Thèse E.N.S.T.

Etude d'architectures VLSI numériques parallèles et asynchrones pour la mise en oeuvre de nouveaux algorithmes d'analyse et rendu d'images

Frédéric ROBIN

Plan

- **Contexte de l'étude**
  - évolution du codage d'images
  - évolution des architectures VLSI pour la communication visuelle
    - limitations et perspectives
- **Introduction à l'asynchronisme**
  - modèles de calcul parallèle, algorithmes, architectures, circuits VLSI
- **Exploitation des différents niveaux d'asynchronisme dans des coprocesseurs parallèles pour l'analyse / rendu d'images**
  - AMPHIN - un réseau VLSI asynchrone pour le filtrage morphologique
  - PACAP - un modèle d'architecture cellulaire asynchrone programmable

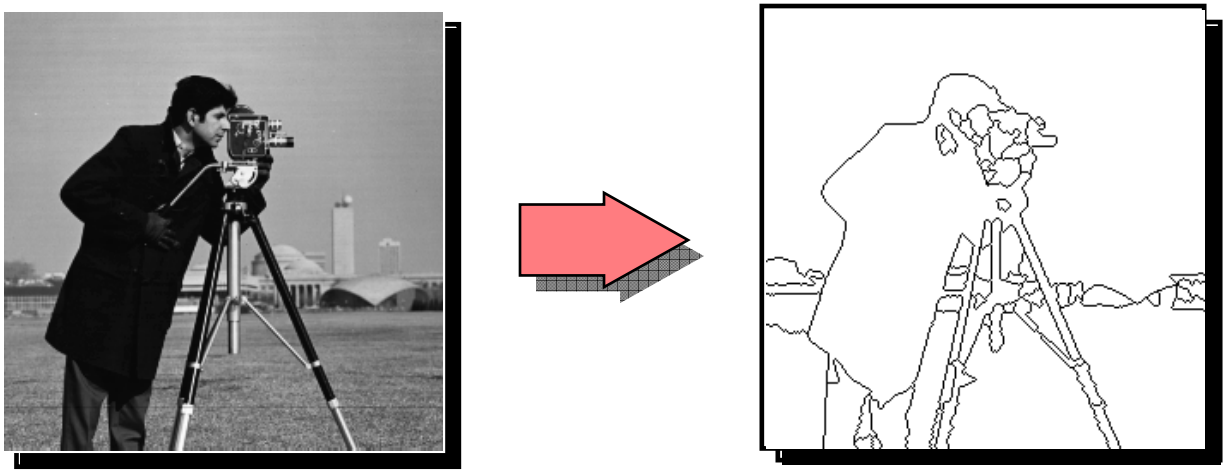
Introduction

- **Développement des applications de communication visuelle numérique**
  - visioconférence, vidéo à la demande, consultation multimédia...
- **Normes de compression d'images animées (H.261/263, MPEG1/2)**
  - exploitation de la redondance spatiale et temporelle
  - codage hybride (par transformée + prédictif) par blocs
- **Evolution du codage d'images**
  - supprimer les artéfacts de codage
  - prendre en compte de nouvelles techniques
  - dépasser la représentation niveau pixel

Evolution du codage d'images (1)

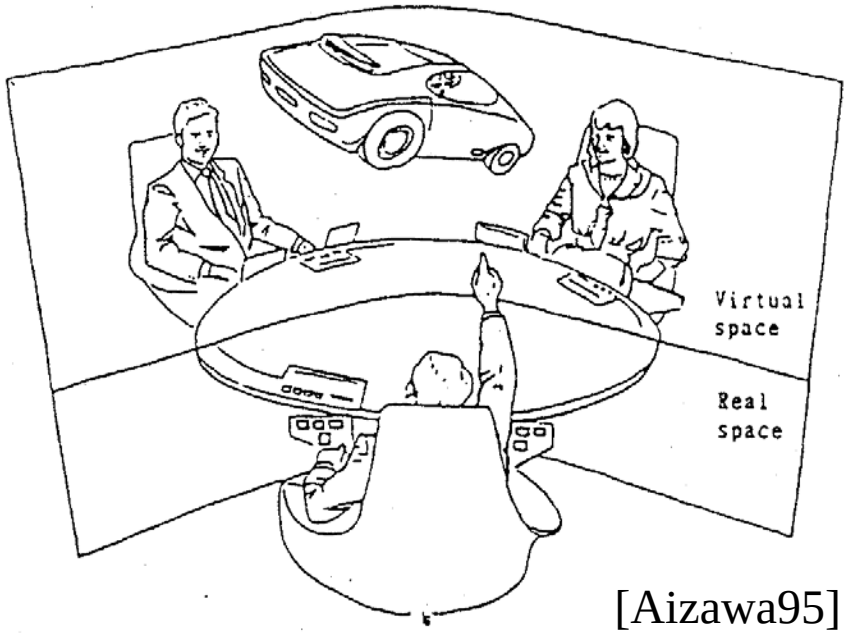
- **Du signal à l'objet**
  - vers une représentation de plus haut niveau
  - du traitement du signal à l'analyse-synthèse basée-objet
  - paramètres de mouvement, de forme, de texture

