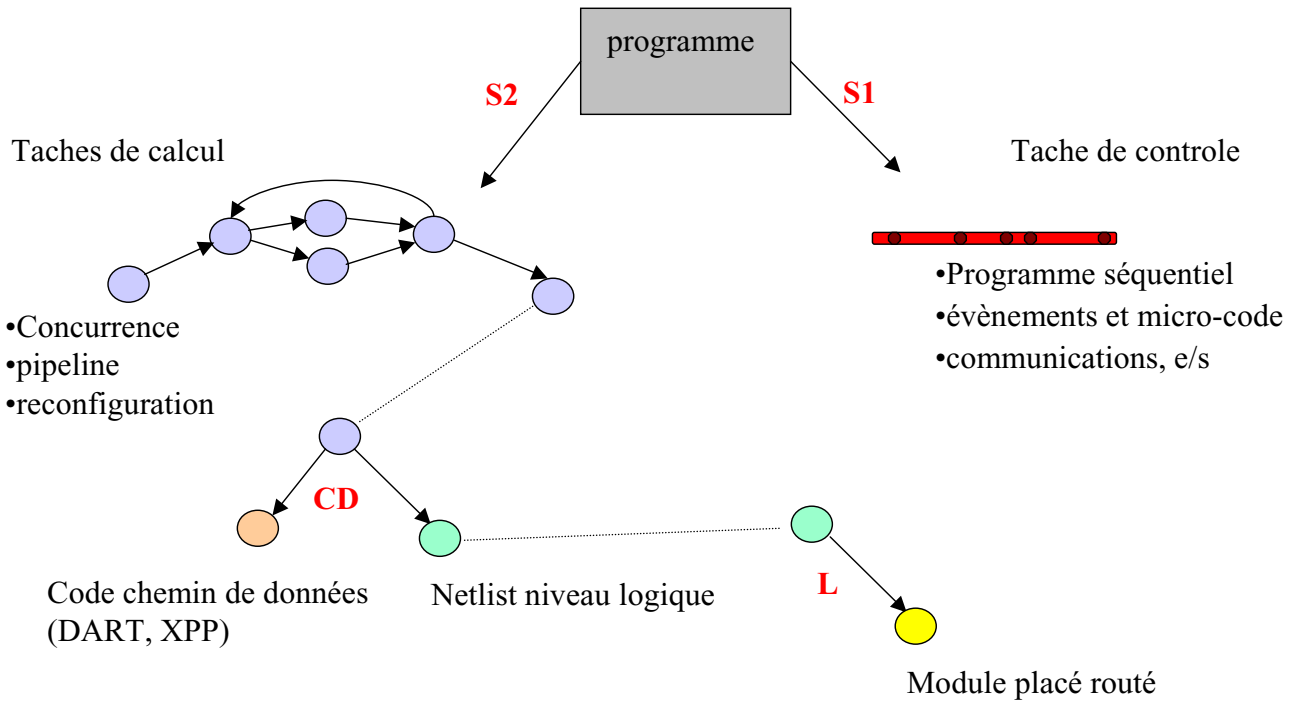


La machine est partagée entre plusieurs processus

2.3 Flot de compilation

- S1+S2
  - partitionner les programmes en tâches, si besoin,
  - générer le flot de contrôle et le programme de séquencement
  - générer les tâches (plusieurs versions possibles)
- CD
  - pour chaque cible possible, compiler, ou synthétiser,
  - générer le code de communication et de synchronisation,
  - générer les contrôleurs locaux
- L
  - bottom-up : produire des librairies de composants pour la synthèse
  - top-down : en complément, ou en alternative à CD
    - produire des opérateurs spécifiques
    - synthétiser le contrôle

Logiciels pour le reconfigurable - Ecole architecture - Roscoff, Avril 2003



Décomposition en taches (S2)

- Compilation de l 'application (S)
  - partitionnement en tâches, automatique, semi-automatique, ou manuel
  - appel du niveau chemin de données (CD) pour produire le code pour une, ou plusieurs unités
  - génération du programme système communication+controle

En se basant sur des outils de TU-Leiden...

•Entrée en matlab, puis productions de:

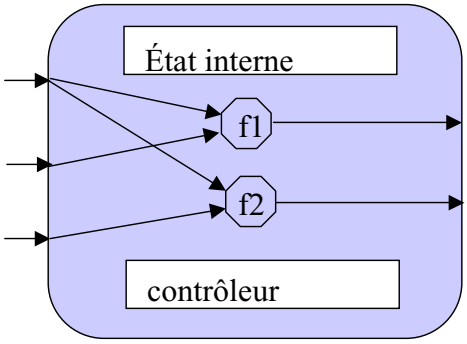
•Single assignment code (SAC)

•Polyedral reduced dependence graph (PRDG)

•Stream-based Function model (SBF)

•Conditionnement pour les unités de calcul (?)

(ref Cases99, Compaan, ..)



Compilation des tâches (CD)

- Il faut des outils spécifiques pour chaque cible
- cas de XPP (CD)
  - entrée = flot de données
  - ordonnancement des opérations
  - allocations de ressources réalisant ces opérations
  - placement
  - génération de la configuration
- cas de Madeo (CD+L)
  - entrée = CDFG + spec des données
  - séparation contrôle données
  - production d 'un graphe hiérarchique de tables synthétisables
  - optimisations sur le graphe
    - fusions de table
    - propagation de constantes
    - élimination des valeurs inutiles
  - génération de code logique
  - appel de la couche L pour le placement routage.

