



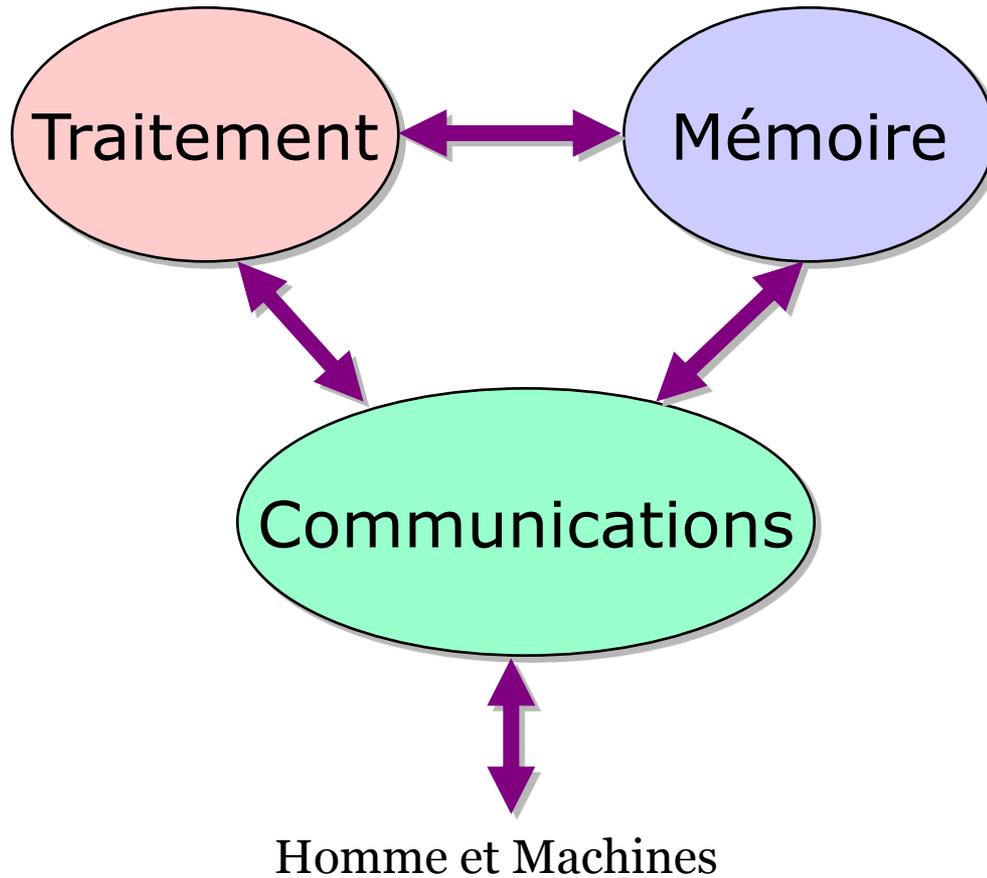
Bases d'architectures des systèmes électroniques

SE204

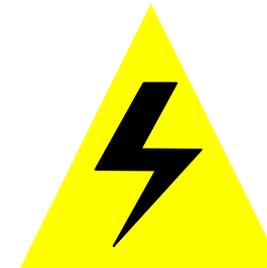
Jean-Luc Danger



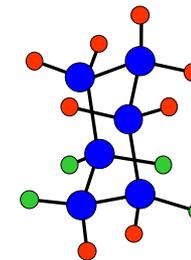
Architecture générique



Test



Energie



Intégration



Sécurité

Le types de signaux

□ Signaux de contrôle

- Horloge
 - Indispensable pour l'évolution des calculs
- Reset
 - Indispensable pour le démarrage
- Commandes
 - Générées pas les machines à états

□ Données

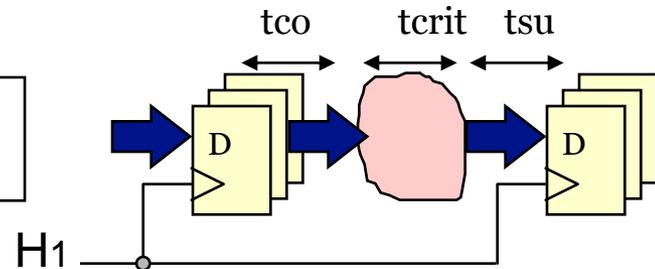
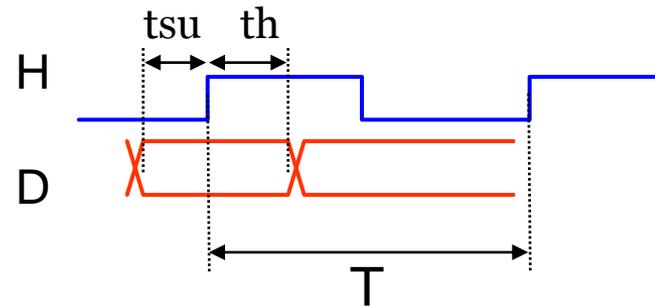
- Données effectives
- Pointeurs de données (adresses)
- Indicateurs

Rappel de la logique synchrone

1 seule horloge de période T
Calcul fiable si :

- $T > t_{co} + t_{crit} + t_{su}$
- $t_{co} > t_h$ (toujours garanti)

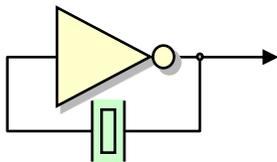
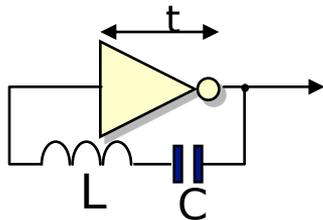
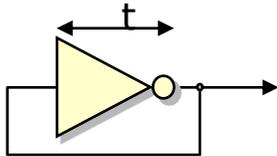
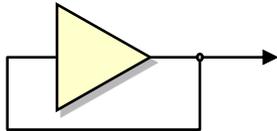
Il suffit de connaître T_{crit} !



Génération d'horloge

gain A

Déphasage □



le système oscille si $\left\{ \begin{array}{l} A = 1 \\ \varphi = 180^\circ \end{array} \right.$

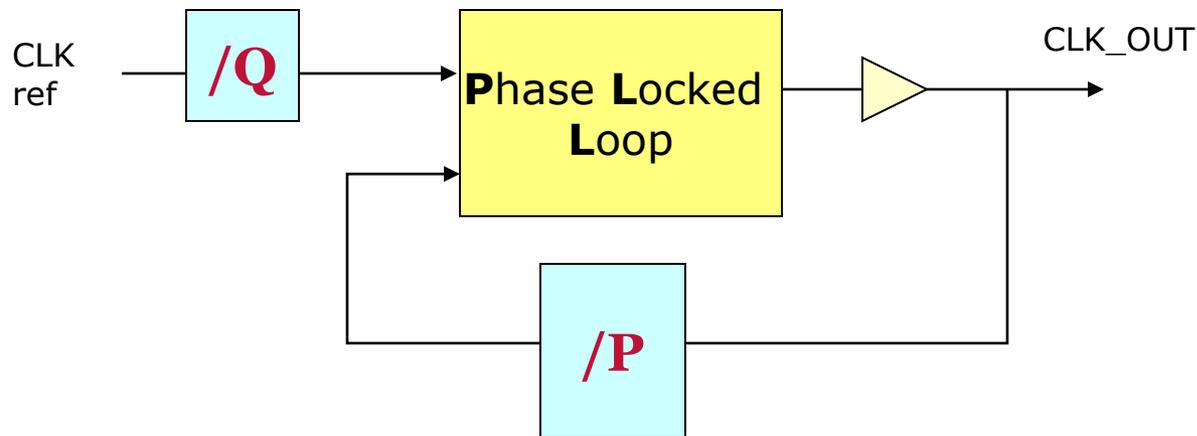
le système oscille à la fréquence $\frac{1}{2t}$

le système oscille à la fréquence $\frac{1}{2\pi\sqrt{LC}}$
(impédance nulle)

la fréquence est plus stable avec un quartz
qui a un facteur de qualité supérieur

Synthèse d'horloge

Principe : système asservi en phase



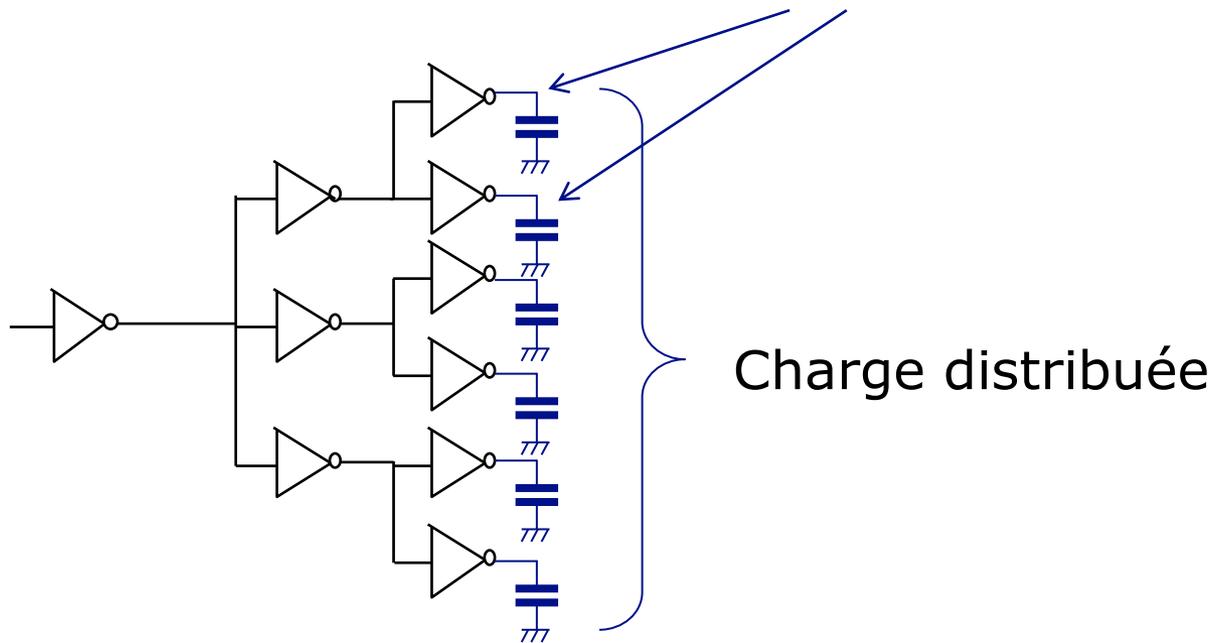
$$F_{\text{out}} = P/Q f_{\text{ref}}$$

- ⇒ compensation le "skew" d'horloge
- ⇒ multiplication de la fréquence si diviseur dans la boucle de réaction

