

# CONCEPTION ET TEST DE CIs

---

## 2. FLOT DE CONCEPTION D'UN CIRCUIT INTEGRE

2.1 Introduction

2.2 Synthèse système ou "Co-Design"

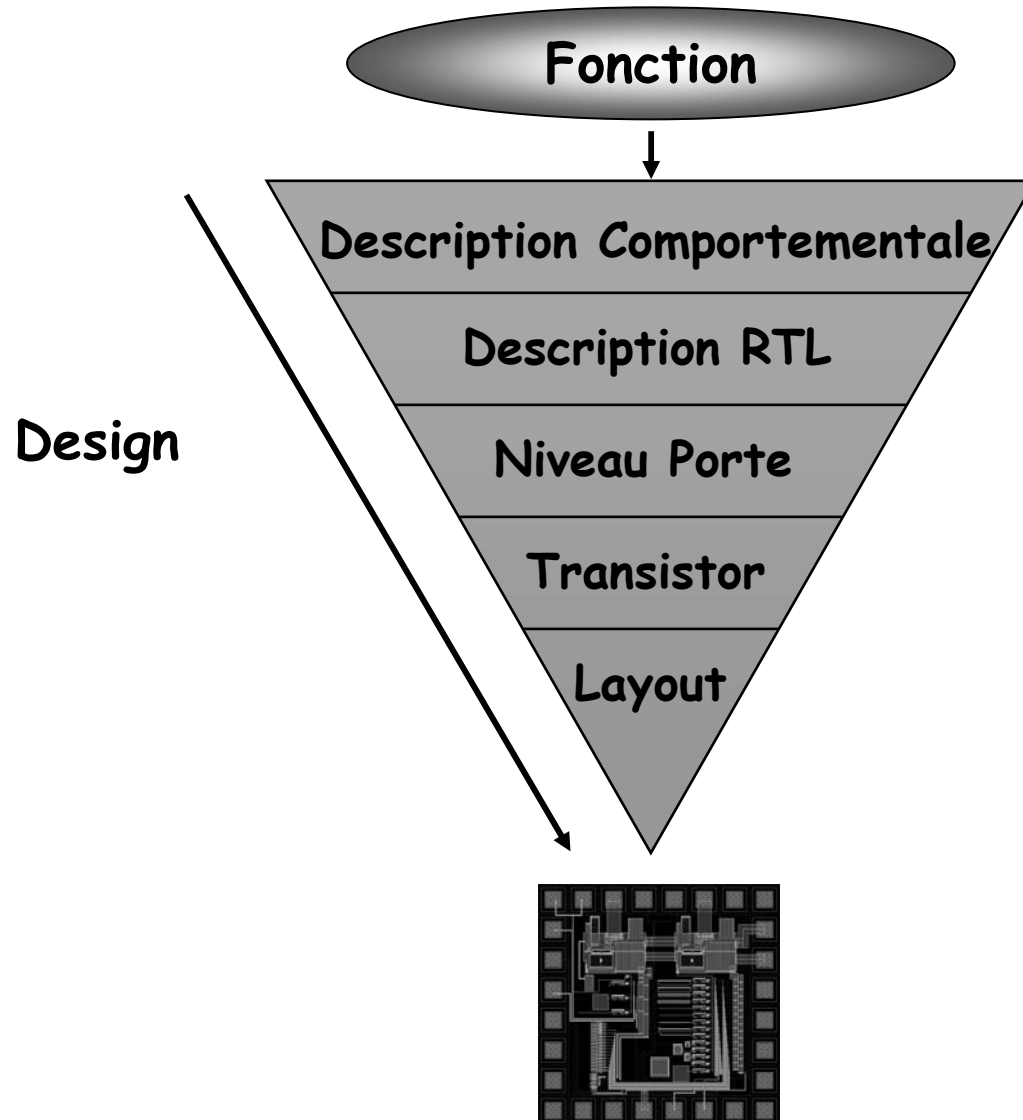
2.3 Synthèse architecturale

2.4 Synthèse logique

2.5 Synthèse physique

## 2. FLOT DE CONCEPTION - Introduction

---



**"Top-Down Methodology"**  
(flot classique)

Concevoir, c'est affiner la description du circuit du niveau comportemental jusqu'au niveau layout.

Passer d'un niveau au suivant et vérifier le résultat obtenu se fait de façon semi automatique en utilisant des outils logiciels

## 2. FLOT DE CONCEPTION - Introduction

---

### Exemple de description comportementale

Une description comportementale définit la fonction du circuit sans partitionner explicitement ce dernier en plusieurs blocs et sans définir les connexions entre éléments.

```
...
architecture behavioral of counter is
...
if clk'event and clk='1' then
  res <= s2;
  s2 <= s1;
  s1 <= a + b;
end if
...
if reset ='0' then in_state <= "00";
case in_state is
  when "00" => in_state <= "01";
  when "01" => in_state <= "10";
...

```

## 2. FLOT DE CONCEPTION - Introduction

### Exemple de description au niveau RTL

- ▶ Decrit l'architecture du circuit sous forme de blocs logiques
- ▶ Chaque bloc est représenté par son code VHDL ou Verilog
- ▶ Des milliers de lignes de code pour un circuit complet

module (nom + liste des ports à rajouter)

always @(posedge ck)

case (ir[15:13])

3'b000: pc <= m [ir [12:0]];

3'b001: pc <= pc + m [ir [12:0]];

3'b010: acc <= -m [ir [12:0]];

3'b011: m [ir [12:0]] <= acc;

3'b100,

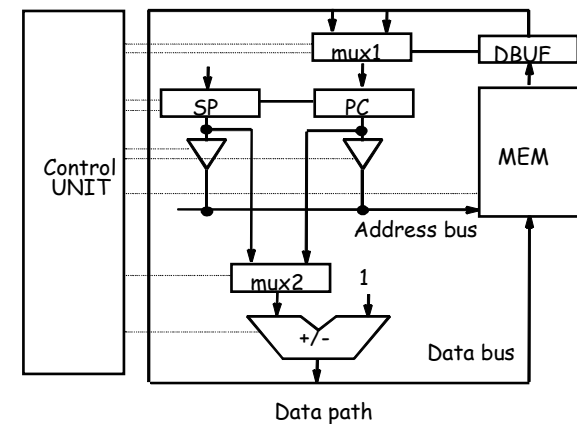
3'b101: acc <= acc - m [ir

[12:0]];

3'b110: if (acc < 0) pc <= pc + 1;