- **Codage par analyse-synthèse**
  - basé-région
  - basé-modèle



Evolution du codage d'images (2)

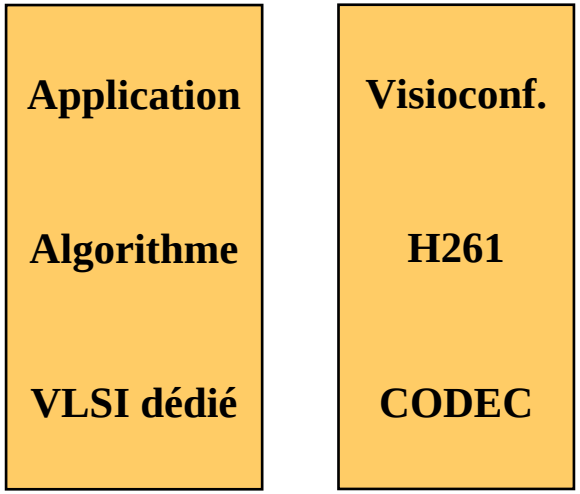
- **Du codage au méta-codage (MPEG4)**
  - au delà de la compression
  - incorporation de nouvelles fonctionnalités
- **Caractéristiques**
  - prise en compte de l'évolution des algorithmes
  - accès universel
  - interactivité basée-contenu
  - intégration de contenus hybrides
  - convergence analyse/codage/rendu
- **Besoins**
  - puissance de calcul
  - généricité, flexibilité



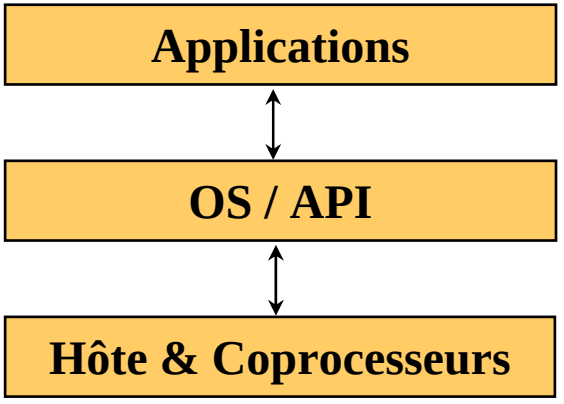
[Aizawa95]

Evolution des architectures VLSI pour la communication visuelle (1)

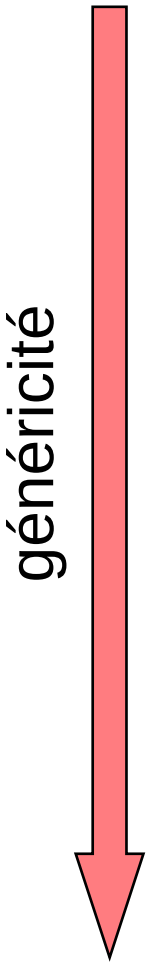
- **Spécialisation "verticale"**
  - analyse d'images
  - compression d'images
  - rendu d'images



- **Vers la programmabilité "horizontale"**
  - des coprocesseurs spécialisés aux processeurs programmables
  - début de convergence vidéo & graphique



Evolution des architectures VLSI pour la communication visuelle (2)



Analyse	Compression	Rendu
processeurs associatifs	chipsets spécialisés	mémoires à traitement intégré
	coprocesseurs vidéo	processeurs de rendu 3D
array-processors SIMD	Video Signal Processors	
	media processors	
	CPUs à extension multimédia	

Limitations des processeurs multimédia

- Spécificité liée aux systèmes de première génération
- Exploitation du parallélisme trop réduite
  - parallélisme instruction vs parallélisme de données
  - SIMD limité (intra-mot) et contraint (régularité)
  - mémoire centrale partagée
- Modèles de séquencement trop contraints
  - flot de contrôle séquentiel
  - fonctionnement SIMD trop rigide
- Limitations de la conception logique classique : les circuits synchrones
  - synchronisation globale de plus en plus contraignante
    - hypothèses et marges temporelles
    - mauvaises modularité, scalabilité, réutilisabilité
    - consommation ?

Perspectives architecturales

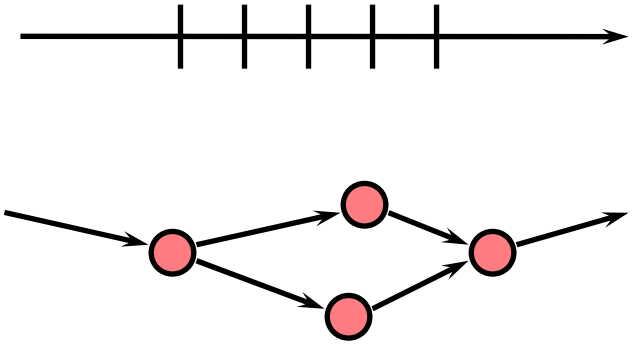
- **L'évolution technologique permet l'évolution architecturale**
  - que faire avec 100 millions de transistors ?
  - nouvelles contraintes et opportunités
- **Parallélisme massif**
  - besoin en puissance pour l'analyse / rendu d'images
  - parallélisme de données inhérent au traitement d'images
- **Asynchronisme à grain fin**
  - besoin en flexibilité de contrôle
    - évolution du contexte multimédia
    - analyse d'images : traitements irréguliers et qui dépendent des données
  - aussi au service du parallélisme et des performances