Distribution de l'horloge

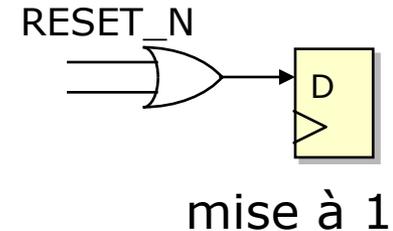
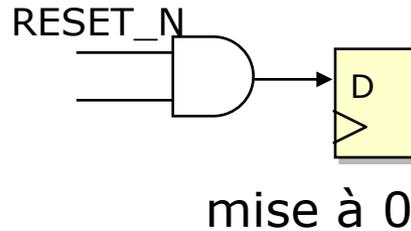
Arbre d'horloge

Signaux d'horloge équitemporels

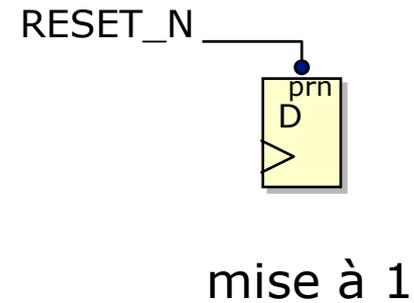
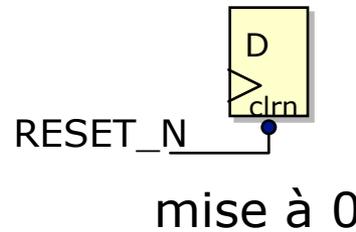


Démarrage du système

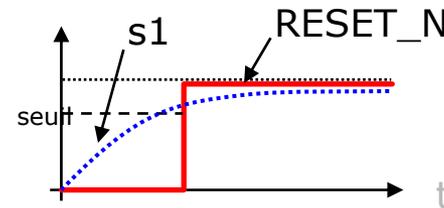
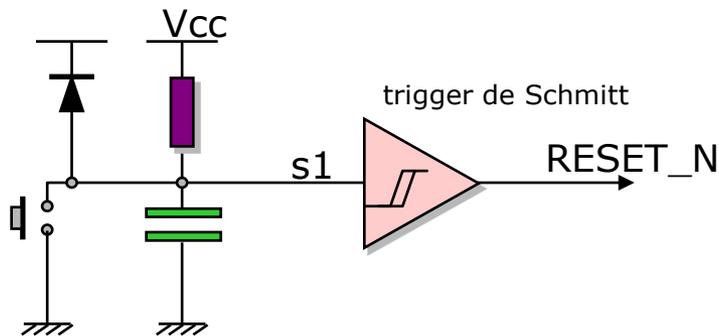
RESET synchrone :



RESET asynchrone :



Exemple de circuit de RESET :

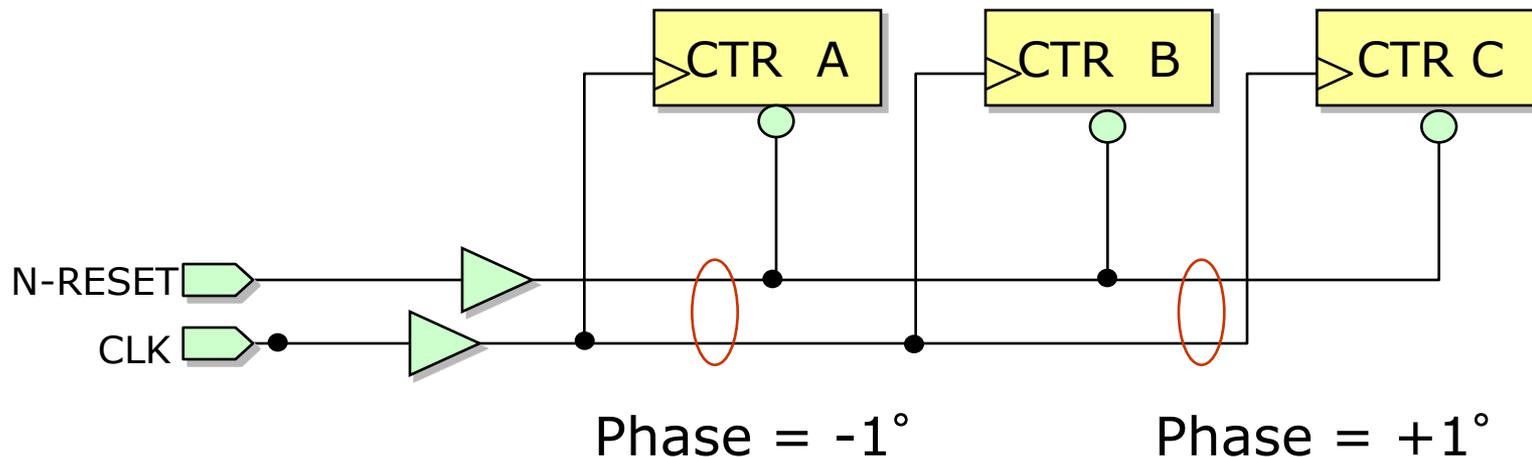


Problèmes de démarrage

❑ Oubli du Reset sur les machines à état

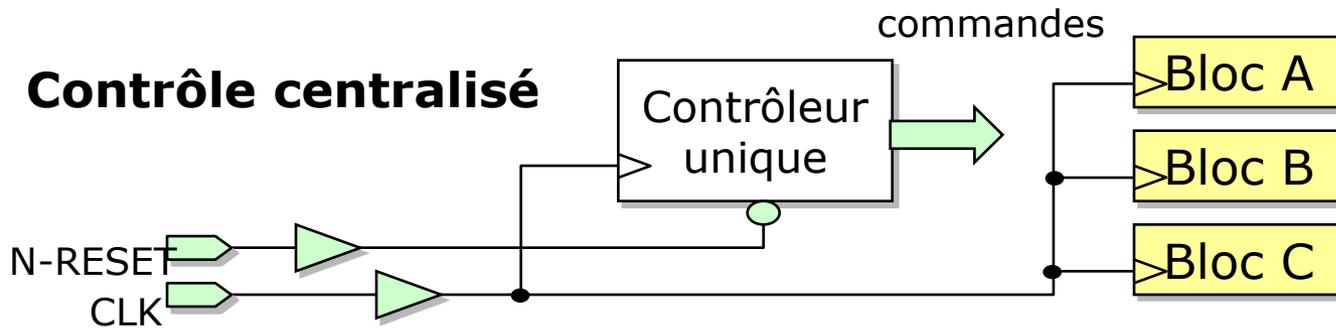
- la variable d'état est initialisée sans condition sur le Reset. La simulation ne détecte pas le problème et le circuit démarre dans un état quelconque.

❑ Les contrôleurs peuvent ne pas démarrer en même temps à cause des différences de phase clk/reset

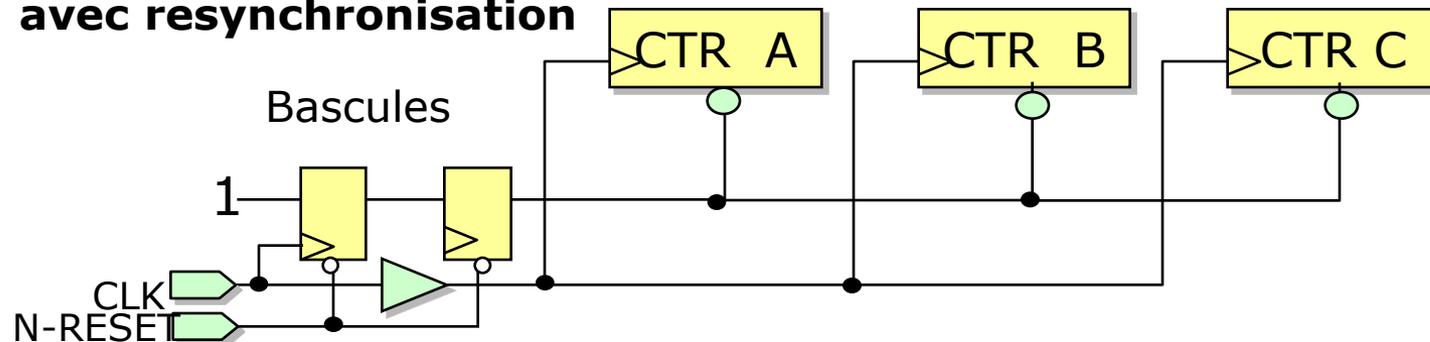


Stratégies pour bien démarrer

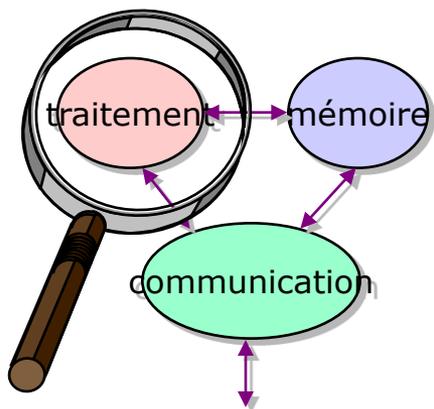
Utilisation systématique du Reset sur tous les processus de contrôle