endcase

endmodule

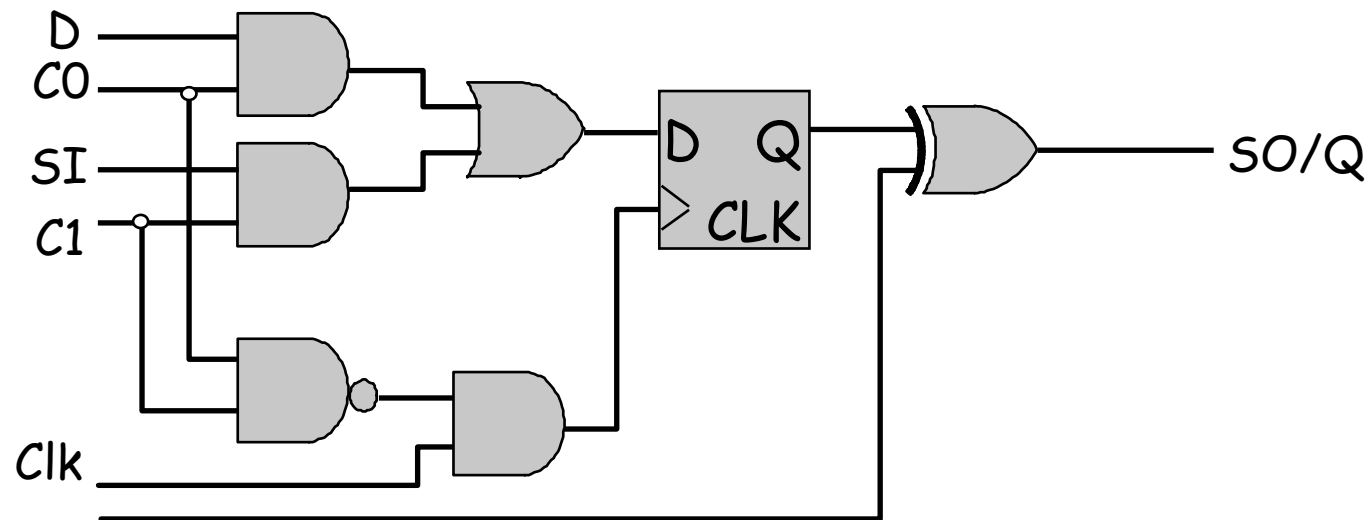


## 2. FLOT DE CONCEPTION - Introduction

---

### Exemple de description au niveau porte

Réseau d'opérateurs booléens et d'éléments mémoire : Netlist

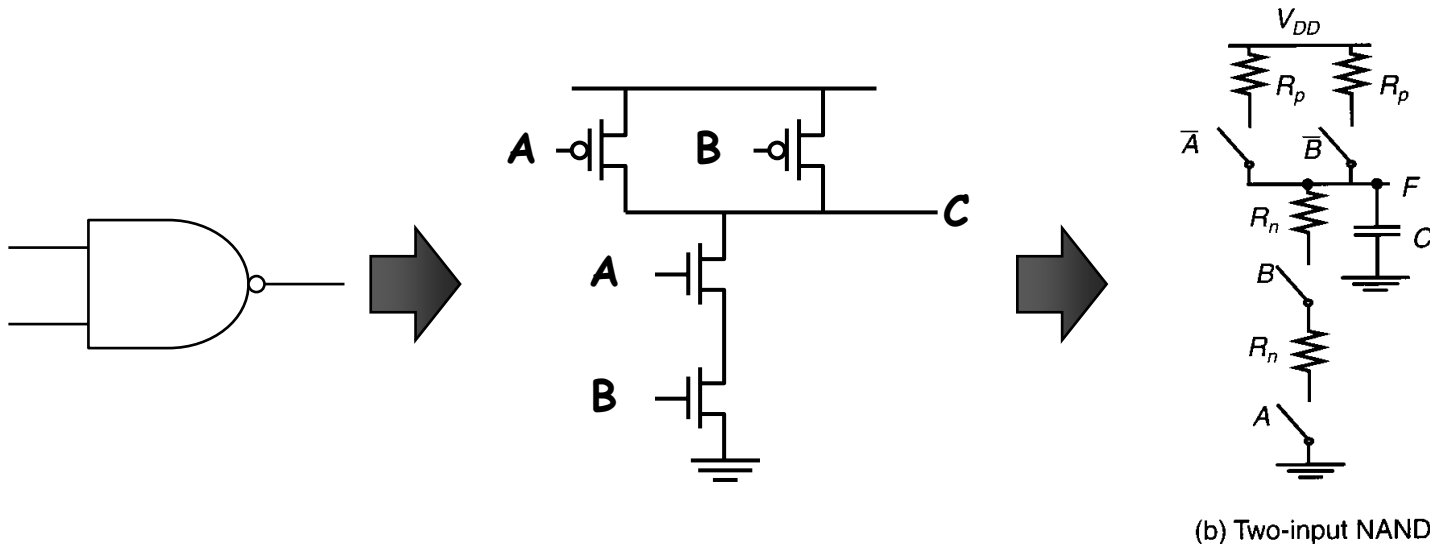


## 2. FLOT DE CONCEPTION - Introduction

---

### Exemple de description au niveau transistor

Chaque porte est remplacée par son réseau de transistors

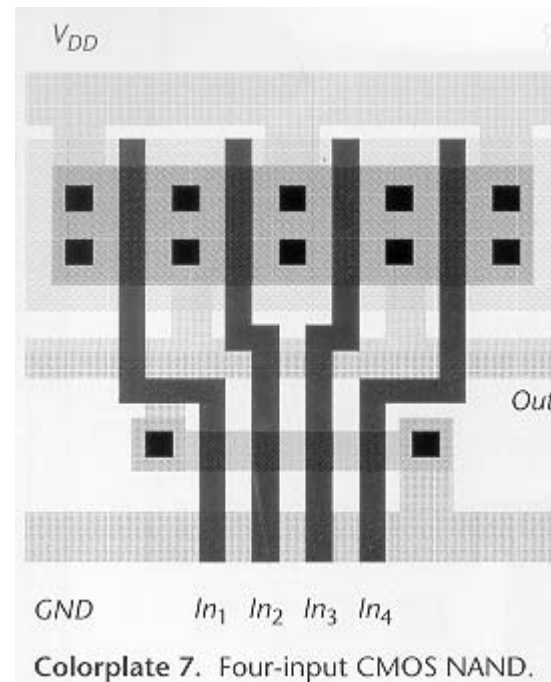
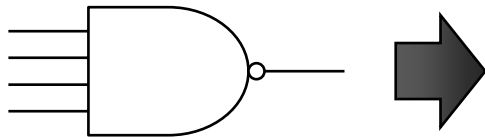


## 2. FLOT DE CONCEPTION - Introduction

---

### Exemple de description au niveau layout

Chaque porte a été placée et routée dans le circuit. Elle est remplacée par son layout (dessin des masques)



## 2. FLOT DE CONCEPTION - Introduction

---

### Vérification du design

- A chaque niveau (comport., RTL, porte) le circuit doit être vérifié :
  - Réalise-t-il la bonne fonction ?
  - Fonctionne-t-il à la bonne vitesse ?
- Deux types de vérification (complémentaires) :
  - Dynamique :
    - simulation à l'aide d'un prototype logiciel du circuit
    - Emulation (à l'aide de circuits programmables par exemple)
  - Statique :
    - Niveau fonctionnel : Prouver la bonne fonctionnalité du circuit (preuve formelle)
    - Niveau temporel : calculer la vitesse du circuit (timing analysis)



## 2. FLOT DE CONCEPTION - Introduction

---

### Vérification du design : la simulation

- La simulation permet de vérifier la conception : on modélise le problème, on simule, on ne réalise pas de prototype,  $\text{simuler} \neq \text{tester}$ .
- La simulation peut être très longue : plusieurs jours de CPU
- Le circuit complet peut être simulé uniquement aux niveaux de description les plus élevés (simulateurs Verilog/VHDL).
- Les sous-composants (blocs) sont simulés au niveau porte ou au niveau transistor
- La generation de stimuli et l'analyse de résultats est un problème complexe:
  - Stimuli : 0, 1 ou code C
  - L'analyse peut être visuelle (waveforms) ou automatique

## 2. FLOT DE CONCEPTION - Introduction

---

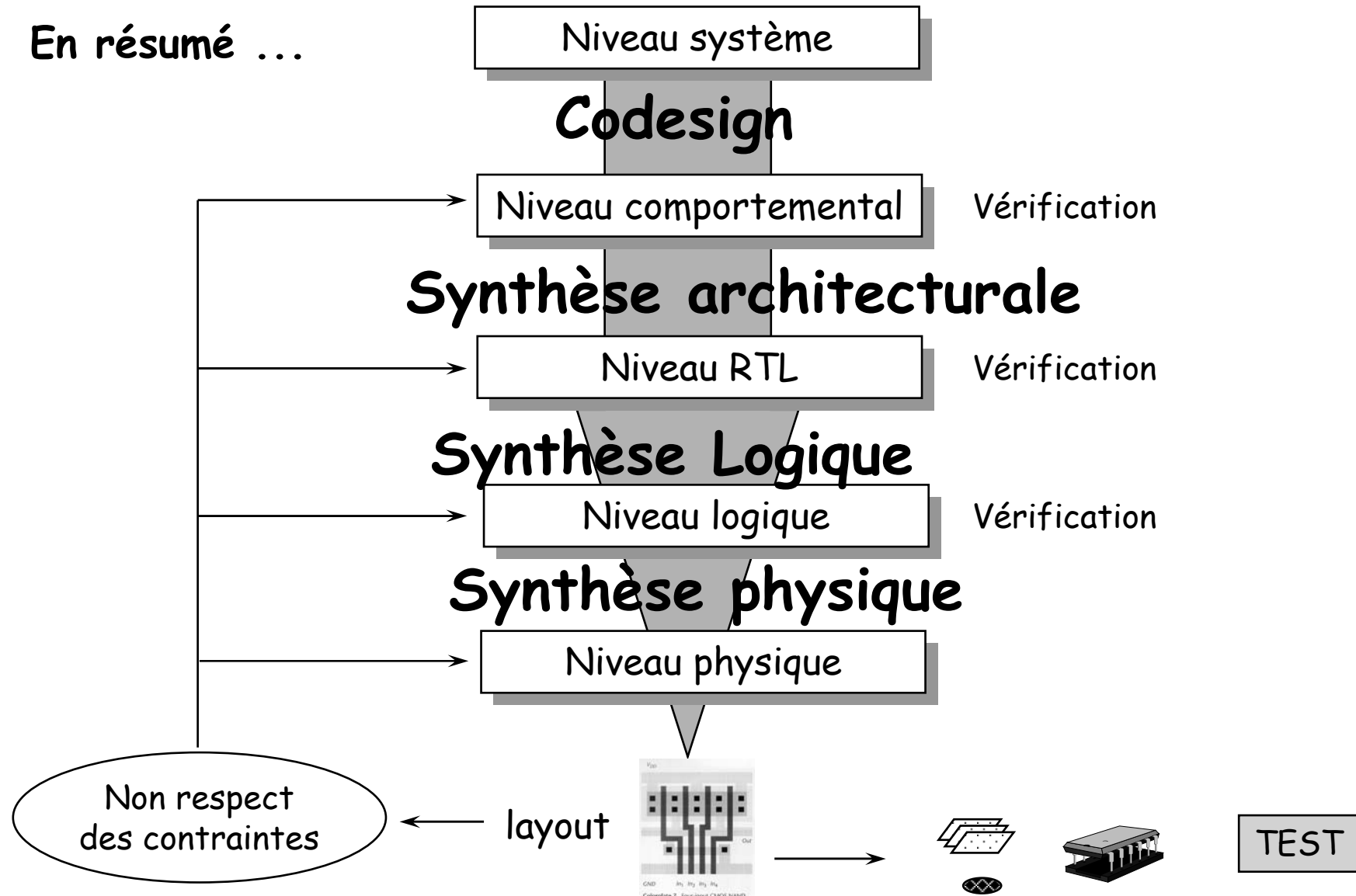
### Vérification du design : l'émulation

- ✦ Pour les systèmes complexes actuels :
  - Emulation : implémenter le circuit sur des FPGAs
  - Prototypage : la même chose sur un seul FPGA à hautes performances

**Il faut plus d'ingénieurs pour faire la vérification  
que d'ingénieurs de conception**

## 2. FLOT DE CONCEPTION - Introduction

En résumé ...



## 2. FLOT DE CONCEPTION - Introduction

---

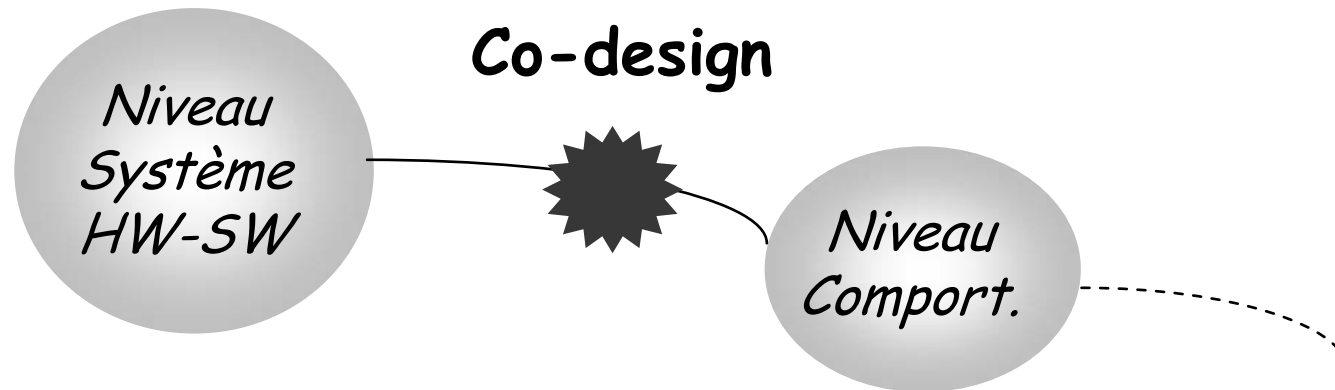
### SYNTHESE

**Génération, traitement et optimisation de descriptions intermédiaires à différents niveaux d'abstraction.**

- Synthèse Système (ou Co-Design)
- Synthèse Architecturale
- Synthèse Logique
- Synthèse Physique