Plan

- **Contexte de l'étude**
  - évolution du codage d'images
  - évolution des architectures VLSI pour la communication visuelle
    - limitations et perspectives
- **Introduction à l'asynchronisme**
  - modèles de calcul parallèle, algorithmes, architectures, circuits VLSI
- **Exploitation des différents niveaux d'asynchronisme dans des coprocesseurs parallèles pour l'analyse / rendu d'images**
  - AMPHIN - un réseau VLSI asynchrone pour le filtrage morphologique
  - PACAP - un modèle d'architecture cellulaire asynchrone programmable

Introduction à l'asynchronisme

- **Vous avez dit "asynchrone" ?**
  - contextes : de la modélisation fonctionnelle à l'implémentation VLSI
  - définitions dépendent du niveau de description
  - synchrone : "se fait dans un même temps"
- **Le temps ("khronos")**
  - date, instant
    - relation d'ordre totale sur l'ensemble des événements
  - succession, causalité
    - ordre partiel



- **Asynchronisme**
  - indétermination de la relation d'ordre entre événements
  - relâchement des dépendances (par rapport à une "référence" synchrone)

Modèles de calcul parallèles asynchrones

- **Interfaces entre algorithmes et architectures**

- **Asynchronisme = indétermination a priori**
  ⇒ plus de degrés de liberté

  - durée / ordre d'exécution des étapes élémentaires
  - taille des phases / nombre relatif de barrières de synchronisation
  - communications / synchronisations locales
  - exécution conditionnelle des communications
  - traitement flot de données

Communications asynchrones

- **Modèles asynchrones (à délais non bornés)**
- **Communications bloquantes**
  - rendez-vous, point de synchronisation
  - attente de la disponibilité des ressources de communication
- **Communications non-bloquantes**
  - pas d'attente de la disponibilité des ressources
  - test de l'état des ressources (ex: test du canal vide)
  - instabilité de la condition testée = indéterminisme possible

Algorithmes asynchrones (1)

- **Influence possible de l'indétermination sur les dépendances fonctionnelles**
  - asynchronisme fonctionnel
  - dépendances dynamiques ou "relâchées" (indéterminisme)
  - pas nécessairement de points de synchronisation déterminés

- **Exemple: algorithme itératif asynchrone**

$$x_i(j) = \begin{cases} x_i(j-1) & si\ i \notin J_j \\ F_i\left(x_1(s_1(j)),\ldots,x_n(s_n(j))\right) & si\ i \in J_j \end{cases}$$

$$s_i(j) \leq j-1$$

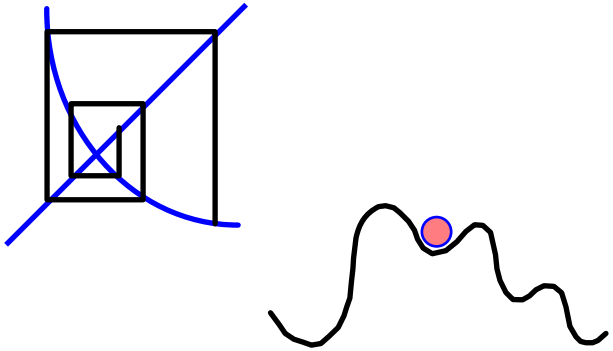
$$\lim_{j\to\infty} s_i(j) = \infty$$

$$\forall K, \exists j > K \,/\, i \in J_j$$

Algorithmes asynchrones (2)

- **Lien avec les communications non-bloquantes**
  - traitement au plus tôt
  - pas d'attente de la disponibilité des ressources de communication
  - test du canal vide = choix indéterministe (cf dépendances dynamiques)

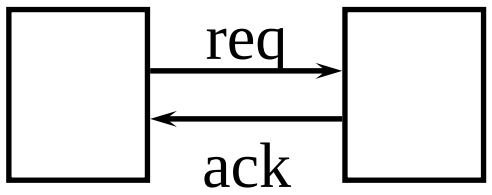
- **Influence sur le résultat**
  - convergence vers un point fixe unique

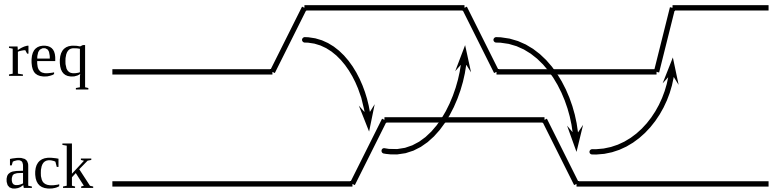


  - solutions multiples (problèmes d'optimisation)
  - systèmes dynamiques complexes et comportement chaotique...