Contrôle distribué avec resynchronisation

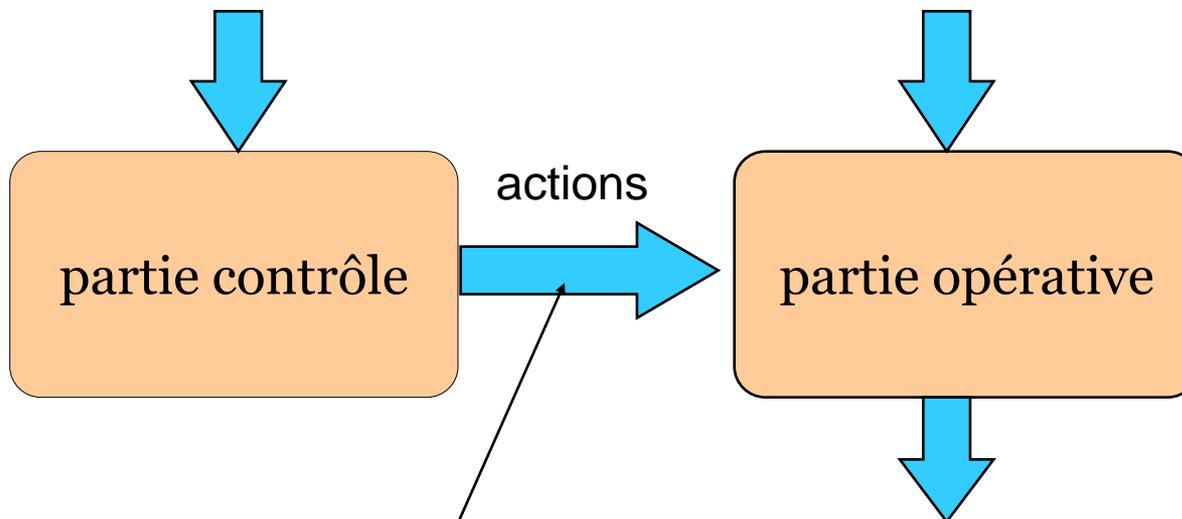


Unité de traitement



entrées : modes et indicateurs

données entrantes

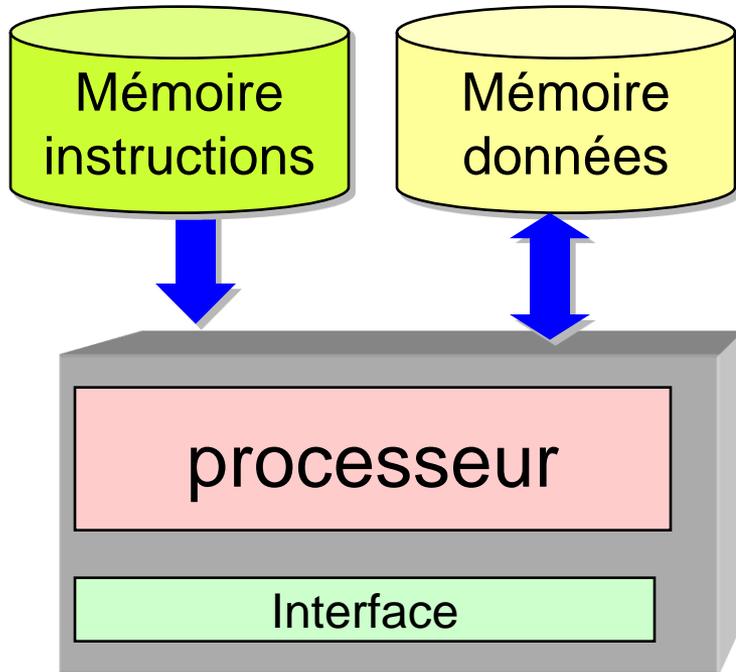


Signaux de contrôle

données sortantes

Les types de traitement

logiciel

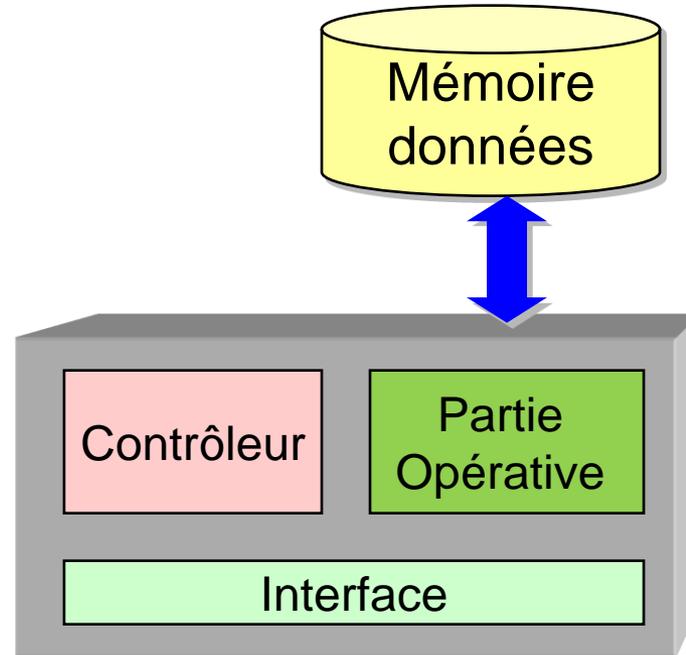


*Le contrôle peut être modifié
par les instructions :*

⇒ Flexibilité

⇒ Temps de développement

Matériel

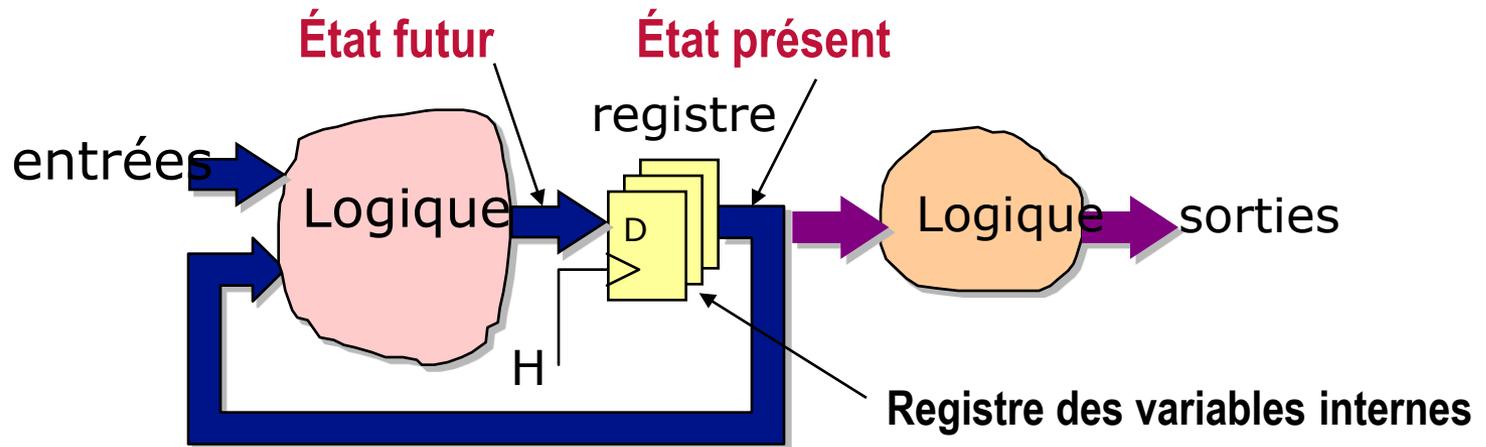


*Le contrôleur est optimisé
Plusieurs opérations
peuvent avoir lieu en même temps*

⇒ Performances

Machine à états

Génère les signaux de contrôle : Machine séquentielle synchrone à états finis



Machine à états de Moore :

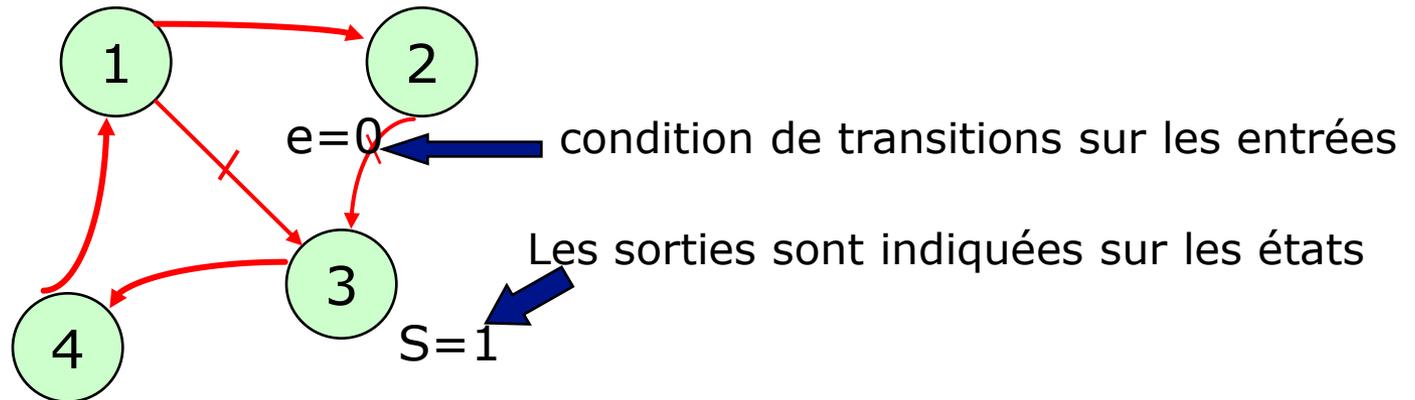
$sorties = f(\text{état})$

Machine à états de Mealy :

$sorties = f(\text{entrées}, \text{état})$

Graphe d'états

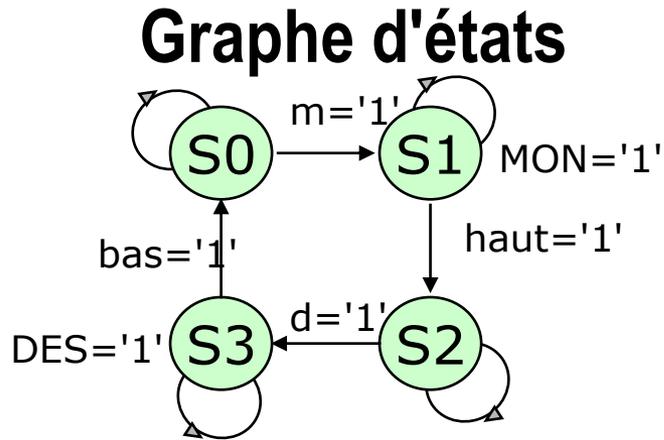
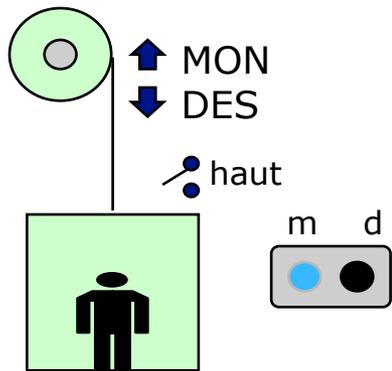
spécification d'une machine à états par graphe :