## 2. FLOT DE CONCEPTION - Co-Design

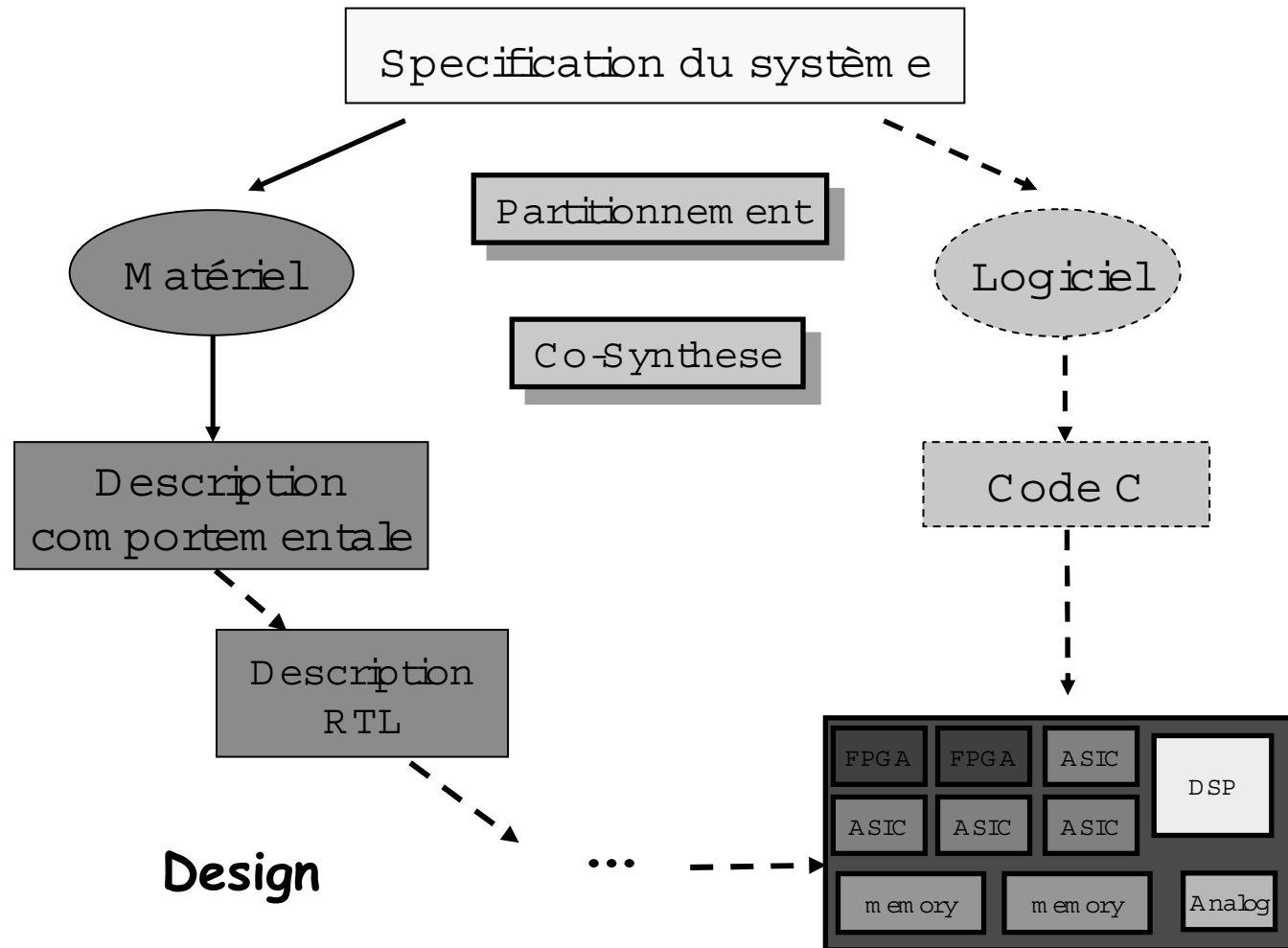
---



**Application** : systèmes très complexes nécessitant un grand nombre d'opérations de natures différentes

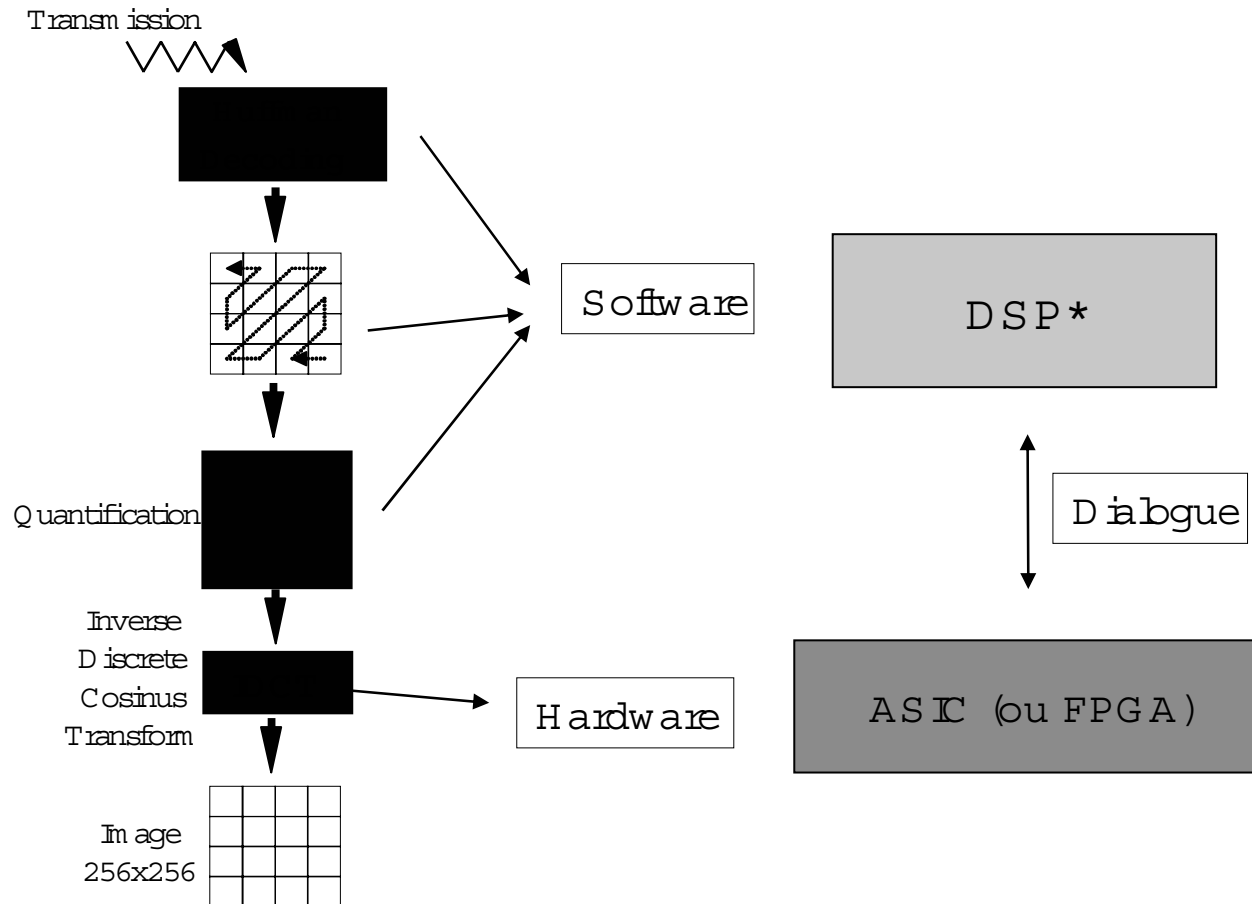
## 2. FLOT DE CONCEPTION - Co-Design

---



## 2. FLOT DE CONCEPTION - Co-Design

### Exemple : Décompression d'une image JPEG

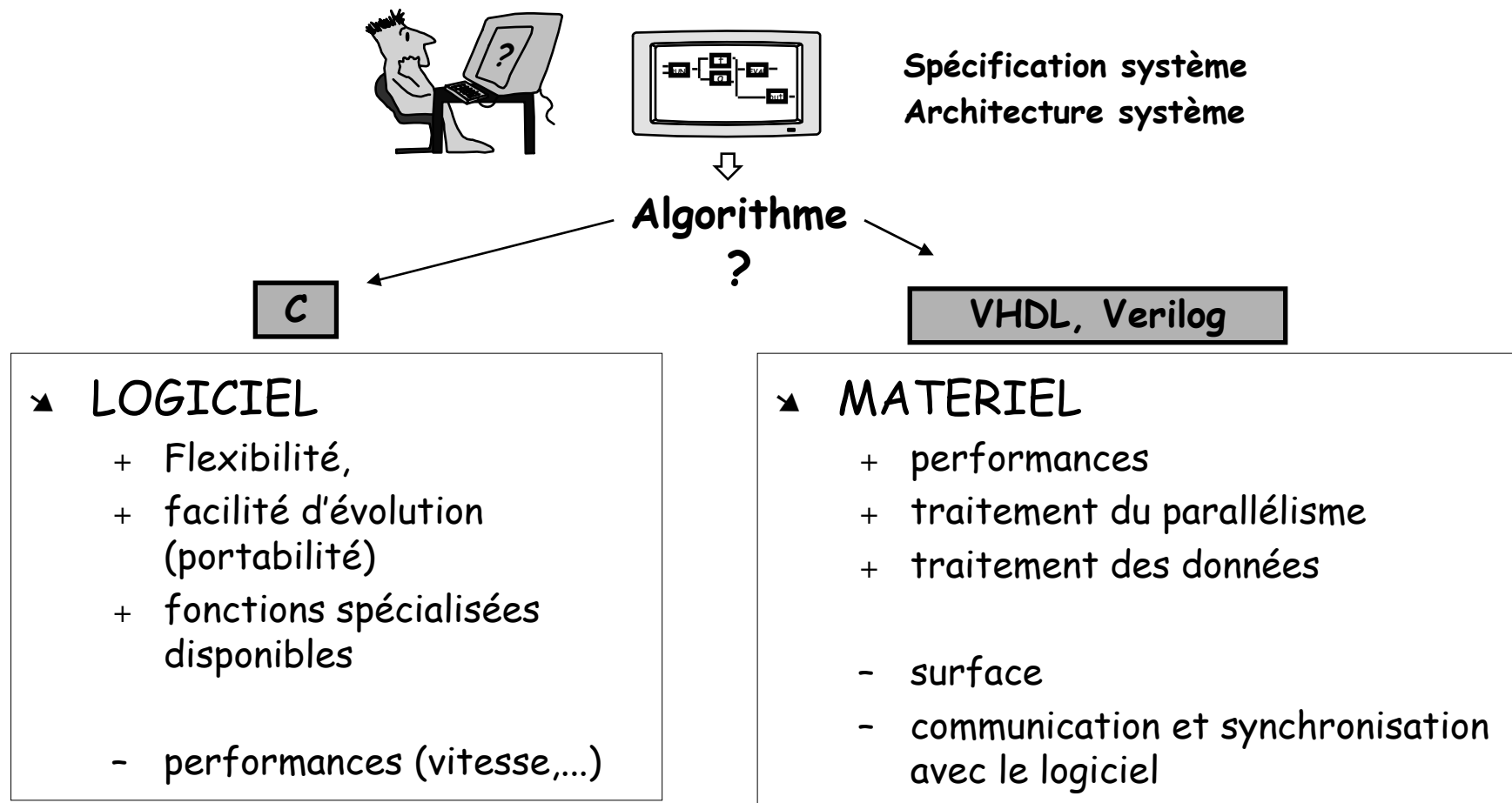


\*DSP : microcontrôleur dont la partie ALU a été modifiée pour permettre le traitement d'opérations de calcul complexes ( $\sum a_i x_i$ ). Application = traitement du signal.