Architectures asynchrones

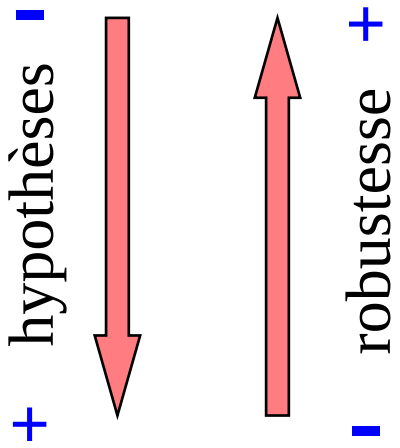
- **Modèle structurel**
  - décomposition d'un système en un ensemble d'unités interconnectées
  - contraintes liées à l'implémentation
- **Mécanismes de synchronisation**
  - architecture synchrone: signal global d'horloge
  - asynchrone = synchronisations locales (unités de contrôle réparties)
  - signalisation de validité des données
  - indétermination de la durée implique signalisation bidirectionnelle
    - protocole de requête-acquittement ou "handshake"





Circuits asynchrones

- **Synchronisation locale (pas d'horloge globale)**
- **Classification selon les hypothèses temporelles**
  - circuits insensibles aux délais
  - circuits quasi-insensibles aux délais
  - circuits indépendants de la vitesse
  - circuits à micropipelines
  - circuits à délais bornés



- **Evolution de la maturité des circuits asynchrones**
  - opérateurs arithmétiques, microprocesseurs

Potentiels de l'asynchronisme

- séparer spécification et contraintes de réalisation
  - élargissement du spectre de solutions
- accélération fonctionnelle
  - convergence des algorithmes itératifs
  - propagation au plus tôt de l'information
- calcul en temps minimum
  - exploitation des variations des temps de calcul

$$\underset{\{PEs\}}{Max} \sum_{\{instr.\}} \left(T_{instr.} \right) \leq \sum_{\{instr.\}} \underset{\{PEs\}}{Max} \left(T_{instr.} \right)$$

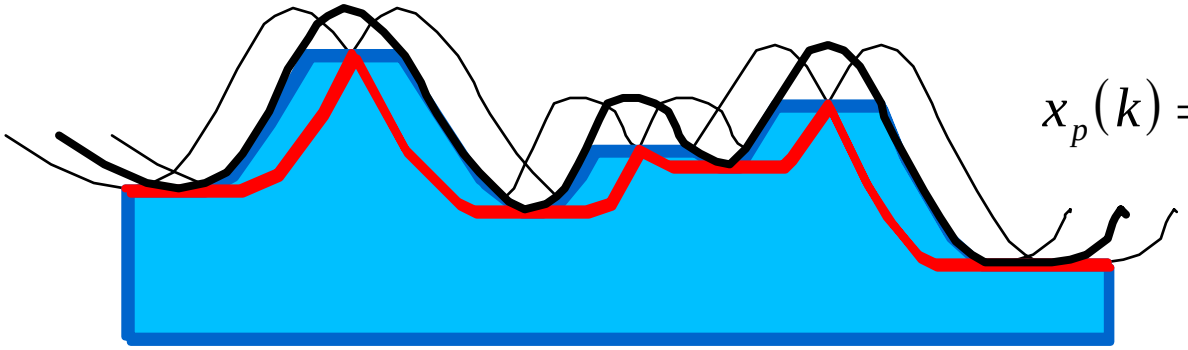
- suppression des pénalités de synchronisation globale
  - pas d'attente inutile, pas de clock skew, pas de marges temporelles
- optimisation de la consommation
- robustesse et auto-adaptation aux conditions de fonctionnement
- modularité, réutilisabilité, migration aisée

Plan

- **Contexte de l'étude**
  - évolution du codage d'images
  - évolution des architectures VLSI pour la communication visuelle
    - limitations et perspectives
- **Introduction à l'asynchronisme**
  - modèles de calcul parallèle, algorithmes, architectures, circuits VLSI
- **Exploitation des différents niveaux d'asynchronisme dans des coprocesseurs parallèles pour l'analyse / rendu d'images**
  - AMPHIN - un réseau VLSI asynchrone pour le filtrage morphologique
  - PACAP - un modèle d'architecture cellulaire asynchrone programmable

Exploitation conjointe de différents niveaux d'asynchronisme

- **AMPHIN le circuit**
  - Architecture Massivement Parallèle Homogène à grain fIN
  - étude conjointe algorithme-architecture
  - asynchronisme structurel et fonctionnel
- **algorithme pilote : filtrage morphologique d'images**
  - approche géométrique et ensembliste
  - ouverture par reconstruction : érosion + dilatation géodésique itérée



$$x_p(0) = \varepsilon_n\left(r_p\right) = Min\left\{r_{p+q}, q \in M_n\right\}$$

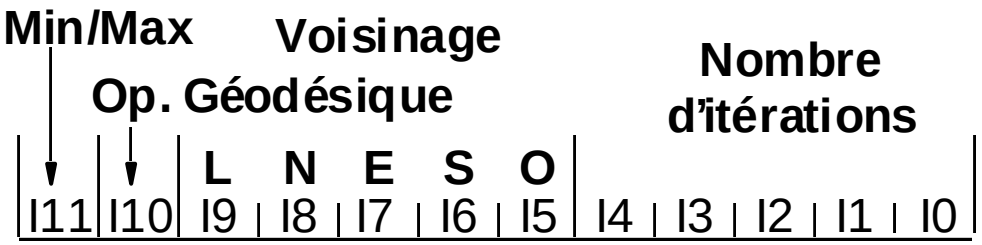
$$x_p(k) = Min\left(Max\left\{x_{p-q}(k-1), q \in M_1\right\}, r_p\right)$$