Le graphe doit être :

- complet : pour chaque état $\sum \text{conditions} = 1$
- Non contradictoire : $\sum_i \sum_{j \neq i} C_i C_j = 0$

Exemple de machine à états



Codage des états

"one hot":

1 état sur 4 bits Q3 Q2 Q1 Q0

S0=0001

S1=0010

S2=0100

S3=1000

Équations des sorties :

MON = Q1

DES = Q3

Équations des états futurs :

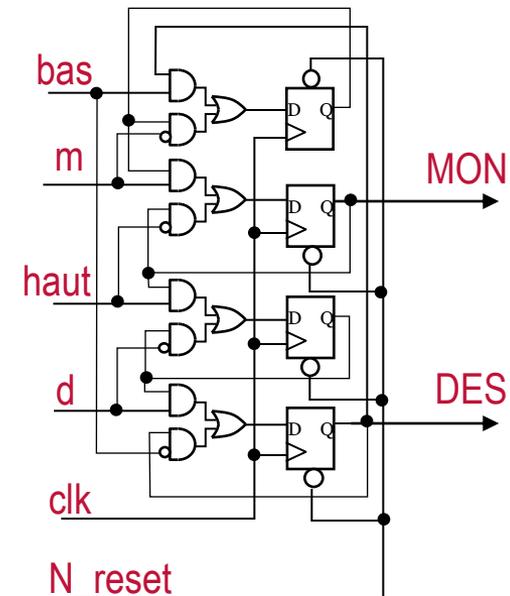
$D1 = Q0.m + Q1. /haut$

$D2 = Q1.haut + Q2. /d$

$D3 = Q2.D + Q3. /bas$

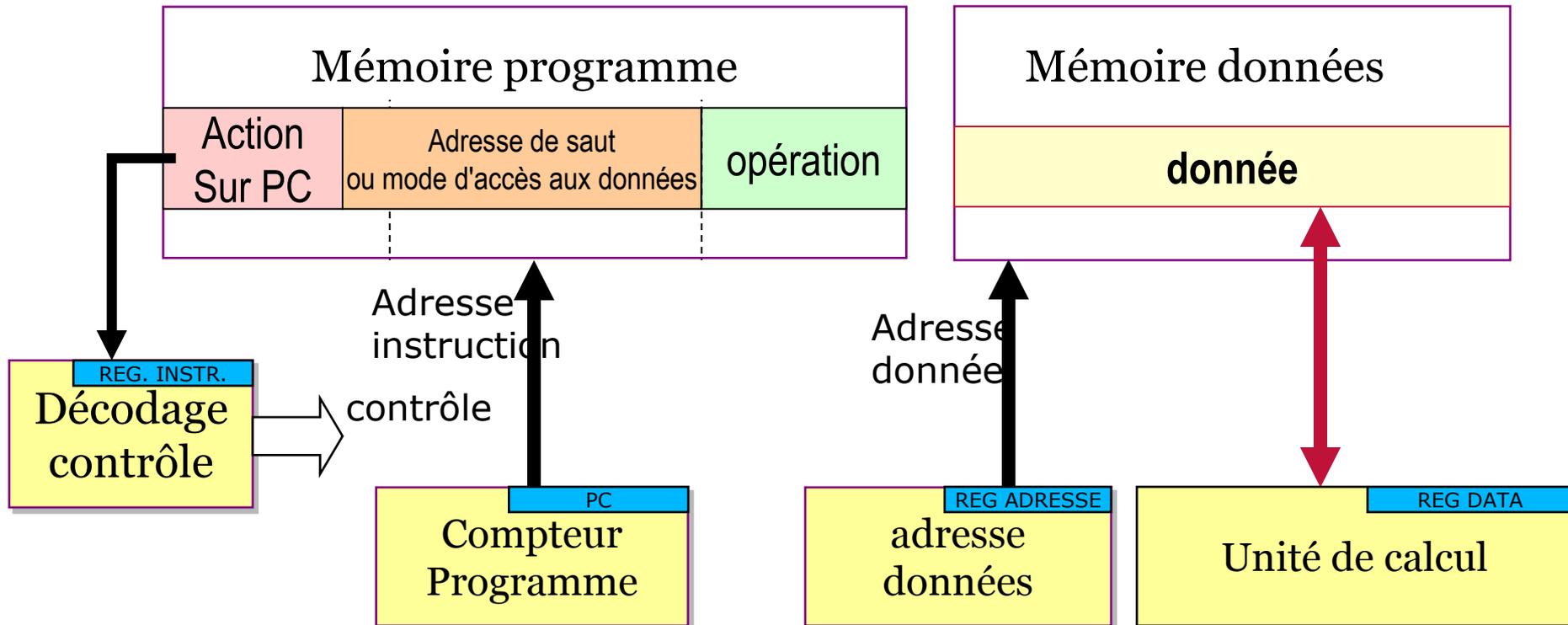
$D0 = Q3.Bas + Q0. /m$

Schéma logique

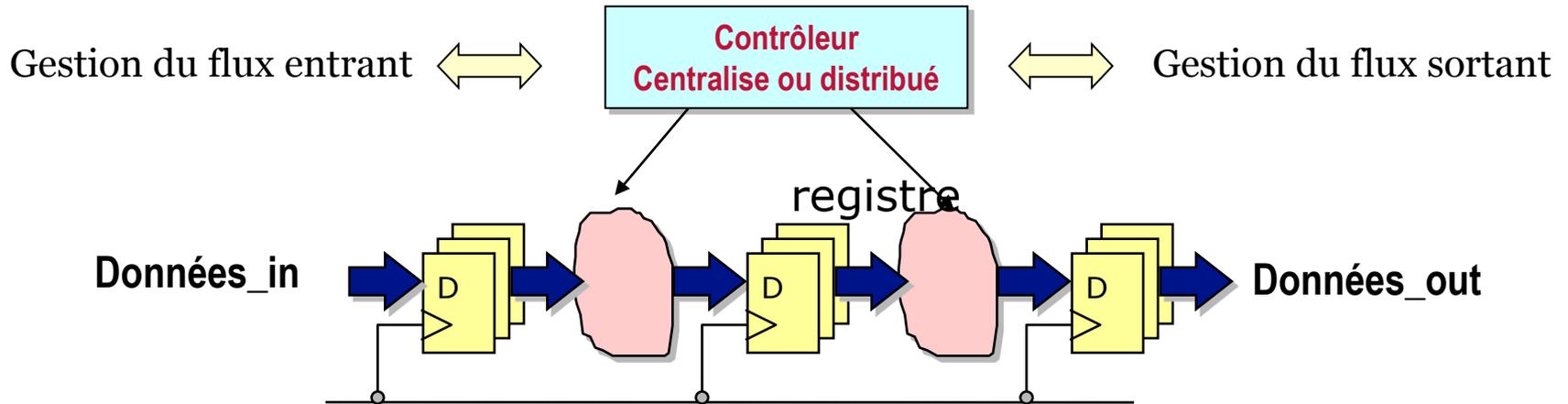


Processeur

Architecture à mémoire unique (Von Neumann) ou séparée (Harvard) :

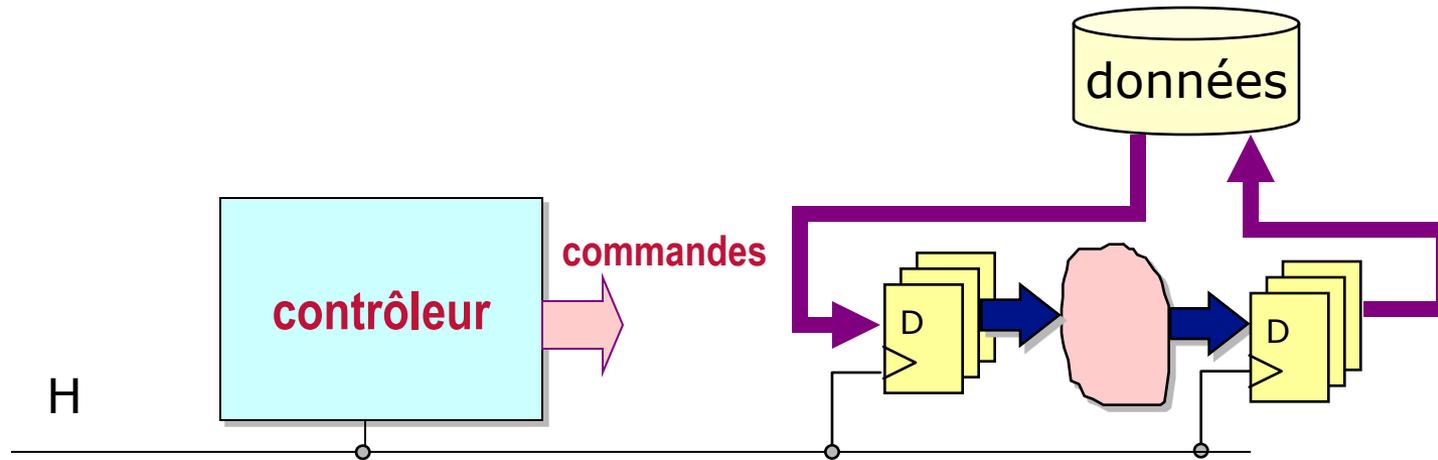


Partie opérative « data flow »



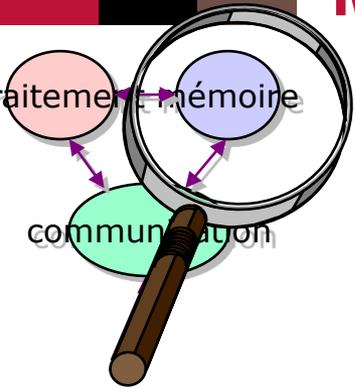
- Traitement en pipeline
- Contrôle simple