## 2. FLOT DE CONCEPTION - Co-Design

---

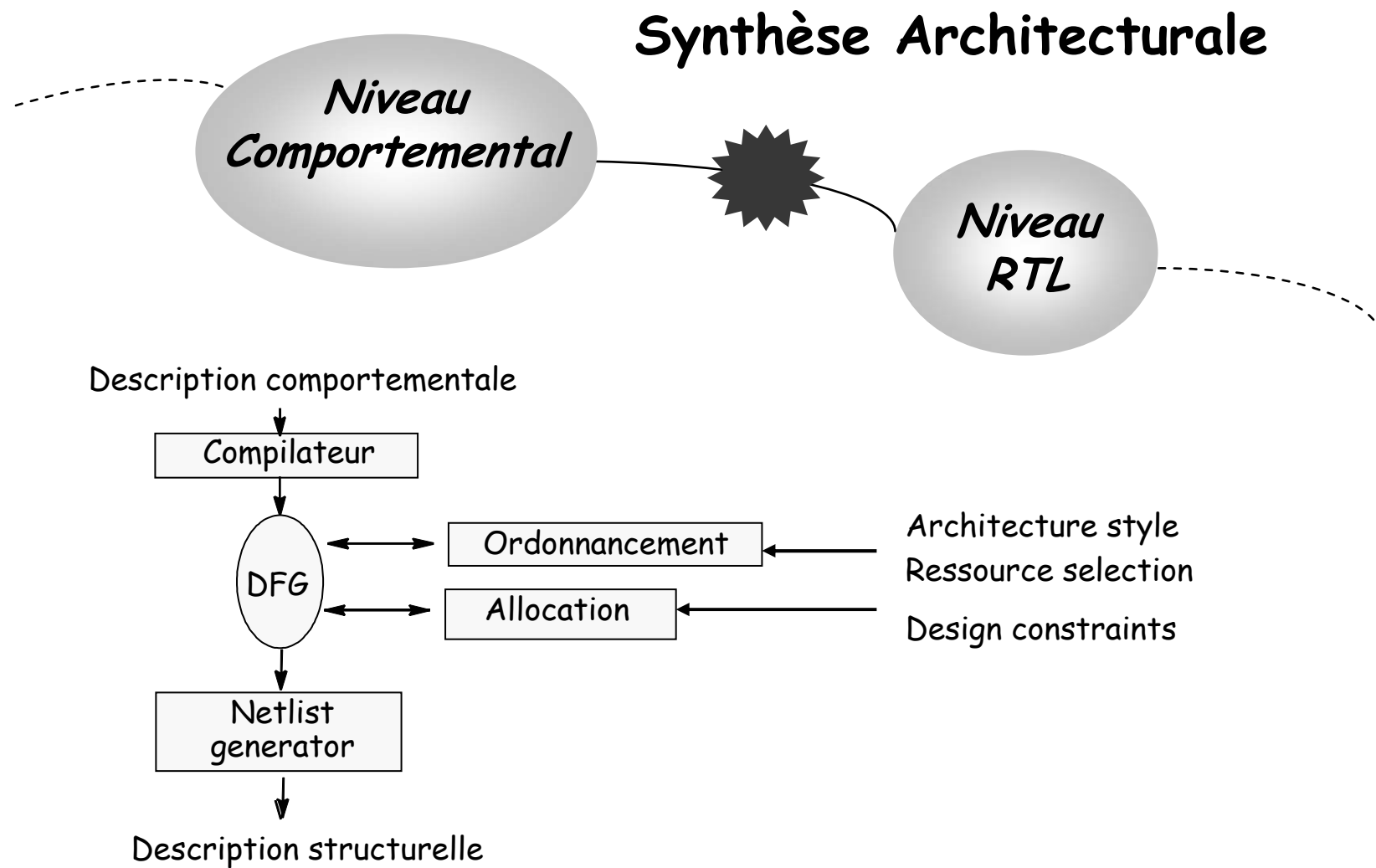
### Critères de Partitionnement





## 2. FLOT DE CONCEPTION - Synthèse architecturale

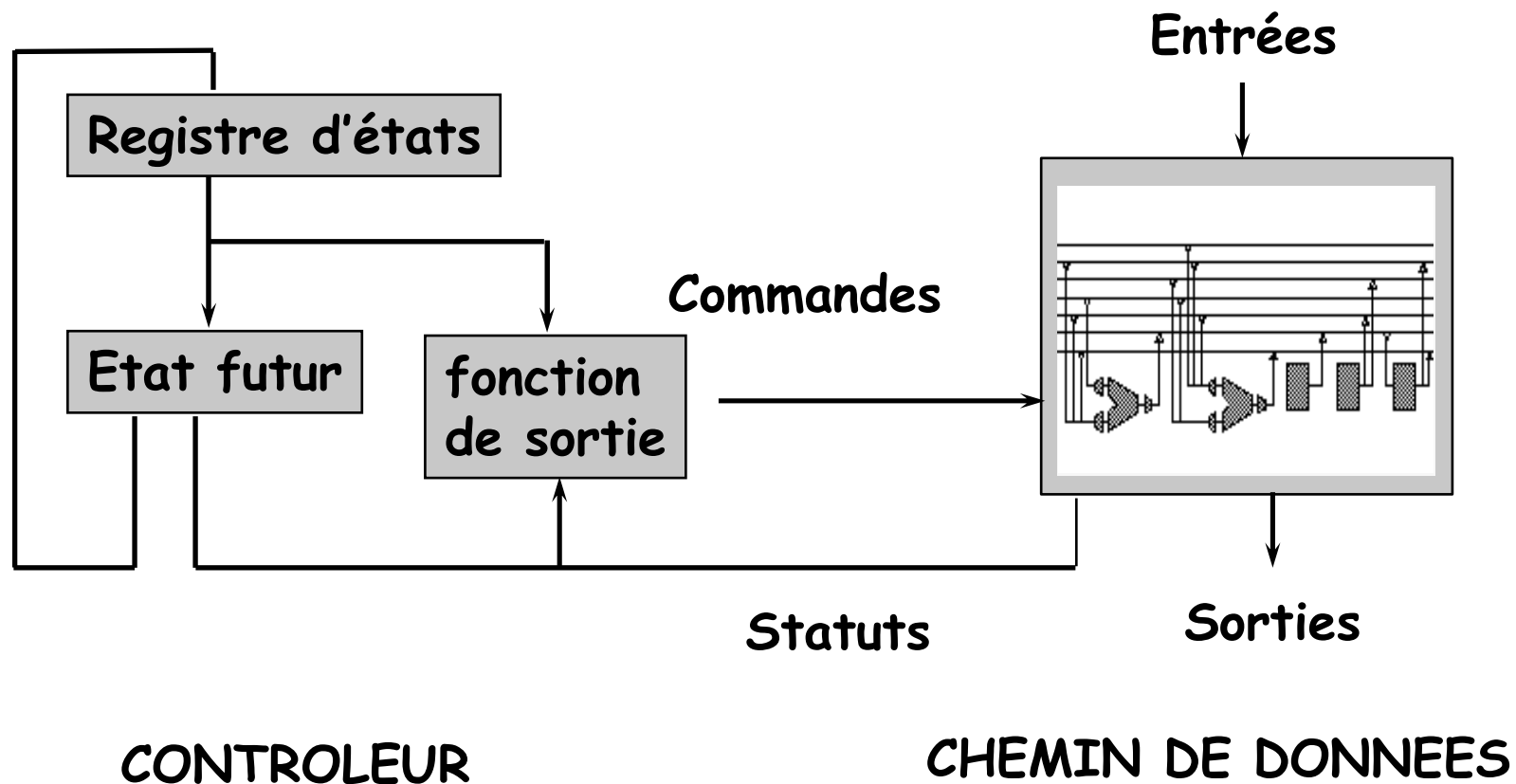
---



## 2. FLOT DE CONCEPTION - Synthèse architecturale

---

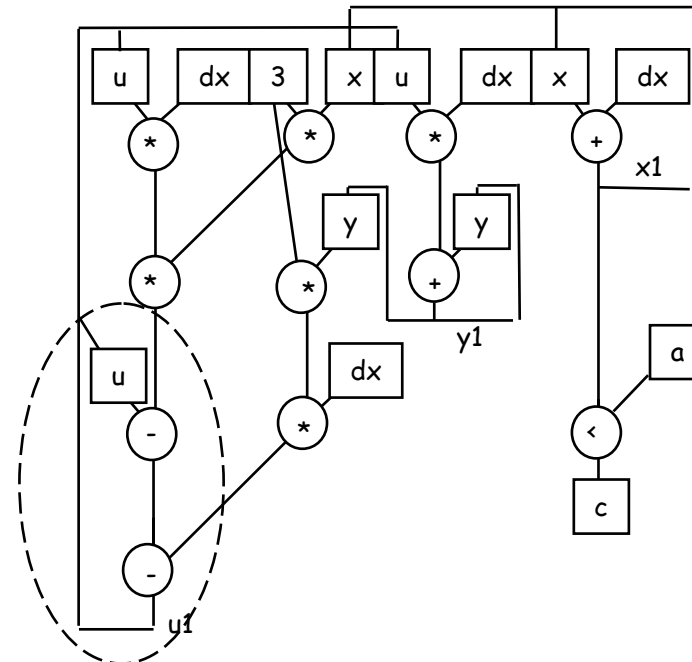
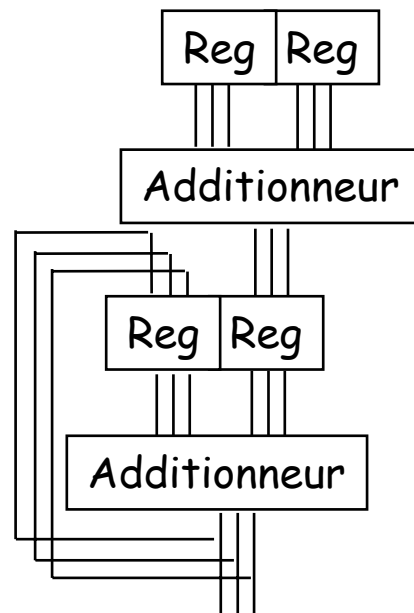
### Architecture type



## 2. FLOT DE CONCEPTION - Synthèse architecturale

```
While (x < a) do  
  x1 := x + dx;  
  u1 := u - (3*x*u*dx) - (3*y*dx);  
  y1 := y + (u*dx);  
  x := x1;  
  u := u1;  
  y := y1;  
endwhile
```

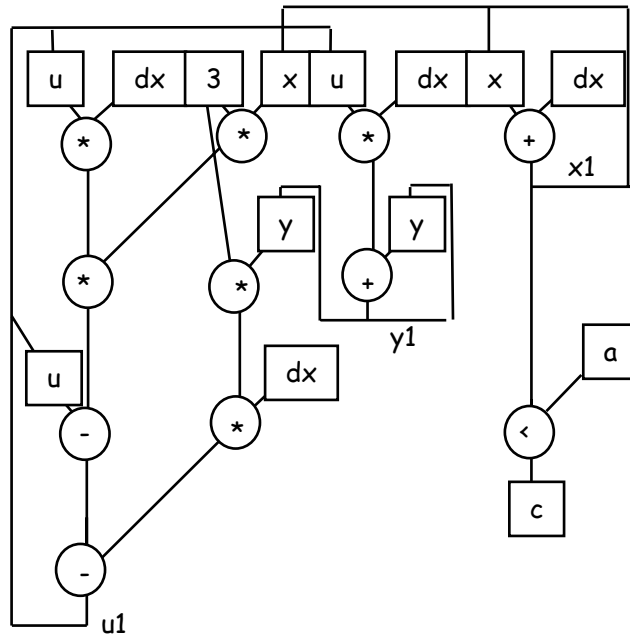
Comportement



Représentation : CDFG

Implantation (structure)

## 2. FLOT DE CONCEPTION - Synthèse architecturale



Allocation : Combien ?  
2 multiplieurs  
2 Alus  
12 registres

Assignment : Où ?  
 $x * dx$  dans Mult1

Ordonnancement : Quand ?  
5 étapes de contrôle

Pour le calcul de  $u1$  :