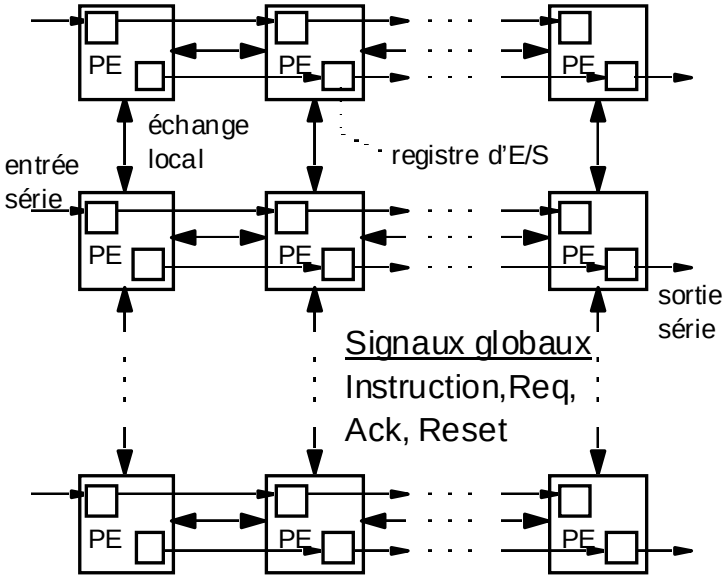
Reconstruction parallèle asynchrone

- **Relâchement des contraintes de dépendances**
  - itération globale synchrone $$x_p(k) = Min\left(Max\left\{ x_{p-q}(k-1), q \in M_1 \right\}, r_p \right)$$
  - itération locale asynchrone $$x_p\left(k_p\right) = Min\left(Max\left\{ x_{p-q}\left(k_{p-q}\right), q \in M_1 \right\}, r_p \right)$$
- **Propriétés**
  - résultat identique
  - pas de synchronisation
  - favorise la propagation au plus tôt

Architecture cellulaire asynchrone

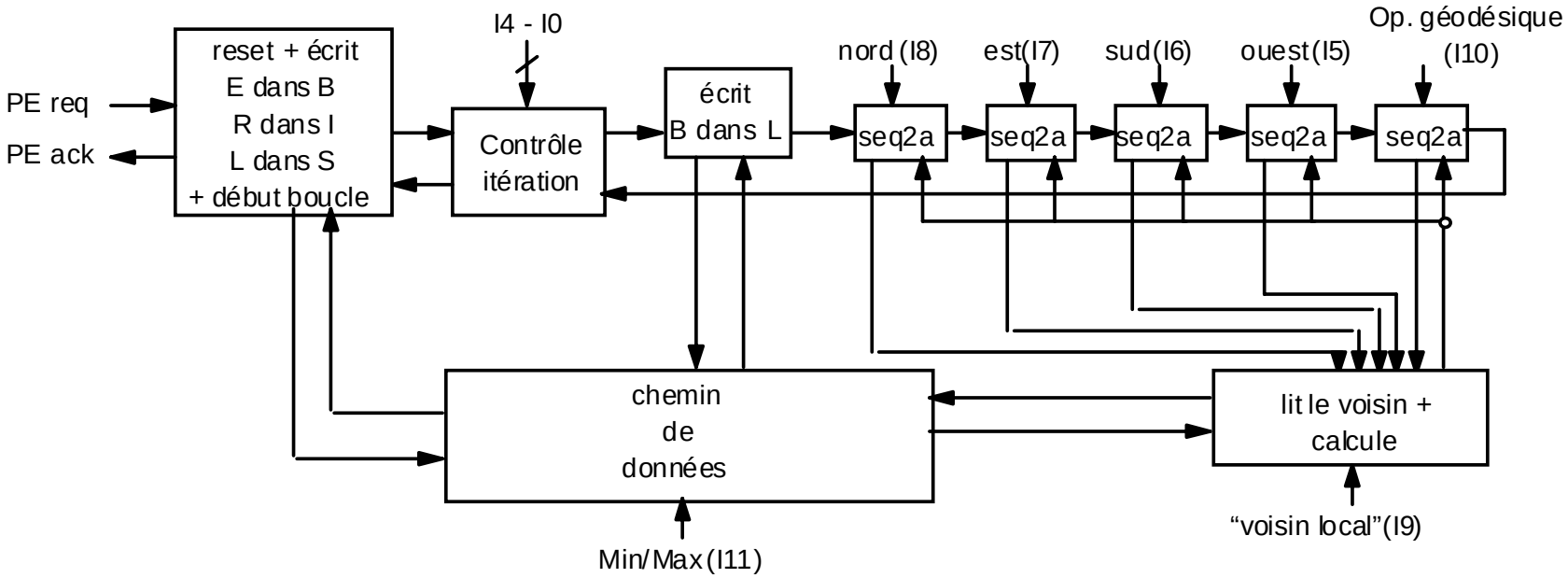
- **Asynchronisme architectural**
  - opérateurs asynchrones à détection de fin
  - communications non bloquantes
    - calcul au plus tôt à partir des valeurs disponibles
- **Réseau de processeurs élémentaires**
  - grille 2D à maille carrée
  - entrées/sorties pipelinées
  - fonctionnement de type SPMD