Génération de logique (L)

- Modélisation de l 'architecture cible
  - formalisation des connectivités, de la logique disponible et de la hiérarchie du composant
- Parcours en profondeur du composant d 'application
  - synthèse des primitives logiques,
  - dessin des composants structurés
  - positionnement géométrique
  - construction des canaux de routage
- Analyse de performance
  - production d 'informations quantitatives

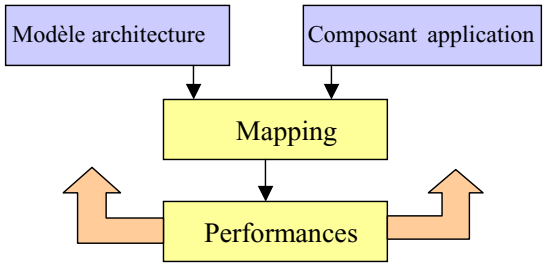
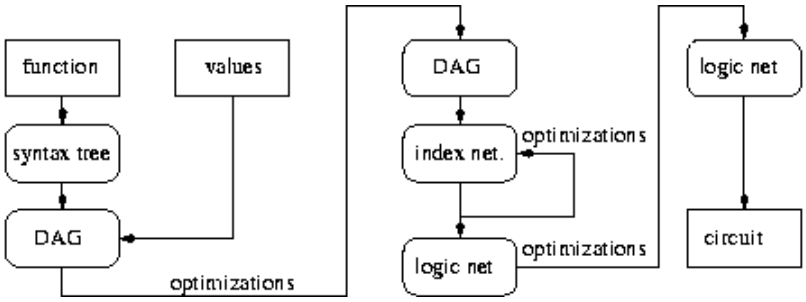


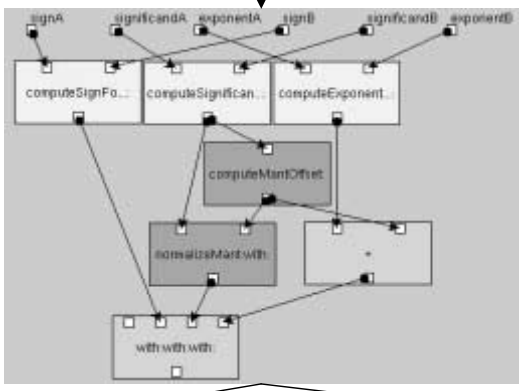
Diagramme en Y

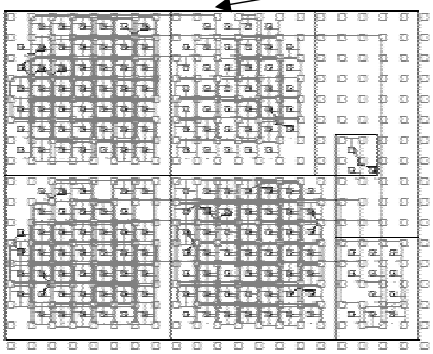
Génération de logique (L)



- Prétraitements à haut niveau avant la synthèse logique
  - moins de charge sur l 'optimisation logique
  - capacité à traiter de gros problèmes
  - niveau de spécification : calcul symbolique
- Construction de libraires d 'opérateurs pour le niveau CD
- Portabilité et support des Y-charts

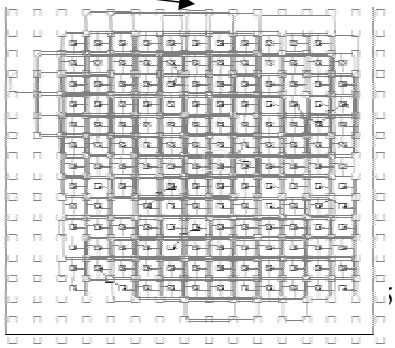
Code symbolique + données transformées





Synthèse hiérarchique

Synthèse à plat



(Architecture FPGA STMicro, Berkeley)

1. Description du domaine

3 niveaux de reconfiguration:

Logique, Chemin de Données, Système

2. Un scénario logiciel pour les systèmes reconfigurables

Description prospective d 'une chaine de compilation

3. Conclusion

Concepts présentés

- Y-chart : paramétrage des outils par des modèles reconnus pour les applications et les architectures
- Chaine ouverte de transformations structurée S/CD/L autorisant des boucles d 'exploration
- Miscibilité avec des composants en dur, donc acceptable
- Peut-etre efficace, mais il y a un problème majeur:
  - la référence est celle des implémentations optimisées en VHDL
  - le typage des langages actuel ne donne pas assez d 'information
  - il faut disposer de spécifications précises sur les données et inférer automatiquement ces spécifications
- Dynamicité de la reconfigurabilité au niveau système

Extraits de bibliographie

fichier bib : http://as.univ-brest.fr/~pottier/roscoff.bib

- **Labo Architectures et Systèmes, UBO**
  - http://as.univ-brest.fr/reports
  - http://as.univ-brest.fr/~pottier/SAMOS.pdf
- **Projet BRASS, U.C.Berkeley (utilisation dynamique des ressources reconfigurables)**
  - papier FPL2001 sur SCORE, implémentation MPEG, + Mescal..
  - Http://brass.cs.berkeley.edu
- **Workshop CASES99, ..**
  - Compilation from Matlab to process networks (TU-Leiden)
- **Workshop SAMOS02 (Dekker, ed. 2003**
  - Future directions of empbedded processors (TU-Delft)
  - Realizations of the extended .. Compaan tool chains (High level system evaluation, liacs, tu-leiden)
- **PACT XPP**
  - A self reconfigurable data processing architecture, autres docs http://www.pactcorp.com
- **Projets (et relations industrielles)**
  - relations régionales (IRISA, LESTER, ..)
  - Projet RNTL OSGAR (CEA/UBO/IRISA/TNI)
  - Projet PHRASE (STMicroelectronics/UBO/IRISA/Enstb)
  - DART (Enssat),
  - Alpha, PIPS, …