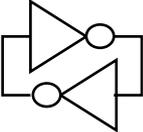
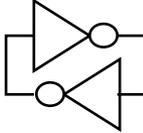
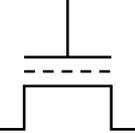
Partie opérative « Control Driven »



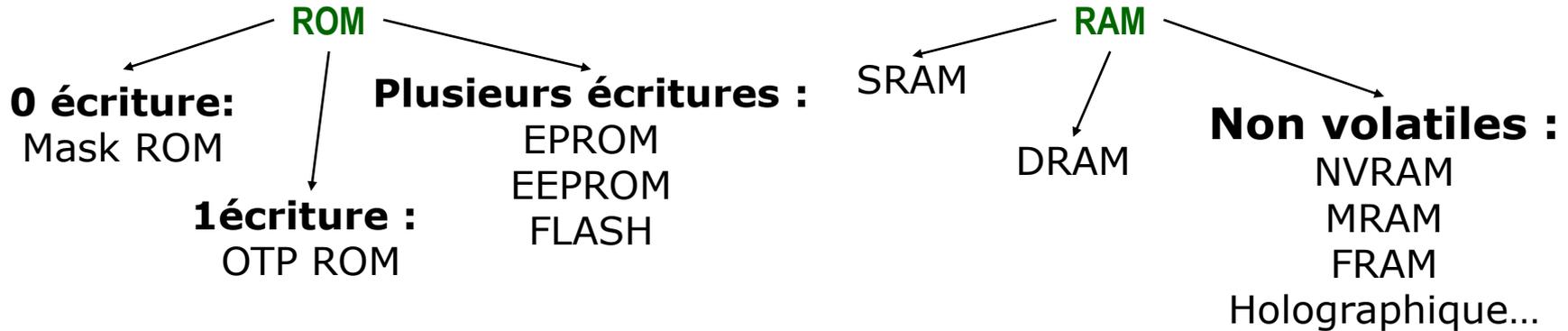
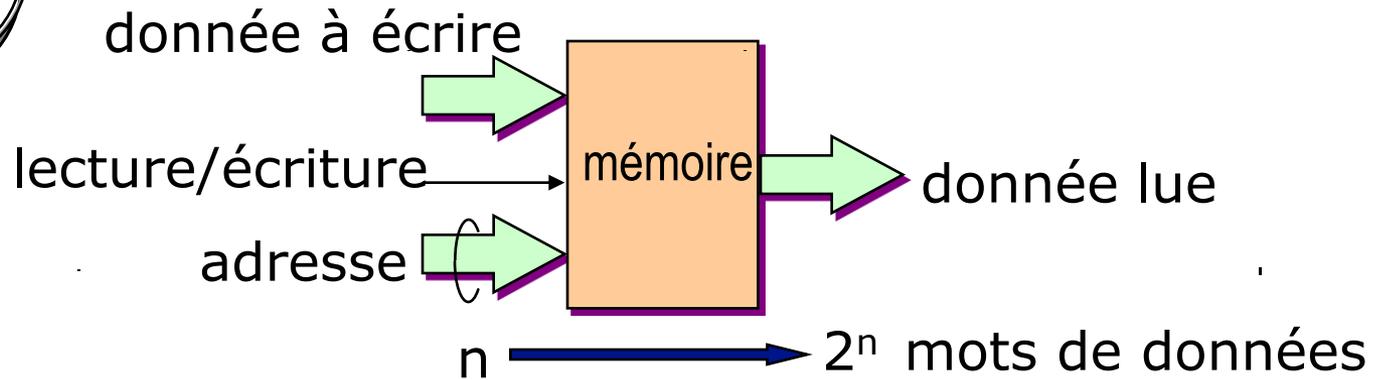
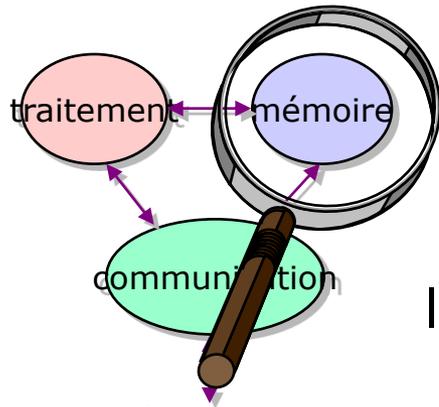
- Contrôleur centralisé complexe
- séquentialisation des opérations
- Utilisé dans les processeurs

Mémoire



Type	Accès	Vitesse	Densité	Volat.	Techno
DFF	direct	****	-	Y	
SRAM	adresse	***	++	Y	
DRAM	adresse	**	+++	Y	
FLASH	adresse	*	++	N	

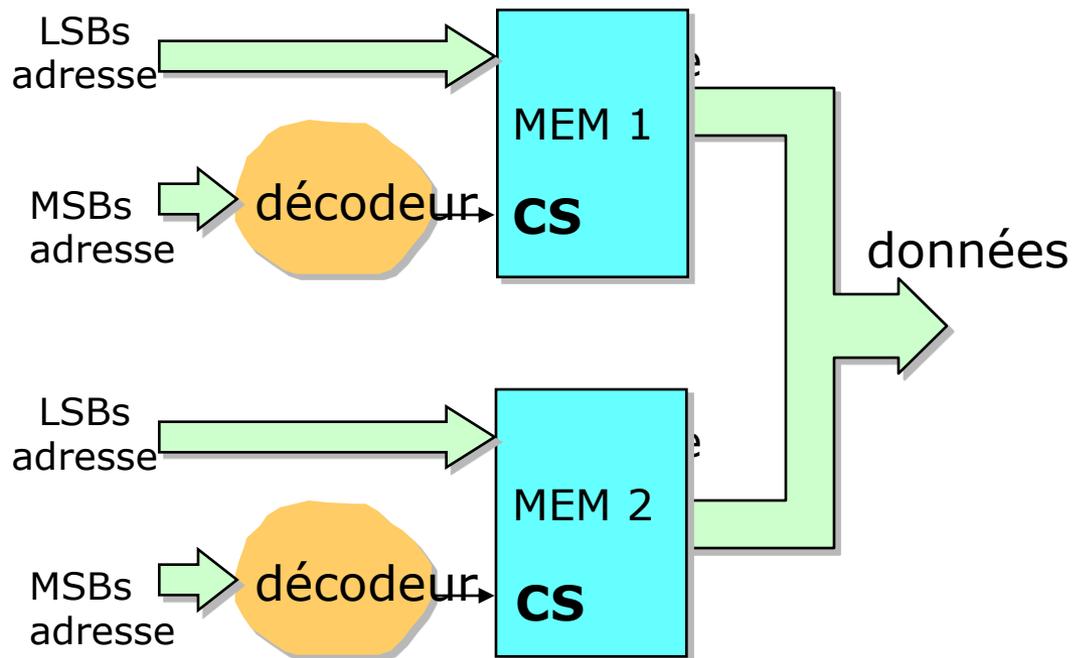
Blocs mémoire



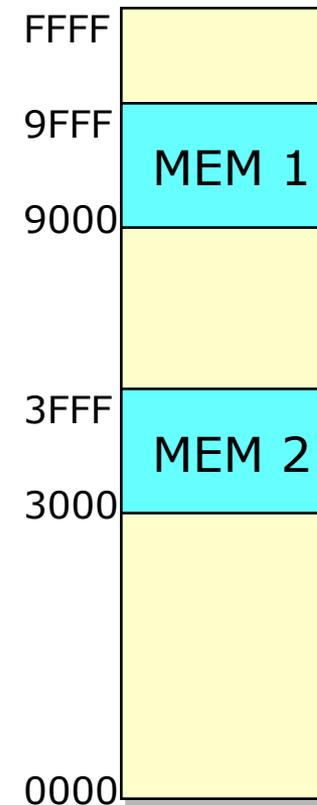
Mappe mémoire

indique les champs d'adresse mémoire dans un système

Chaque boîtier a son "**Chip Select**"

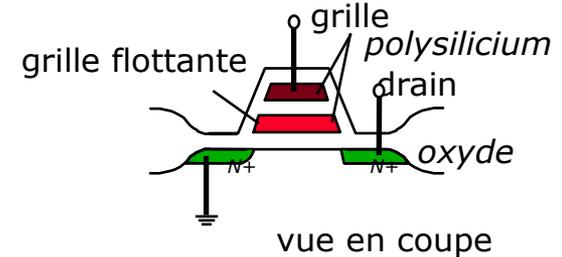


mémoire adressable

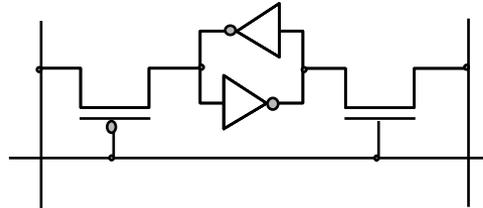


Principales technologies silicium

EEPROM, FLASH : basée sur les transistors à grille flottante

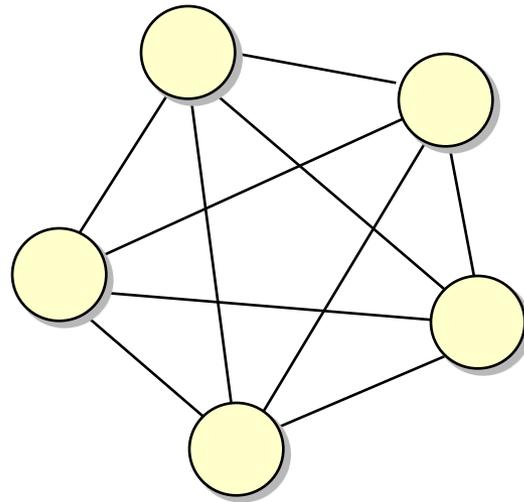
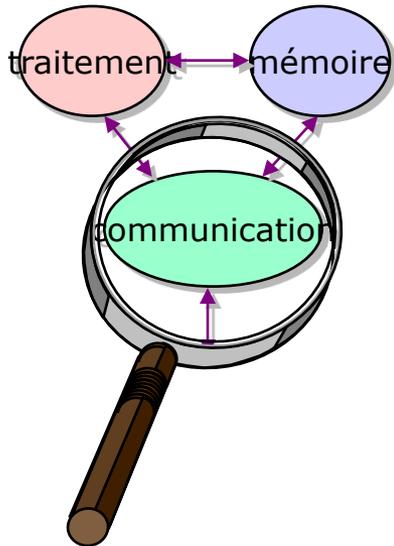