Conception asynchrone d'un PE

- approche "standard cells" + ascendante
- circuits indépendants de la vitesse
- partie opérative en logique DCVS double rail
- partie contrôle composée à partir de modules élémentaires simple rail
- interface de communication à base de "Q-Flops"



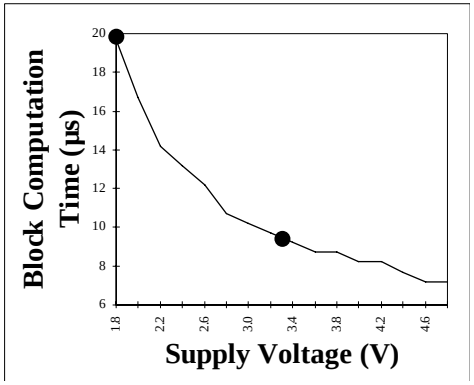
AMPHIN : circuit test

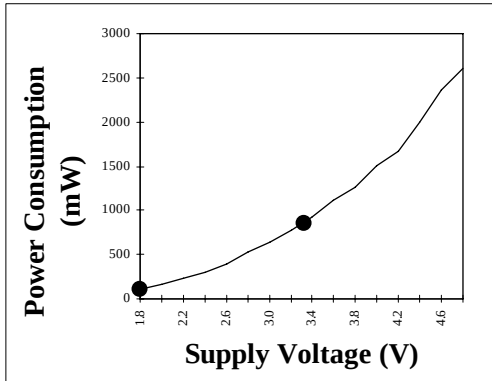
Caractéristiques

- réseau de 16*16 processeurs-pixel asynchrones paramétrables
- 800.000 transistors, 72 mm^2, CMOS 0.5 µm
- filtrage morphologique en temps réel (images 256*256 à 30 Hz)