1 multiplieur et 1 alu  $\Rightarrow$  7 étapes de contrôle

2 multiplieurs et 2 alu  $\Rightarrow$  4 étapes de contrôle

## 2. FLOT DE CONCEPTION - Synthèse architecturale

---

### Allocation / Assignment

- ▶ définition : L'allocation des ressources matérielles consiste à :
  - déterminer un ensemble de ressources matérielles nécessaires à l'implantation des opérations et des variables (allocation)
  - faire correspondre les opérations à des opérateurs, et les variables à des modules de mémorisation (assignment).
- ▶ objectif
  - Minimiser le nombre de ressources matérielles et la connectique (surface).
- ▶ contraintes
  - Temporelles (nombre maximum d'étapes de contrôle)

## 2. FLOT DE CONCEPTION - Synthèse architecturale

---

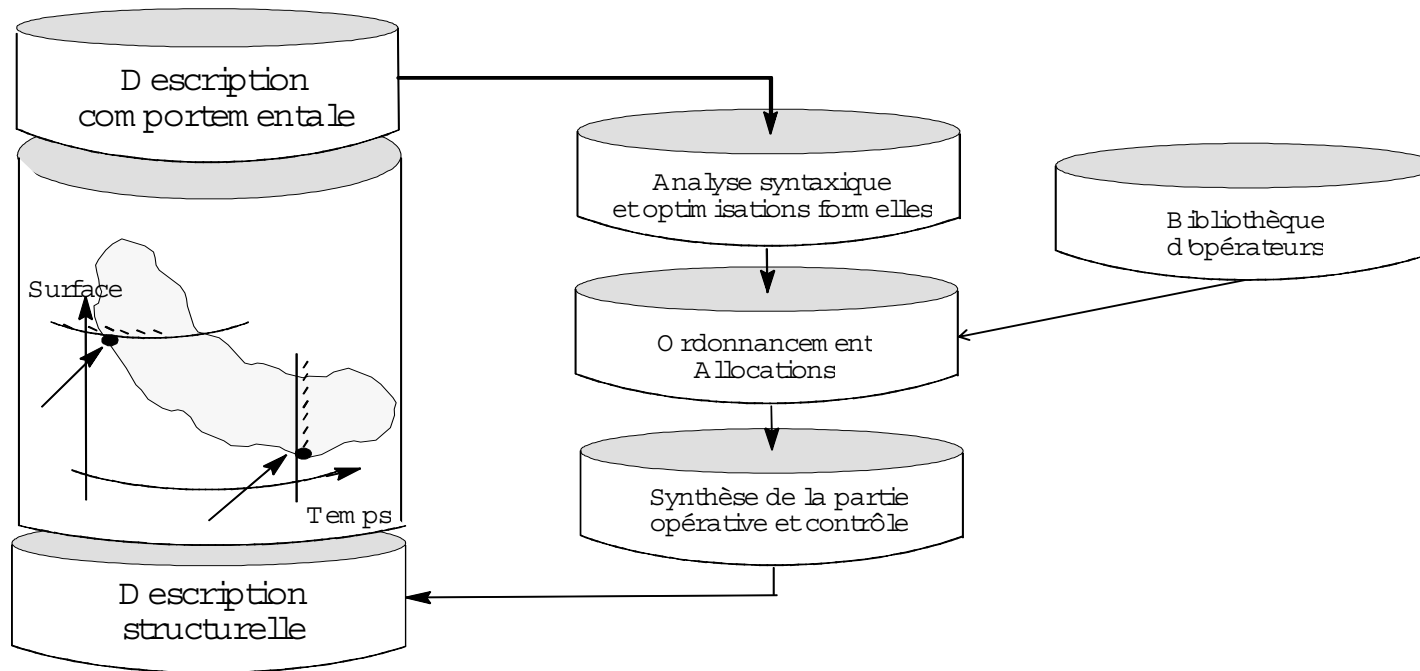
### Ordonnancement

- Définitions
  - l'ordonnancement répartit l'ensemble des opérations dans les étapes de contrôle en vérifiant un ensemble de contraintes (gestion du parallélisme)
  - l'enchaînement de ces étapes constitue le graphe d'états du contrôleur
- Objectif
  - Minimiser la latence (le nombre d'étapes de contrôle)
- Contraintes
  - de dépendance de données
  - matérielles

## 2. FLOT DE CONCEPTION - Synthèse architecturale

---

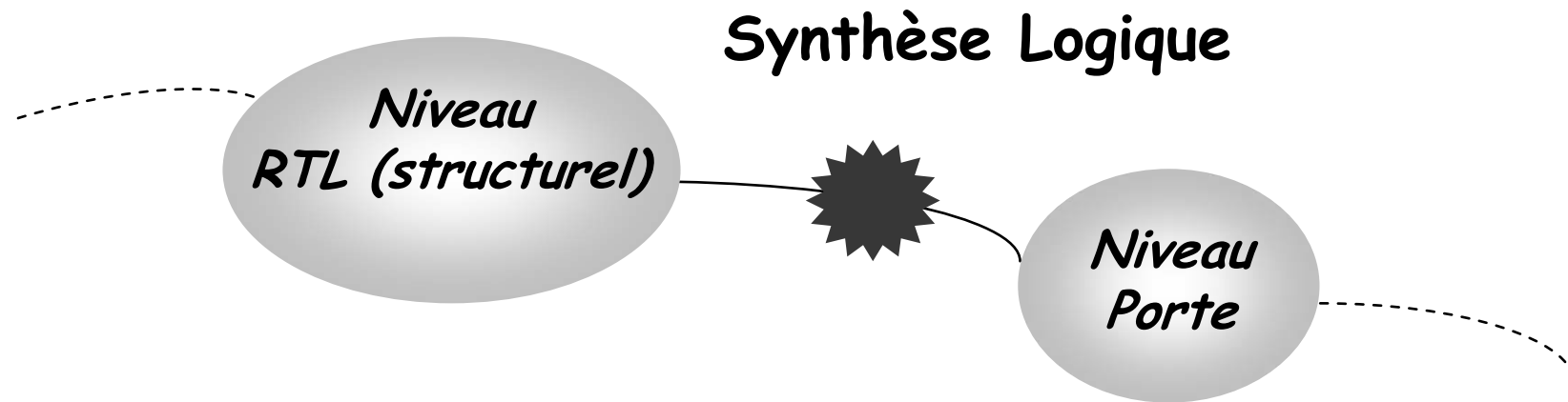
### Flot général



- Temps = Nbre de pas de contrôle
- Surface = ? = Nombre d'unités fonctionnelles + nombre de registres + nombre de muxs + nombre de connexions + ...

## 2. FLOT DE CONCEPTION - Synthèse logique

---



- **Synthèse Logique : 2 étapes**
  - **Optimisation des équations logiques**
  - **Assignation Technologique** (Technology Mapping) vers une cible (par exemple, une bibliothèque de cellules pré-caractérisés)
- ⑤ **Objectif** : minimiser une fonction de coût liée à la technologie (surface, vitesse, puissance) et calculée à partir de la pré-caractérisation des cellules.

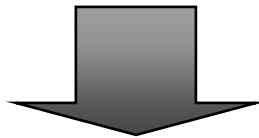


## 2. FLOT DE CONCEPTION - Synthèse logique

---

### Optimisation logique

Ensemble d'équations booléennes  
représentant la fonction du module  
à implanter (ex. contrôleur)



Ensemble d'équations  
booléennes minimisées

## 2. FLOT DE CONCEPTION - Synthèse logique

---

### Méthodes élémentaires d'optimisation logique

#### Base première complète

- Karnaugh
- Mc Cluskey
- Consensus (Tison)

#### Base minimale

- Résolution algébrique
- Table de choix (heuristique)
- Branch & bound

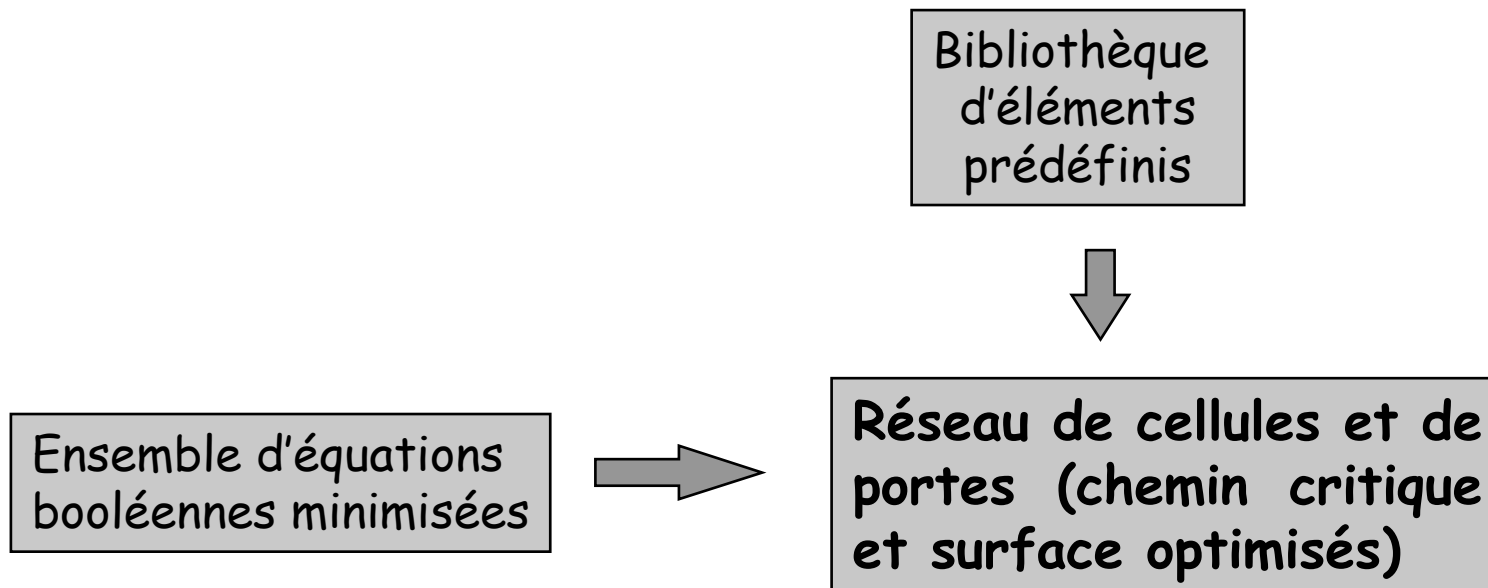
### Méthodes modernes d'optimisation logique

- "Grosses" fonctions (20 variables 500 monômes)
- Méthodes heuristiques
- Recherche d'une base première irrédundante
- Ne nécessitent pas la base première complète
- Basées sur le test d'inclusion
- Nécessite des méthodes de preuve de tautologie efficaces (BDD, Cube de position, ...)