SRAM : CMOS



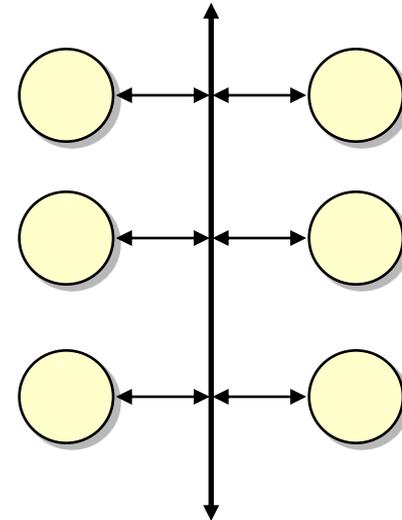
DRAM : capacité grille d'un transistor MOS
=> nécessité de rafraîchissement mais capacité x4

Canaux de transmission locaux



point à point

$$C_2^n = \frac{n(n-1)}{2} \text{ connexions}$$

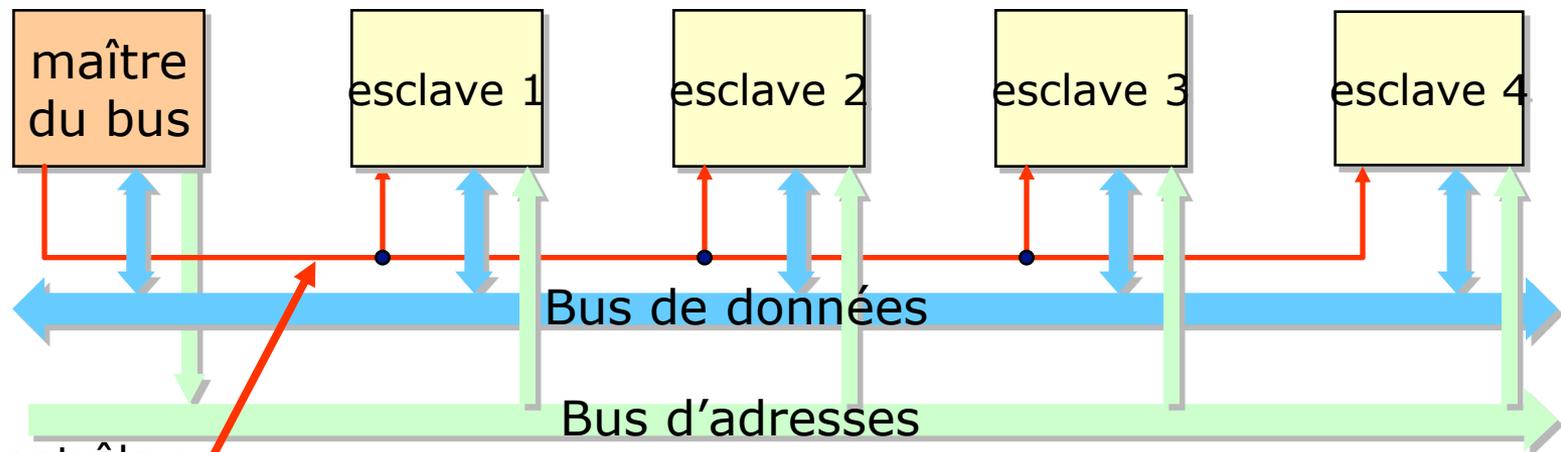


Bus

- 1 seule connexion
- nécessite un arbitrage
- transfert simultané entre 2 éléments seulement

Transfert sur un bus

- ❑ **Nécessité d'associer un champ d'adresse à un élément.**
 - chaque élément a son propre décodeur d'adresse générant un "CS"
- ❑ **Espace adressable**
 - soit dans la mappe mémoire
 - soit dans la mappe E/S (n'existe pas toujours)



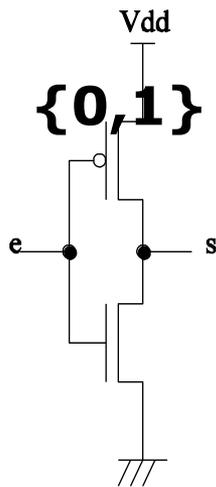
bus de contrôle :

- horloge, activité du cycle de transfert, direction des transferts (lecture/écriture), gestion du protocole

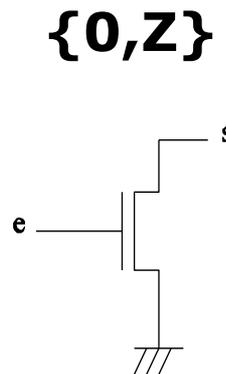
Sorties 3 états et "open drain"

Modèles fonctionnels des sorties de portes logiques :