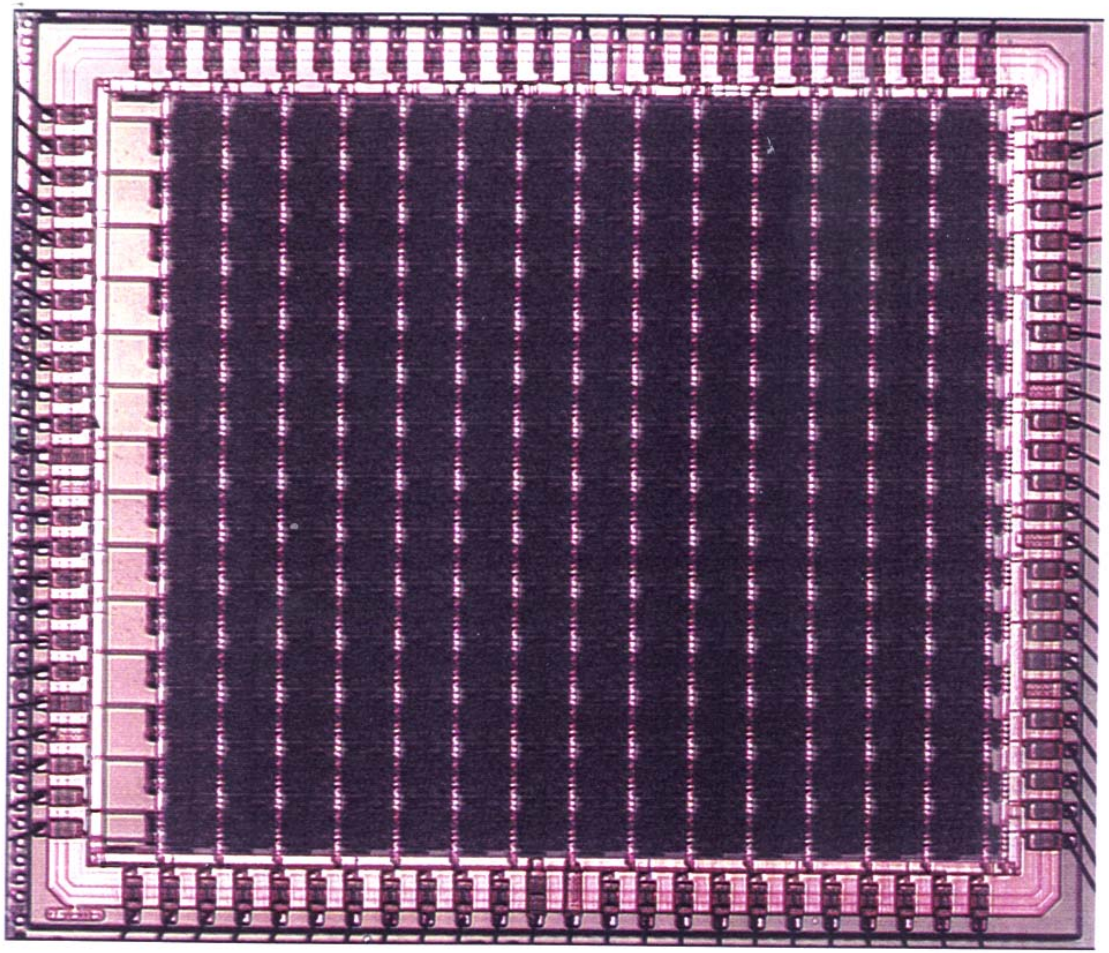
Résultats de test

- fonctionnel pour Vdd de 1.75 V à 4.75 V
- temps de calcul d'un bloc (31 itérations)
  - à 3.3 V : 9 µs
  - à 1.8 V : 20 µs
- consommation
  - à 3.3 V : 900 mW
  - à 1.8 V : 100 mW





AMPHIN : photomicrographie



FT-VD-033-1

AMPHIN : conclusion

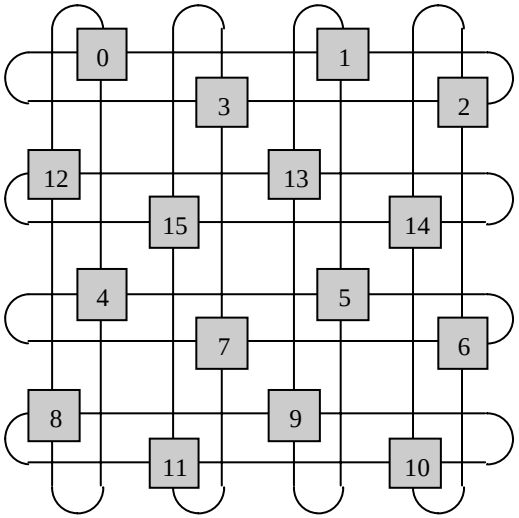
- **Mise en oeuvre d'un asynchronisme fonctionnel et structurel**
  - application au filtrage morphologique
  - offre de nouvelles possibilités de conception conjointe (algo.+archi.)
    - relâchement des contraintes de synchronisation
- **Avantages**
  - calcul en temps minimum
  - adaptation de la vitesse / consommation
  - facilité de conception, robustesse

Généralisation : PACAP

- **Modèle d'architecture cellulaire asynchrone programmable**
  - répondre aux exigences des systèmes de deuxième génération
- **Exploiter le parallélisme potentiel**
  - convergence mémoire-traitement et granularité intermédiaire
- **Contrebalancer la régularité structurelle avec la flexibilité de séquencement**
  - contrôle distribué mixte : SPMD, cellulaire, associatif, flot de données
- **Principe de base: synchronisation minimale**

Un réseau toroïdal asynchrone programmable

- partitionnement d'une grille = tore
- mémoire distribuée
  - convergence mémoire-traitement
    - résultats intermédiaires + buffer vidéo +E/S indépendantes
  - mémoires associatives
    - support du traitement associatif et flot de données
- synchronisation minimale à tous les niveaux
  - contrôle SPMD : autonomie des PE
  - contrôle flot de données
  - circuits asynchrones



Coeur RISC asynchrone d'un PE

- **PE = cœur RISC de faible complexité**
  - architecture "registre-registre"
  - traitements de niveau pixel : entiers de 8 bits
  - instructions spécifiques
    - mémoire associative
    - communications inter-PE
- **Passage aux circuits QDI**
  - travaux en cours (M. Renaudin, P. Vivet : projet ASPRO-216)
  - méthodologie de conception basée sur le CHP (A. Martin, Caltech)
    - approche descendante
  - robustesse et performance
    - architecture superpipeline + exécution dans le désordre

Mémoire partitionnée associative

- **Accès associatifs**
  - recherche associative
    - donnée d'une valeur et d'un masque
    - comparaison parallèle
    - un "tag" est positionné sur chaque ligne de la mémoire
  - comptage des tags actifs
  - lecture associative
    - retourne une donnée dont le tag est actif et son adresse
  - écriture associative
    - donnée d'une valeur et d'un masque
    - modification parallèle de toutes les valeurs dont le tag est actif
- **Support du traitement associatif et flot de données**
  - ex: traitements basés-régions

Interfaces de communications locales (1)

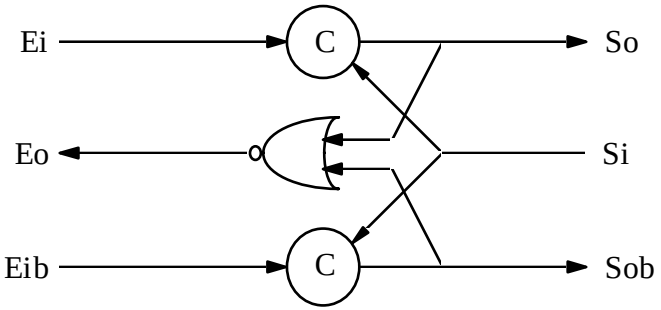
- **Instructions pour communications non-bloquantes**
  - passage de messages
  - support de l'asynchronisme fonctionnel
  - instructions de synchronisation pour communications bloquantes
- **Voisinage programmable**
  - registre banalisé
  - lecture
    - choix d'un message parmi les voisins spécifiés
    - retourne l'origine du message
  - écriture
    - duplication du message vers tous les voisins spécifiés
    - retourne les voisins à qui le message n'a pu être envoyé