## 2. FLOT DE CONCEPTION - Synthèse logique

---

### Décomposition technologique (Technologie Mapping)



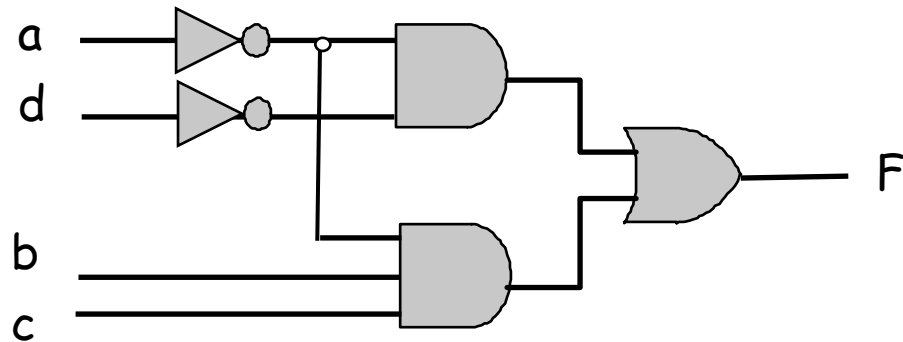
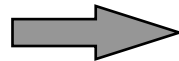
## 2. FLOT DE CONCEPTION - Synthèse logique

---

### Exemple 1

Bibliothèque complète: AND, NAND, OR, NOR, NOT, ...

Fonction :  
 $F = a' d' + a' b c$



## 2. FLOT DE CONCEPTION - Synthèse logique

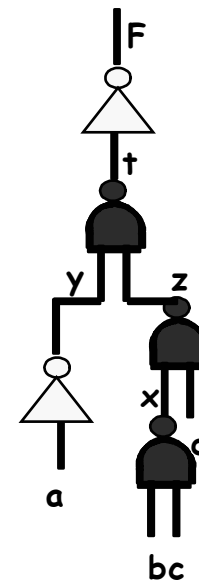
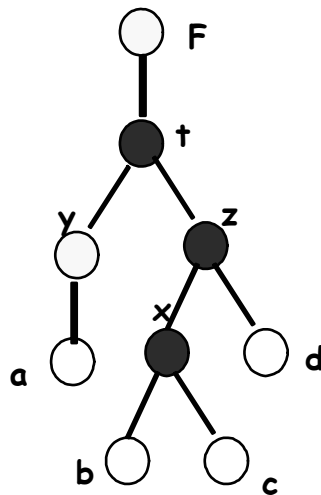
---

### Exemple 2

Bibliothèque incomplète : décomposition en NAND2 et NOT

Soit  $F = a'd' + a'bc = a' \cdot (d' + bc) = [(a' \cdot (d \cdot (bc)'))']$

$F = \text{NOT} (\text{NAND2} (\text{NOT} (a), \text{NAND2}(d, \text{NAND2}(b, c))))$



## 2. FLOT DE CONCEPTION - Synthèse logique

---

### Simulation logico-temporelle

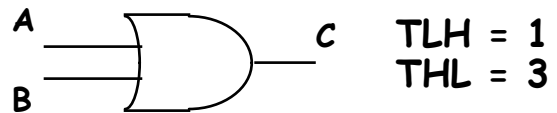
- ▶ Modélisation des états des noeuds : 0 1 X
- ▶ Modélisation des retards à partir de la précaractérisation des cellules
- ▶ Simulation événementielle ("EVENT DRIVEN") à partir des stimuli appliqués sur les entrées du circuit :
  - évènement E = changement d'état d'un nœud
  - E = (DATE, LIEU, TYPE DE TRANSITION)
  - échéancier : gestion des évènements (*"ROUE TEMPORELLE"*)
    - propagation des évènements
    - filtrage d'évènements

## 2. FLOT DE CONCEPTION - Synthèse logique

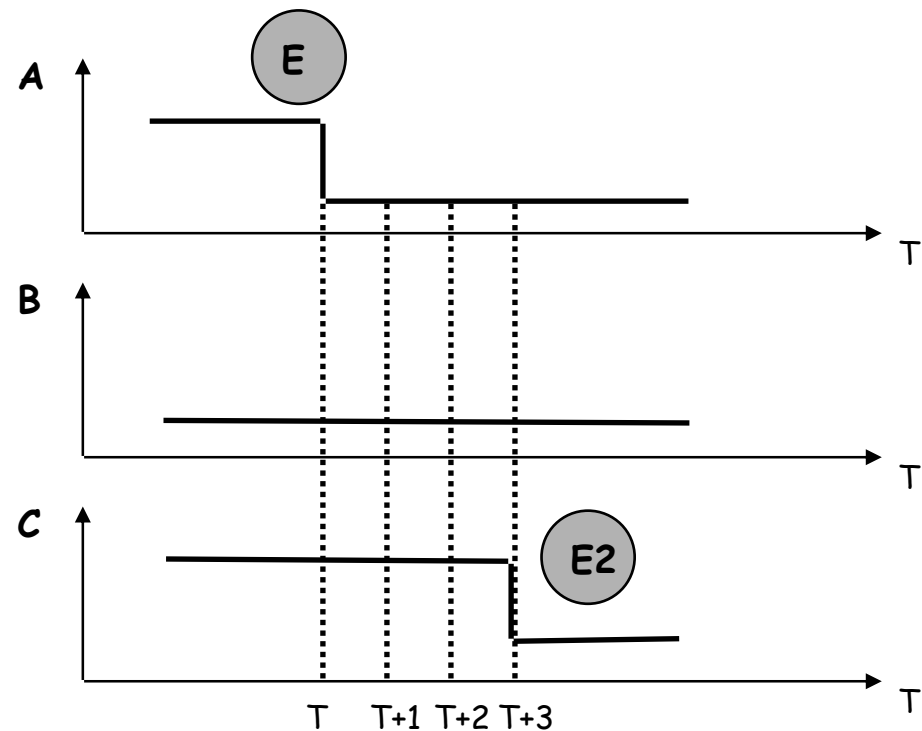
---

### Simulation logico-temporelle

#### Exemple

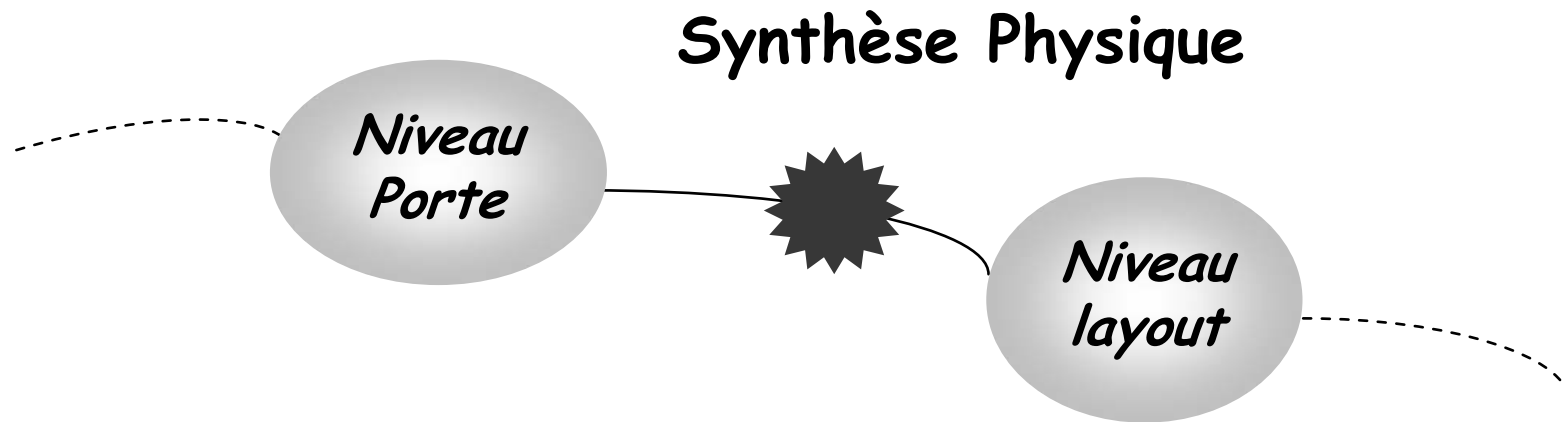


TLH = 1  
THL = 3



## 2. FLOT DE CONCEPTION - Synthèse physique

---



- ✧ **Synthèse Physique : 3 étapes**
  - **synthèses électrique** : niveau porte  $\Rightarrow$  niveau transistor
  - **synthèses topologique** : niveau transistor  $\Rightarrow$  niveau layout
  - **placement / routage** : assemblage optimisé des cellules (niveau layout)  $\Rightarrow$  layout complet du circuit



## 2. FLOT DE CONCEPTION - Synthèse physique

---

### Synthèse électrique

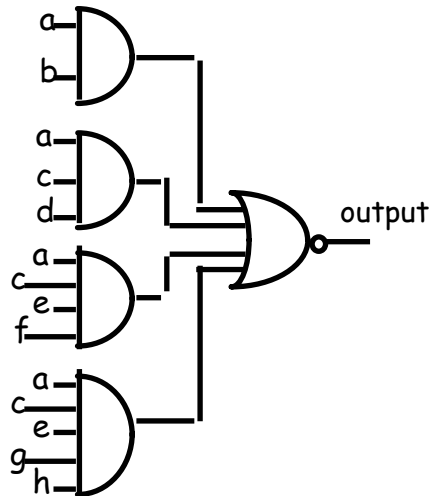


schéma logique

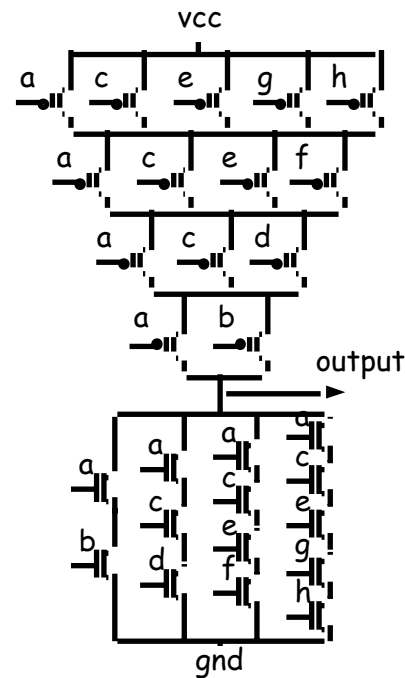
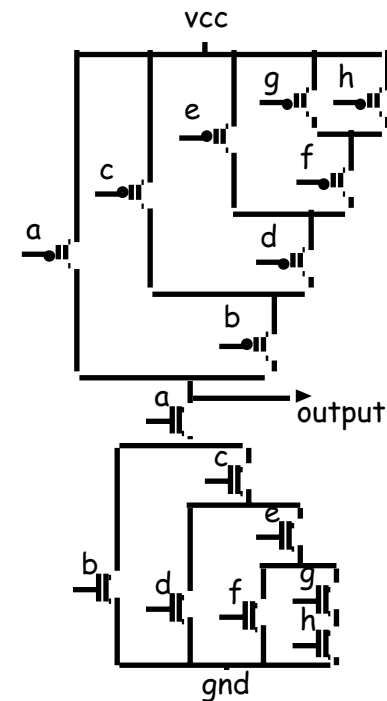


schéma électrique  
(28 transistors)



optimisation électrique  
(16 transistors)

## 2. FLOT DE CONCEPTION - Synthèse physique

---

### Simulation électrique : SPICE

#### Modélisation des composants

Exemple: model TN NMOS VTO ....

#### Entrée : netlist SPICE + Stimuli

Exemple: M1 3 2 0 0 TN W= ... L=...

C1 3 0 ...