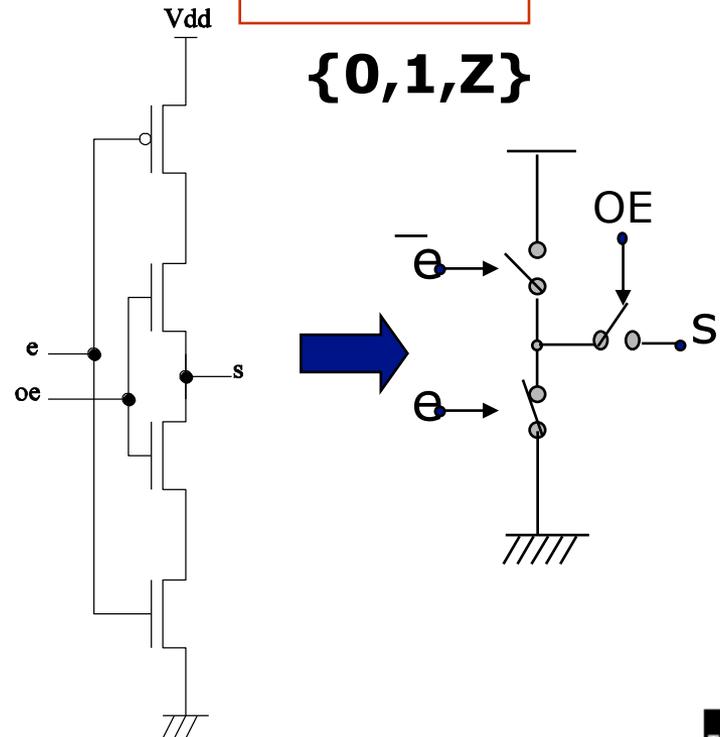
sortie classique



open drain



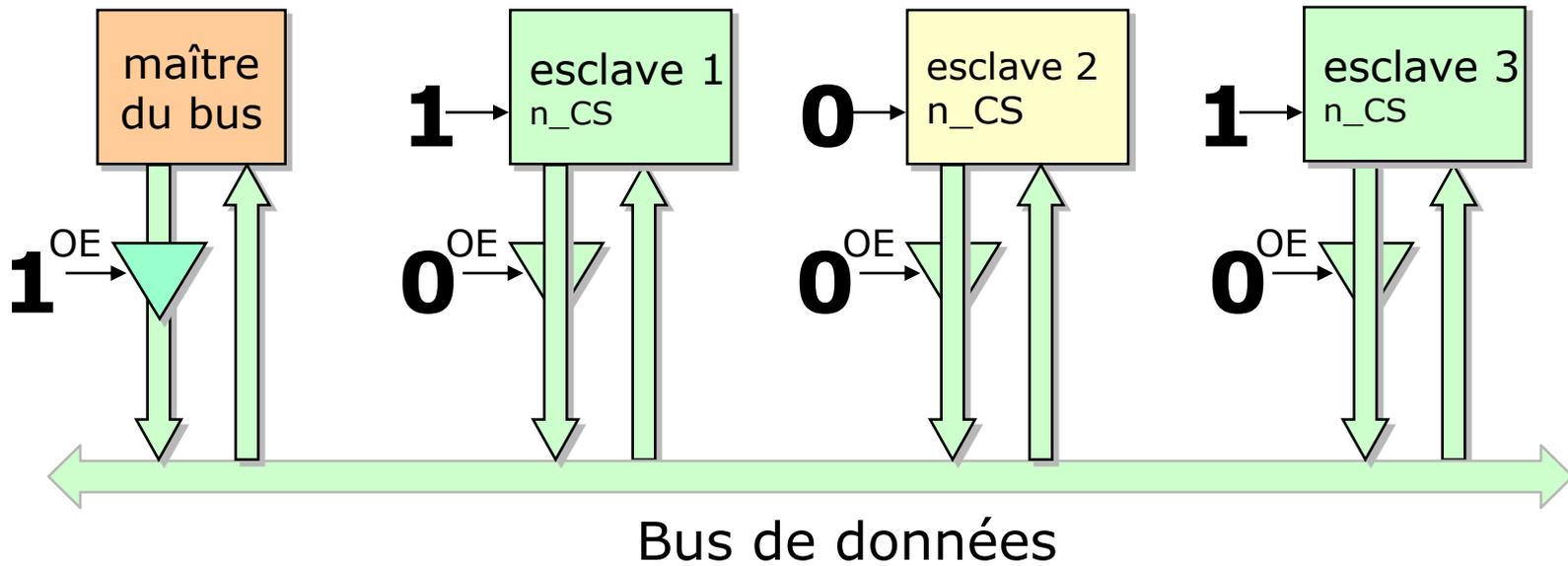
trois états



Intérêt du 3 états pour les bus

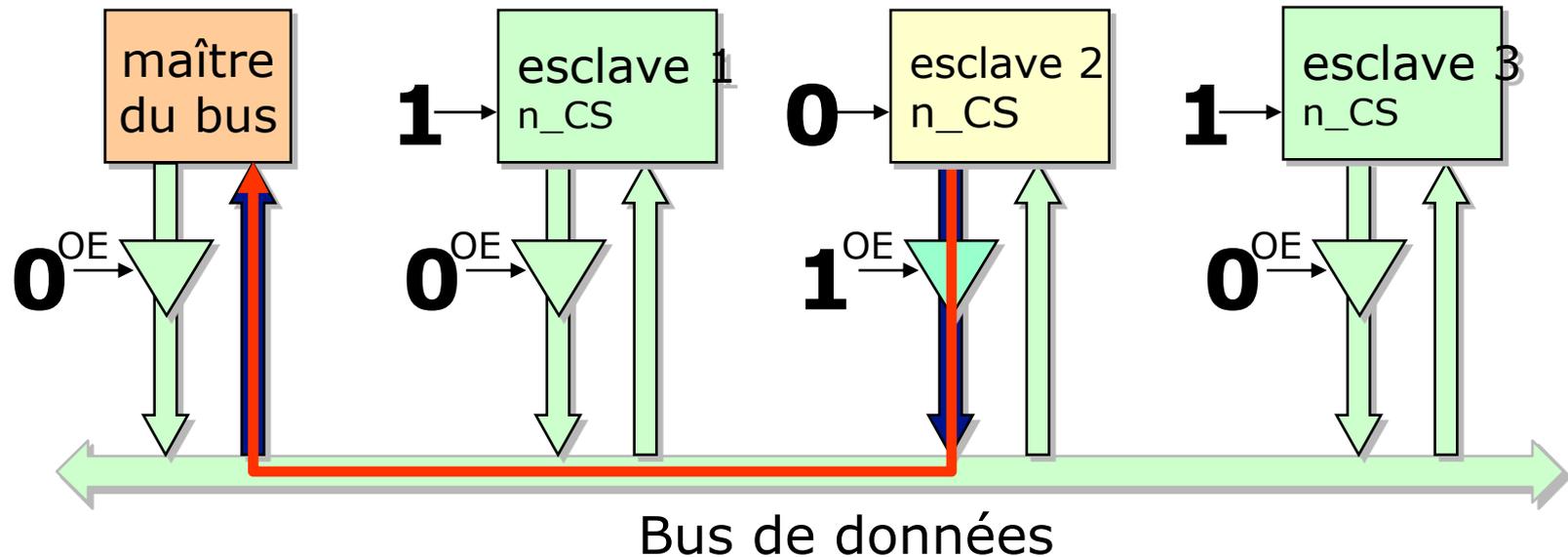
Nécessité d'une logique 3 états pour assurer la bidirectionnalité du bus :
→ 1 seul buffer doit être actif sinon conflit.

écriture



Intérêt du 3 états pour les bus

lecture

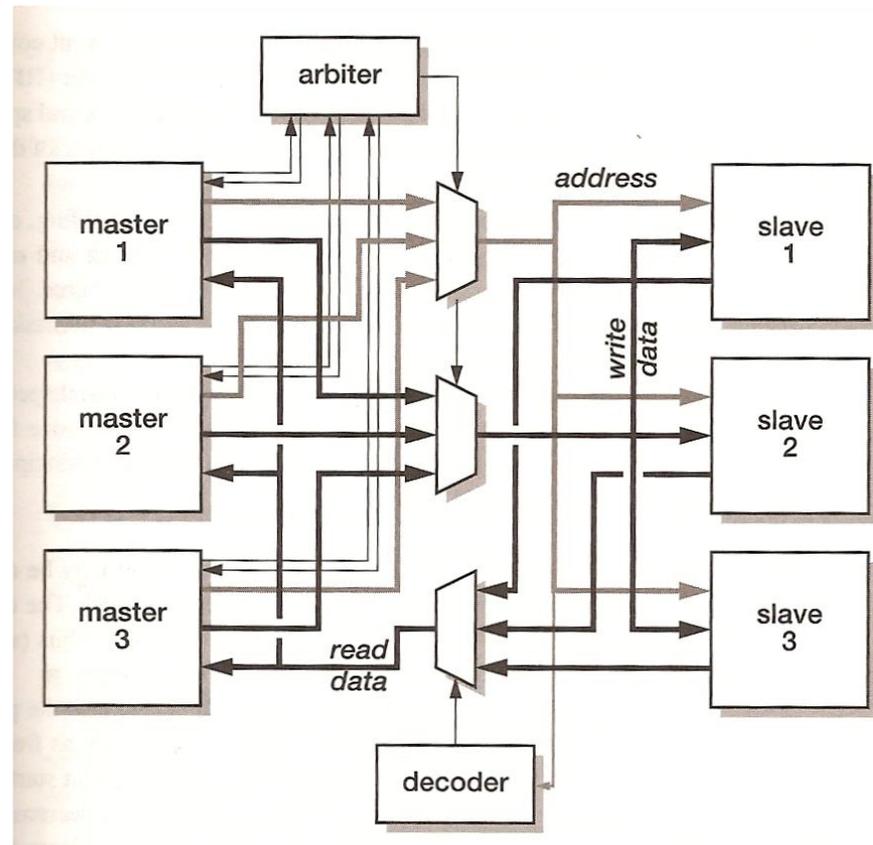


en résumé :

OE maître = écriture
OE esclave_i = lecture ET CS_i

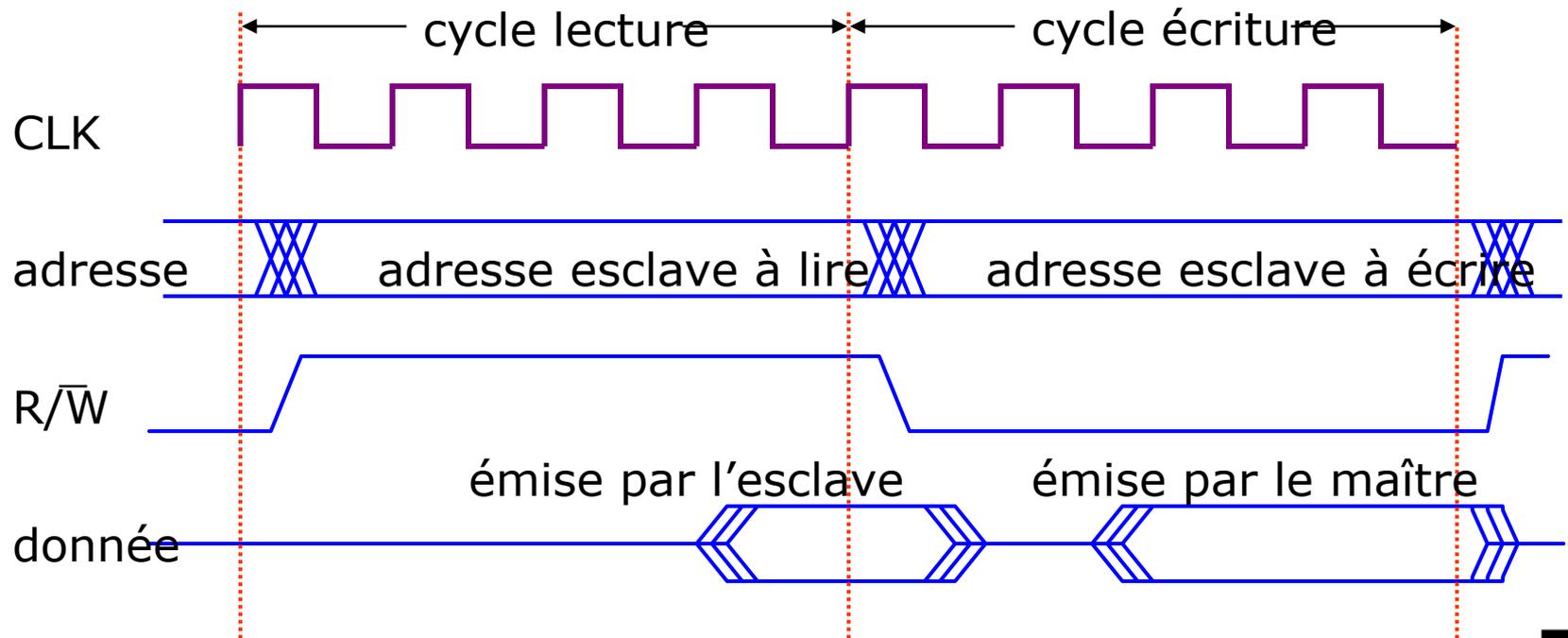
Bus multiplexé

- ❑ Permet d'éviter les conflits potentiels
- ❑ Pas besoin de 3-états
- ❑ Exemple : AMBA



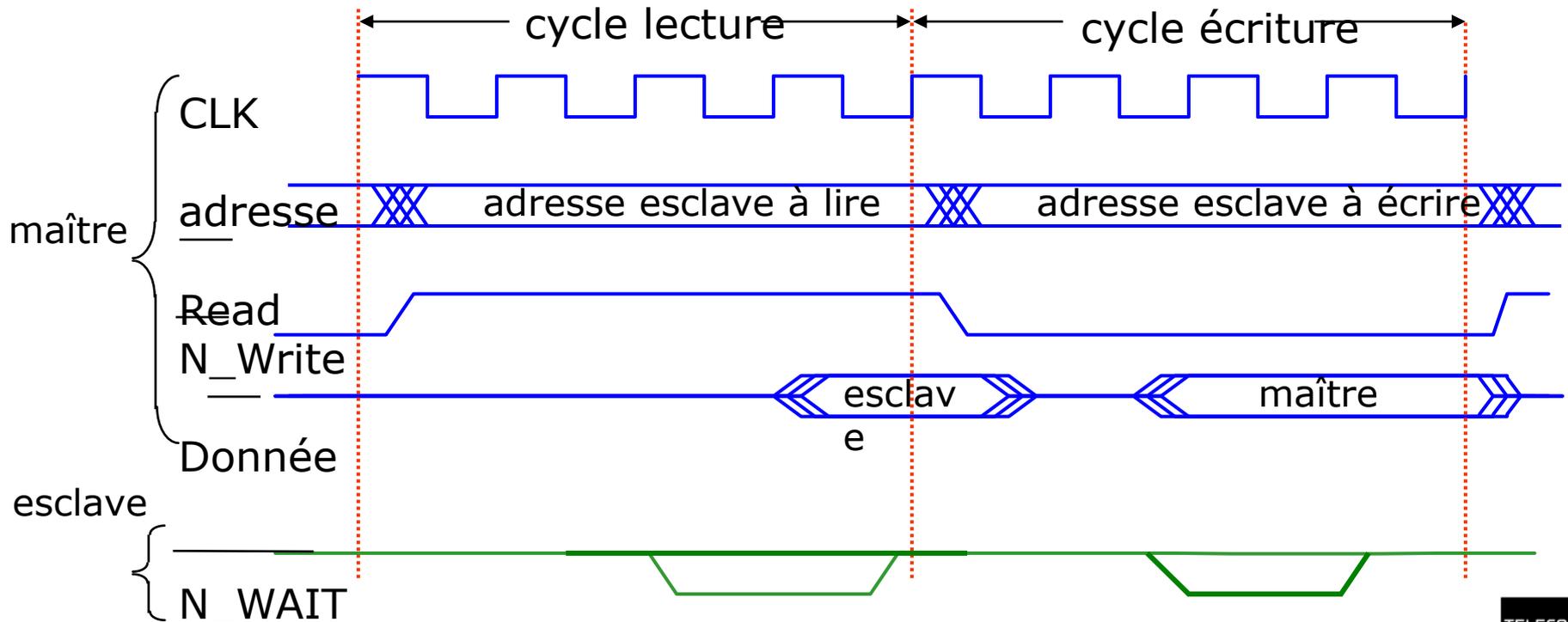
Protocole de communication synchrone

L'horloge rythme les échanges entre maître et esclave.
Le nombre de cycle est constant