Interfaces de communications locales (2)

- communication bloquante

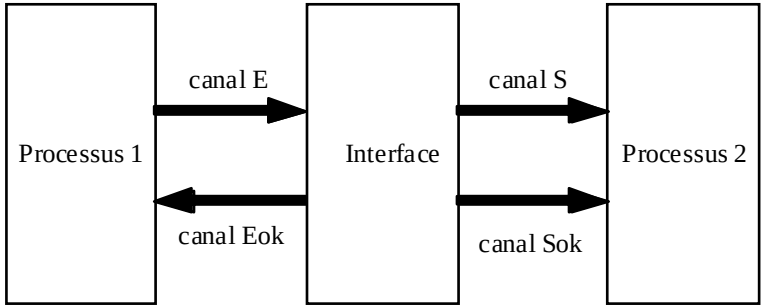
  * [E?x ; S!x]

  [Si^Ei] ; So+ ; Eo- ; [Si^Ei] ; So- ; Eo+
  [Si^Eib] ; Sob+ ; Eo- ; [Si^Eib] ; Sob- ; Eo+



- communication non-bloquante

```
* [ #E ->  [ sema=0 -> E?x , Eok!1 , sema:=1
            @ sema=1 -> E? , Eok!0
            ]
  | #S ->  [ sema=1 -> S!x , Sok!1 , sema:=0
            @ sema=0 -> S!x , Sok!0
            ]
  ]
```



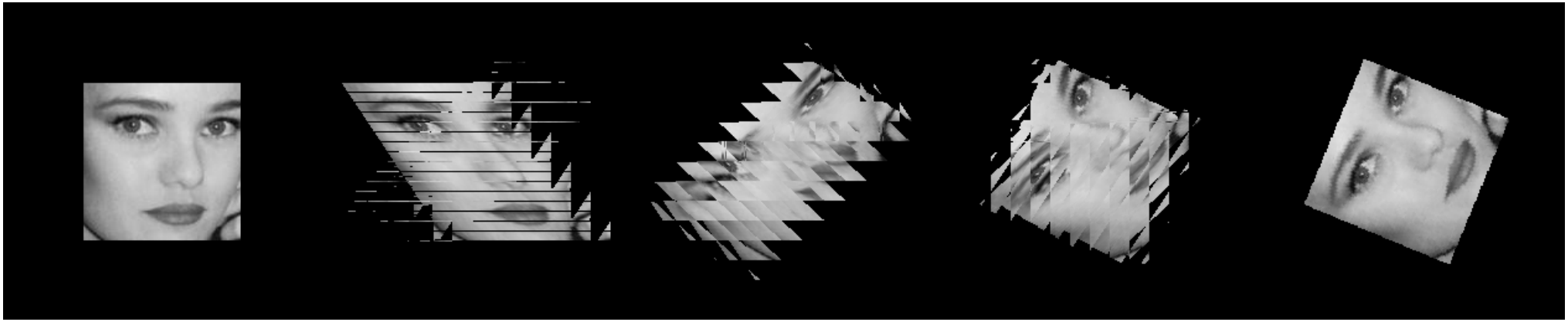
Evaluations de primitives algorithmiques (1)

- **Simulateur événementiel au niveau instruction**
  - entrée : image + programme assembleur SPMD
  - sortie : image + mesures liées à l'exécution de l'algorithme
- **Mise en oeuvre des différents modes de contrôle**
  - produit matriciel
    - algorithme cellulaire régulier, communications bloquantes
  - filtrage morphologique
    - versions: itérative LPGS, flot de données LSGP, flot de données entrelacée
    - communications non bloquantes et asynchronisme fonctionnel
  - rotation d'objets
    - algorithme cellulaire, translations entières
    - traitement basé-région
    - contrôle associatif, flot de données et cellulaire

Evaluations de primitives algorithmiques (2)

- confirmation des bénéfices de la synchronisation minimale
  - calcul en temps minimum
  - partitionnement LPGS + entrelacement flot de données + communications non-bloquantes = bon équilibrage de charge

- illustration du comportement asynchrone



Conclusion

- **Combinaison du parallélisme et de l'asynchronisme à grain fin**

- **Exploitation de l'asynchronisme à plusieurs niveaux**
  - de la spécification algorithmique à la conception VLSI
  - relâchement des dépendances et synchronisations

- **Nouvelles perspectives de conception conjointe**
  - prototype AMPHIN : faisabilité
  - modèle d'architecture PACAP + simulateur
    - exploitation en termes de programmation et d'application

Perspectives

- **Localité = meilleure exploitation des technologies ULSI**
  - algorithmes et architectures cellulaires
  - circuits asynchrones
  - étude du modèle PACAP à poursuivre

- **Exploration du lien entre asynchronisme et traitement d'images**
  - influence sur le résultat

- **Convergence capteur-traitement et systèmes de vision intégrés**