#### Sortie : Analyse numérique

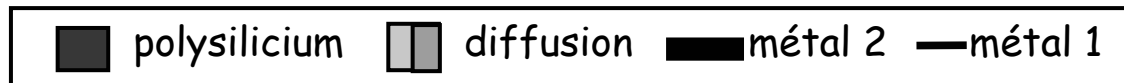
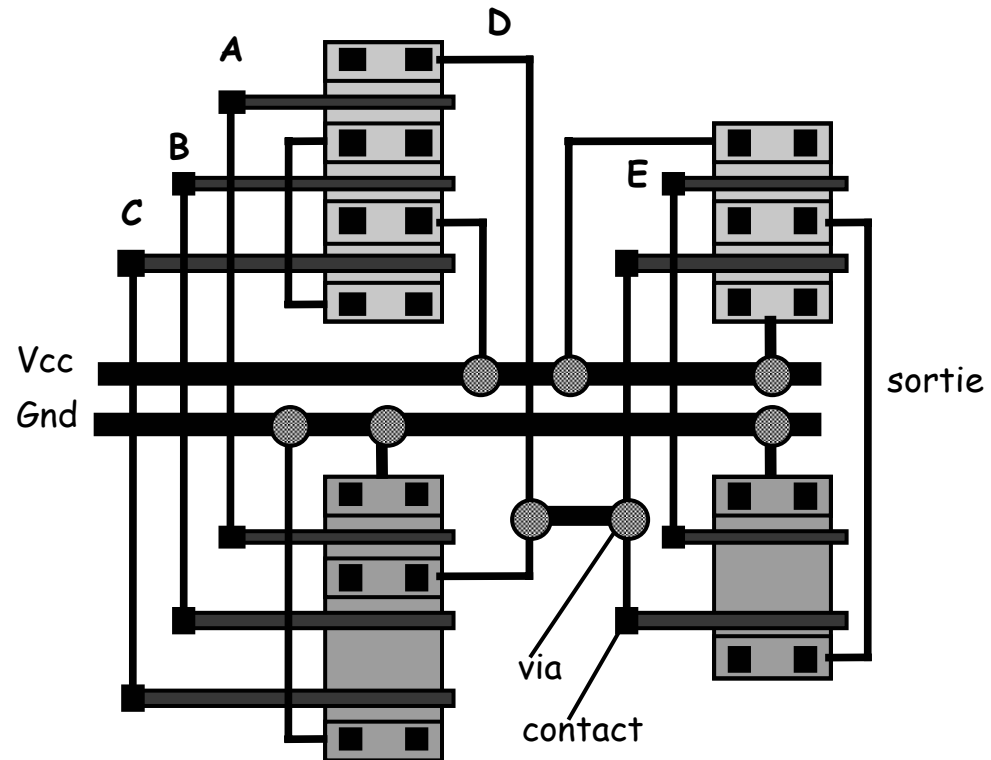
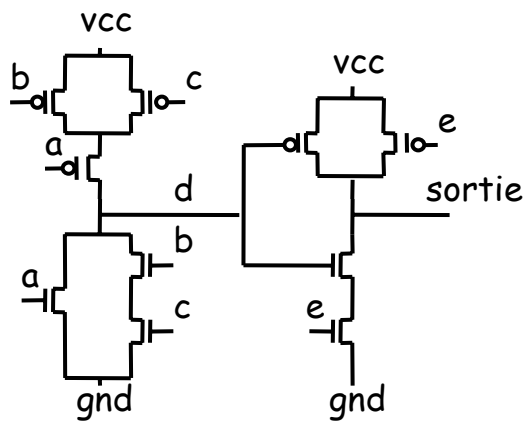
- Régime continu (caract. de transfert)
- domaine fréquentiel (Bode)
- domaine temporel

## 2. FLOT DE CONCEPTION - Synthèse physique

### Synthèse topologique

Plusieurs styles d'implantation :

- gate matrix
- linear matrix



## 2. FLOT DE CONCEPTION - Synthèse physique

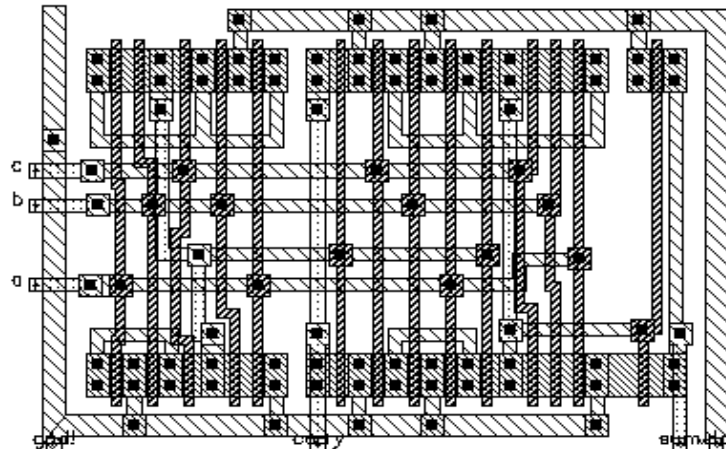
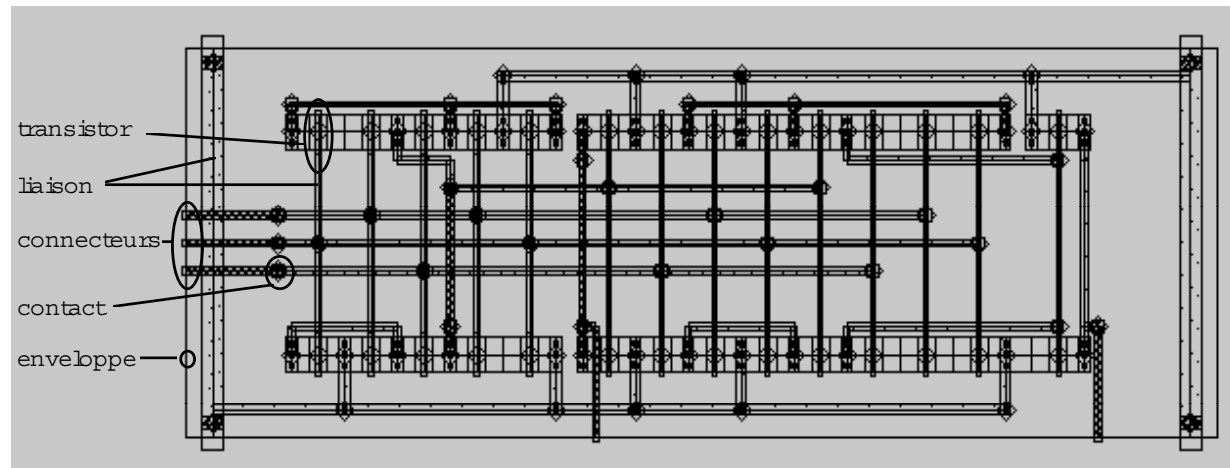
---

### Synthèse topologique

DESSIN  
SYMBOLIQUE



DESSIN DES  
MASQUES  
(LAYOUT)

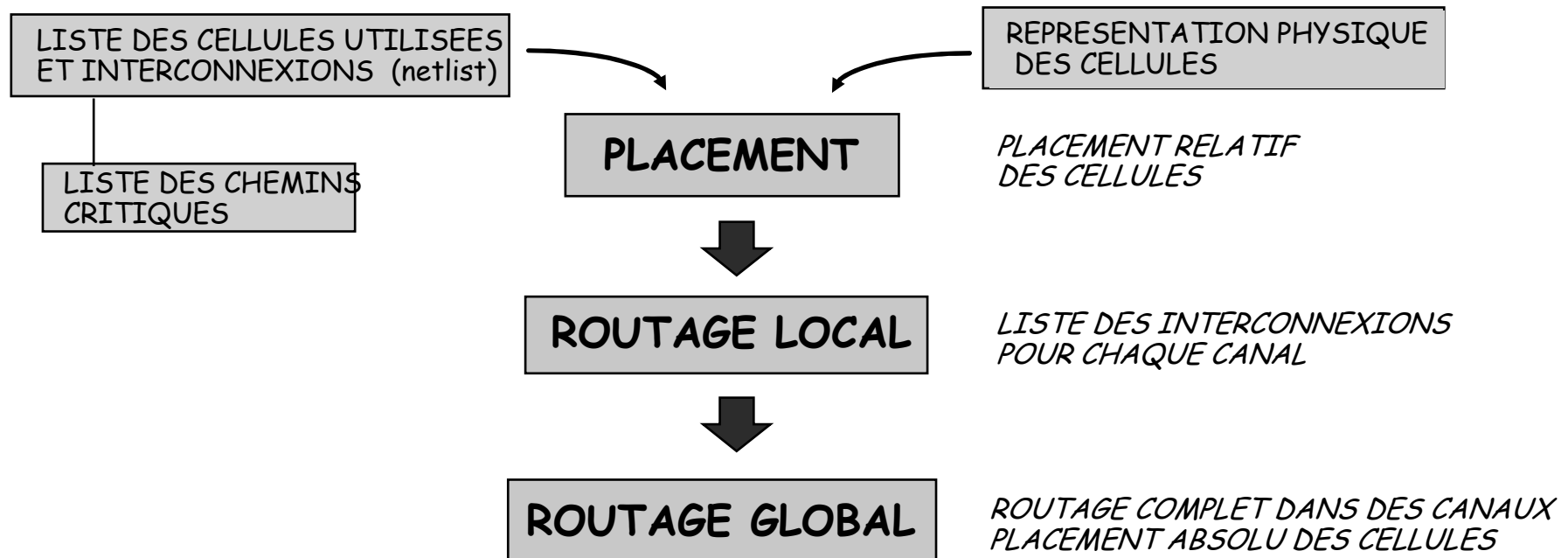


## 2. FLOT DE CONCEPTION - Synthèse physique

---

### Placement / Routage

- Chaque cellule est représentée par une "boite"
- Les boites sont placées dans le circuit et sont interconnectées



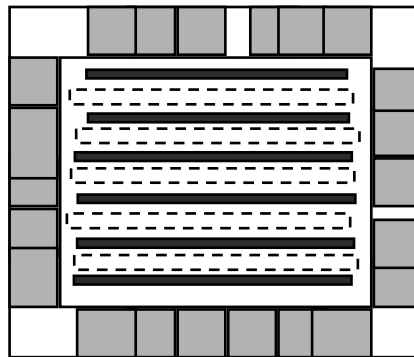
## 2. FLOT DE CONCEPTION - Synthèse physique

---

### Placement / Routage

#### CELLULES PRE-CARACTERISEES :

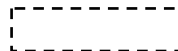
- rectangulaires de hauteur fixe
- connexions sur deux côtés ou dessus



TOPOLOGIE EN BANDES



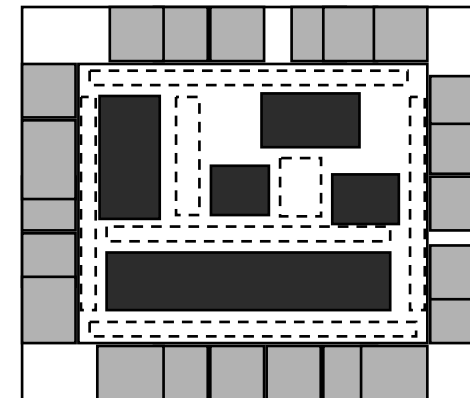
Cellules



canaux de routage

#### BLOCS COMPLEXES :

- forme variable (rectangles,..)
- connexions sur 4 côtés ou dessus



## 2. FLOT DE CONCEPTION - Synthèse physique

---

### Placement / Routage

#### ➤ Paramètres

- contraintes topologiques : nombres de bandes (pour pré-caractérisés)
- contraintes électriques : chemins critiques