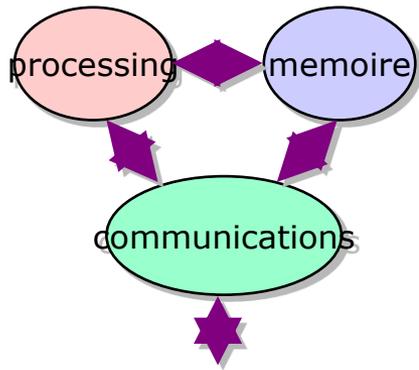


Protocole de communication asynchrone

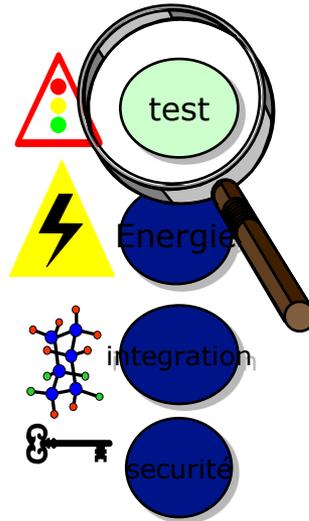
L'échange est assuré par un processus de "Handshake", l'esclave renvoie au maître soit un signal pour acquiescer ou ralentir l'échange.



Test



Homme et Machines



■ Exhaustivité du test

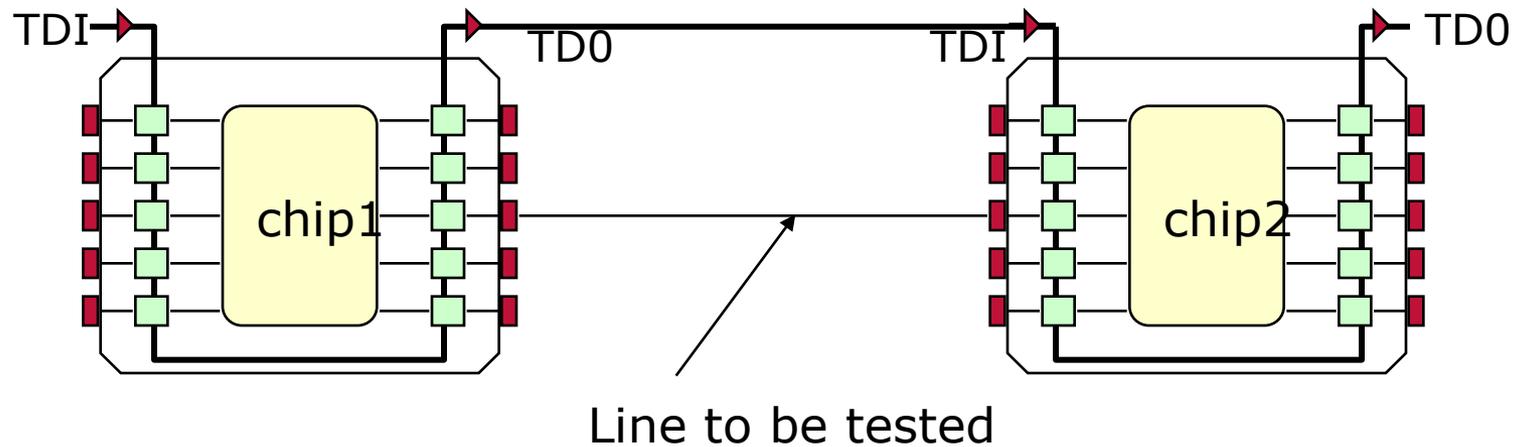
- Electrique (testeur sous pointes)
- Optique
- Boundary scan test par JTAG pour les cartes
- Scan chain pour les ASICs

Port JTAG

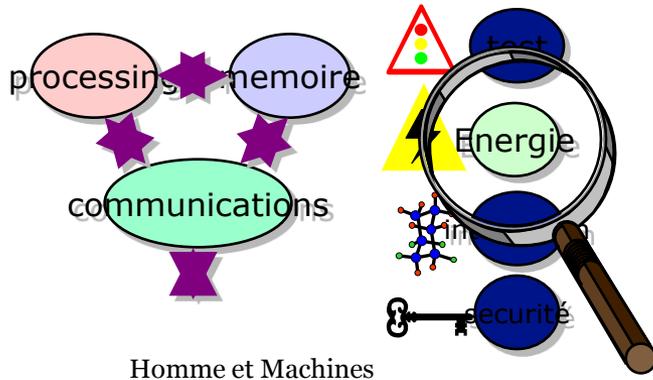
Le JTAG est un contrôleur interne ou "TAP controller" qui permet

- Test de la connectivité (par Boundary Scan testing)
- Accès aux ressources internes des processeur pour le débogage
- Chargement de netlists pour les FPGAs
- Fonctions personnalisées

exemple de Boundary Scan Test :



Contrainte de consommation d'énergie



■ Consommation dynamique

En CMOS L'essentiel de la consommation provient des courants de charge et décharge des capacités réparties

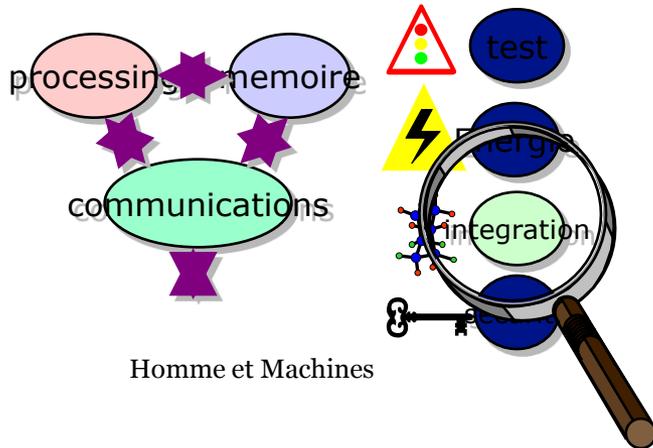
$$P = \frac{1}{2} CV_{dd}^2 f$$



■ Consommation statique

- Due aux courants de fuite. Quasi nulle dans les technologies >130nm mais devient non négligeable

Contrainte d'intégration sur carte



■ Grande Densité d'intégration sur circuit imprimé

- 2*n couches de routage
- Composants CMS sur les 2 faces
- Boîtiers très petits avec de + en + de broches

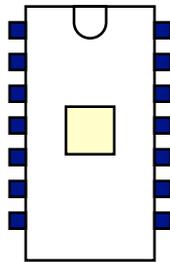
■ Problèmes d'Intégrité du signal

- Interférence entre lignes (émission/réception)
- Adaptation de lignes (réflexion)

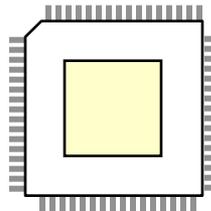
Encapsulation des circuits

■ Plus d'E/S* en moins d'espace

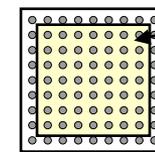
- Réduire l'espace entre broches "pitch". : 2,54mm => 0,5 mm
- Utiliser une matrice de broches plutôt que sur la périphérie du boîtier
- Utiliser les circuits "FlipChip" où les E/S sont internes à la puce



DIP (Dual In Line Package)



QFP (Quad Flat Package)
pitch=0,5mm



BGA (Ball Grid Array)
pitch=1mm

circuit
boîtier

■ Dissipation de puissance

- Boîtier céramique
- Système de refroidissement

$$* \text{ Loi de Rent } N = K.G^{\beta}$$

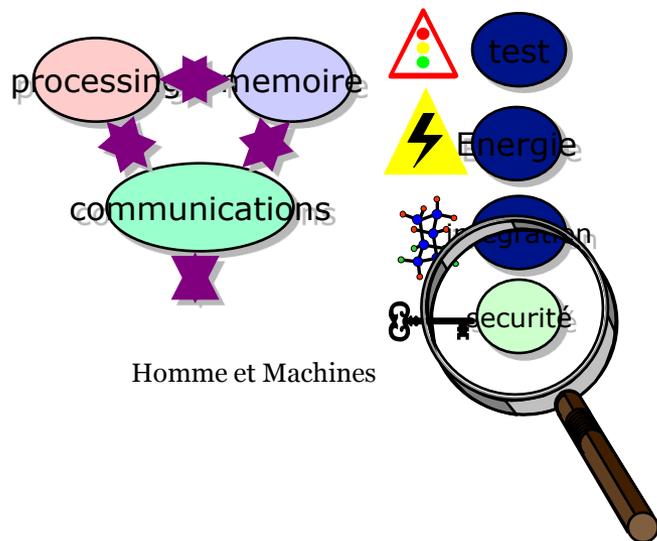
N=nombre d'E/S

K=nombre d'entrées/portes

G=nombre de portes

β=constante de Rent dépendant de l'application ~0,611

Contrainte de sécurité



■ Eviter le copiage

- Utiliser du matériel dédié

■ Protéger les communications

- Cryptographie : Algorithmes très efficaces. Matériel vulnérable car fuit de l'information par le biais de canaux physiques.
- Utiliser des structures résistantes aux "attaques"