#### ➤ Objectifs

- routage à 100%
- temps de calcul
- performances électriques
- interactivité

#### ➤ Méthodes

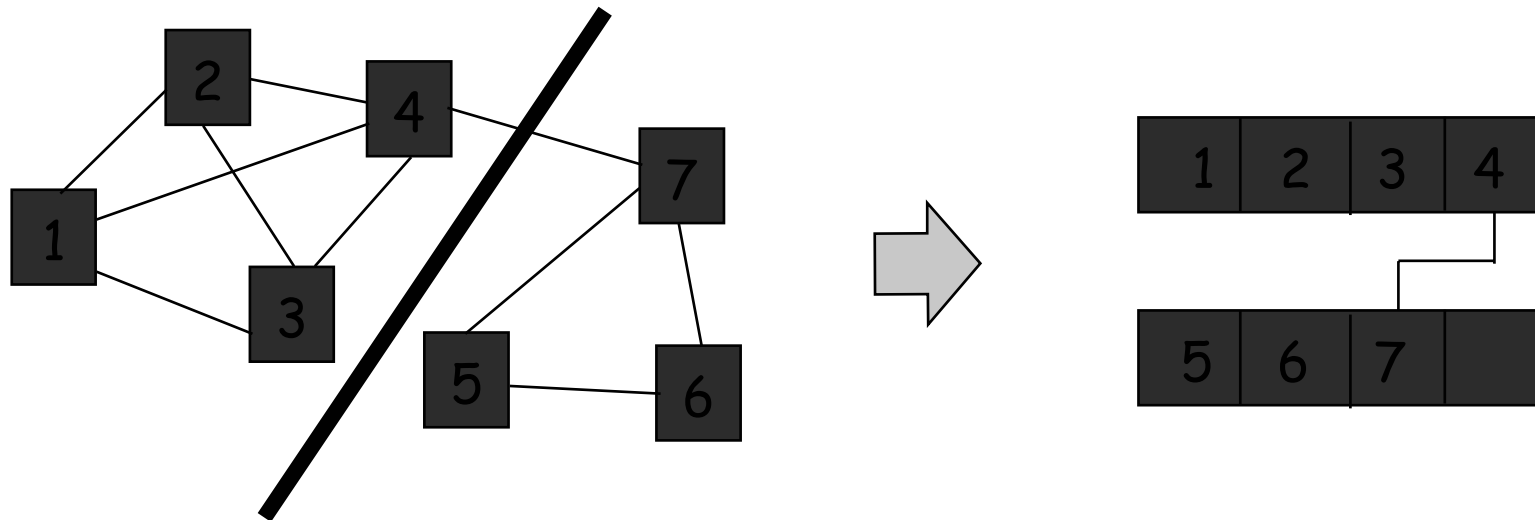
- plusieurs algorithmes utilisés suivant le type de problème à résoudre

## 2. FLOT DE CONCEPTION - Synthèse physique

---

### Algorithme de Placement : "Min-Cut"

**Principe :** Partitionnement du graphe constitué des cellules en sous-graphes ayant un nombre minimum de connexions





## 2. FLOT DE CONCEPTION - Synthèse physique

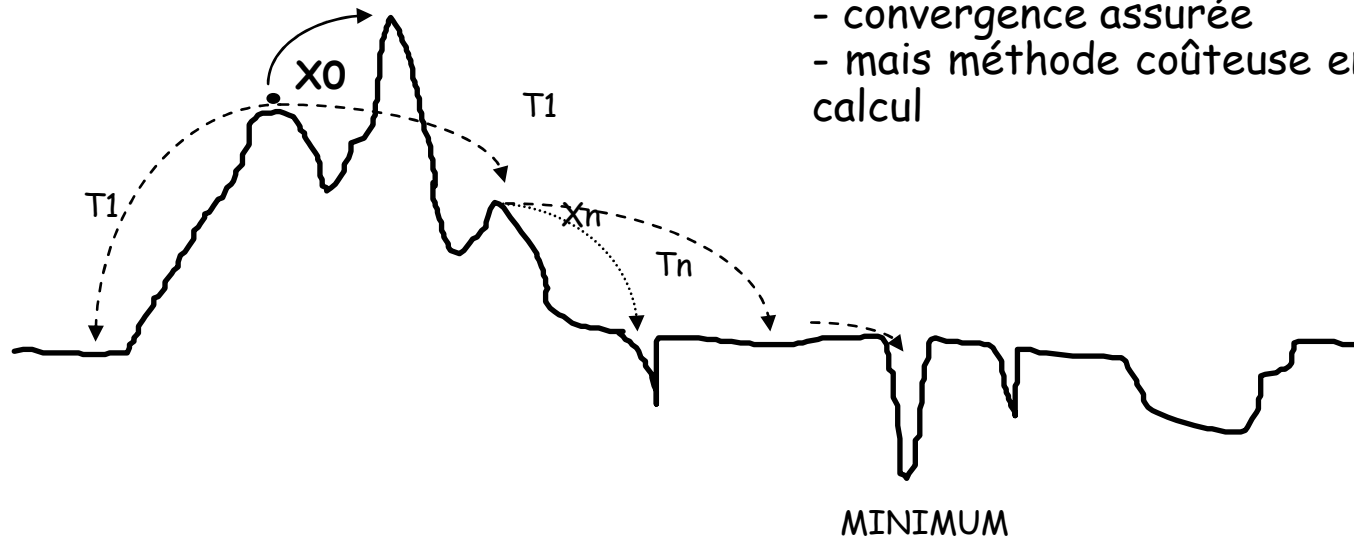
---

### Algorithm de Placement : "Recuit Simulé"

RECHERCHE DU MINIMUM :

- point de départ :  $X_0$
- "température"  $T_n < T_{n-1}$  (recuit)

- "sauts" aléatoires
- possibilité de passer par des états dégradés, ce qui permet de sortir des minimums locaux
- convergence assurée
- mais méthode coûteuse en temps de calcul



Définition d'une fonction de coût du placement : longueur des connexions (par exemple)

## 2. FLOT DE CONCEPTION - Synthèse physique

---

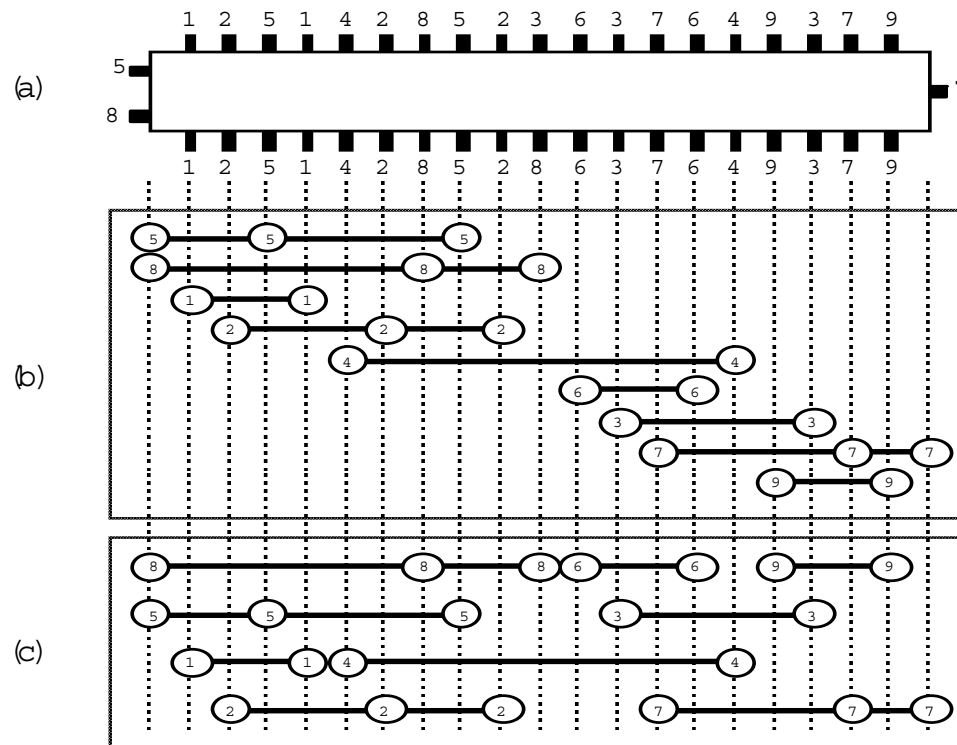
### Routage (caractéristiques)

- ✧ NIVEAUX DE COMPLEXITE :
  - Routage d 'un bloc ou d 'un circuit pre-caractérisé élémentaire : problème à une dimension (canal)
  - Routage d 'un circuit complexe : problème à 2 dimensions
  
- ✧ EVOLUTION DES TECHNOLOGIES :
  - Hier : routage sur 2 ou 3 niveaux de métal
  - Aujourd 'hui : routage sur 6 ou 7 niveaux de métal
    - moins de problèmes de surface, mais problèmes de vitesse

## 2. FLOT DE CONCEPTION - Synthèse physique

### Algorithme de Routage : "Left Edge"

Problème à une dimension : routage horizontal



**CANAL DE  
ROUTAGE**

**MISE EN ORDRE  
A PARTIR DE  
L'EXTREMITE  
GAUCHE**

**ROUTAGE  
FINAL**

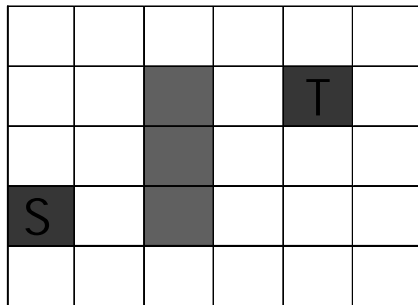
Il faudra donc 4  
niveaux de métal

## 2. FLOT DE CONCEPTION - Synthèse physique

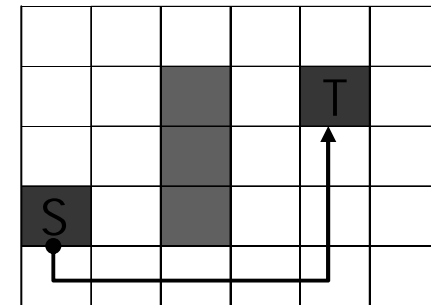
---

### Algorithme de Routage : LEE

- Problème à 2 dimensions
- Historiquement, un des premiers routeur
- Problème : temps CPU et espace mémoire



3	4	5	6	7	
2	3		7	T	
1	2		6	7	8
S	1		5	6	7
1	2	3	4	5	6



## 2. FLOT DE CONCEPTION - Synthèse physique

---

### Simulation Post-Layout

- Simulation prenant en compte les éléments "parasites" extraits du layout :  $R, C, \dots$
- Modélisation des retards à partir de la pre-caractérisation des cellules :  $T = T_0 + DT \times CL + \dots$

