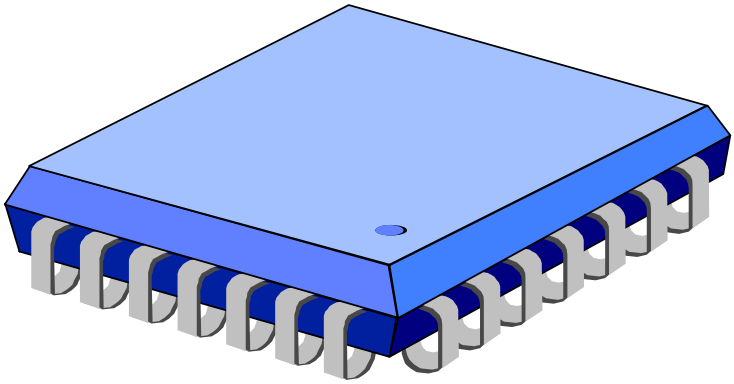
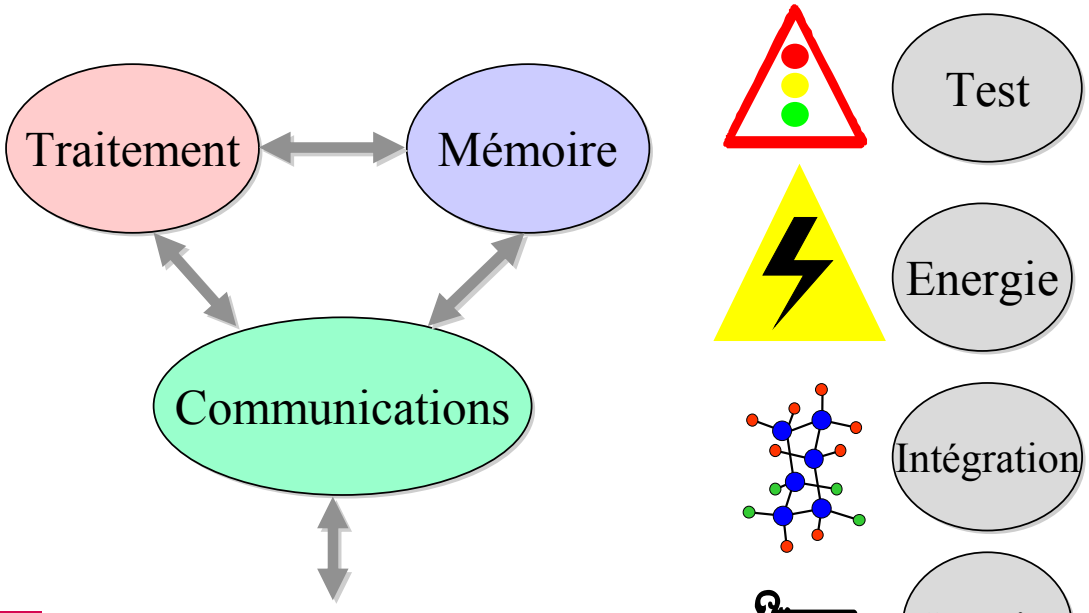
Architectures programmables et langages de description de matériel

Bases d'architectures des systèmes électroniques



TELECOM PARIS
école nationale supérieure des télécommunications

Architecture générique



TELECOM PARIS
école nationale supérieure des télécommunications

A côté des 3 grands blocs fonctionnels (**traitement, mémoire et communication**) que composent une architecture de système électronique, se trouvent des blocs très importants pour la bonne marche du système :

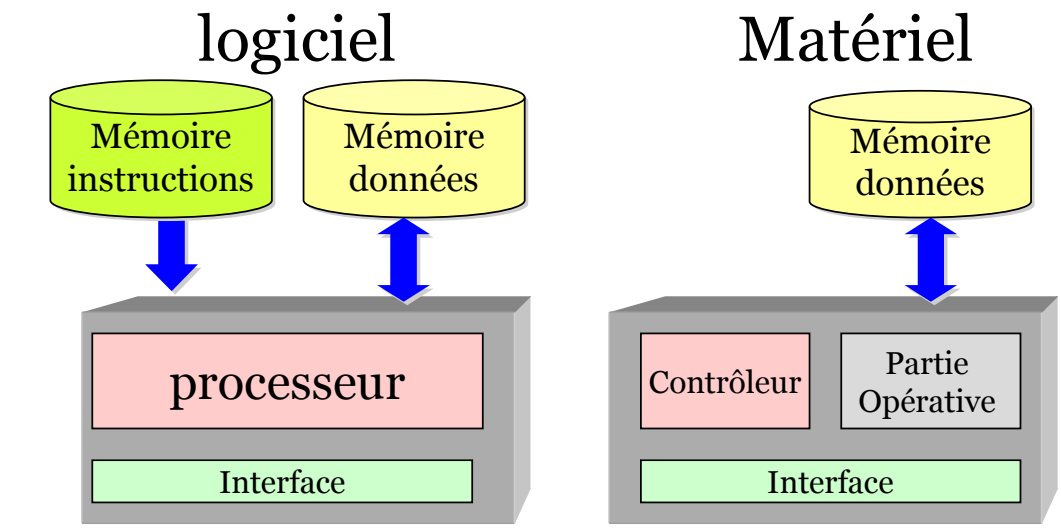
• La partie **test** : qui garantit la fiabilité du système. Par exemple pour les ASICs il faut utiliser des structures de “Scan Path” pour générer des vecteurs vérifiant l’intégrité des équipotentielles après la fabrication du circuit.

• La partie **énergie**. Qui permet d’alimenter le système d’en façon optimale, c’est-à-dire en ayant la consommation de puissance minimale. Cette partie peut consister en une optimisation architecturale ou bien un bloc spécifique contrôlant dynamiquement l’alimentation de blocs fonctionnels comme c’est le cas dans les appareils portables avec les mises en veille des blocs à forte consommation.

• Le bloc **intégration** qui doit faciliter et/ou respecter la compatibilité de l’environnement dans lequel le système va s’insérer. Cette partie concerne principalement le bloc communication ou il est question de respecter des caractéristiques électriques et des protocoles de communication.

• Le bloc **sécurité** qui permet doit protéger à la fois au sens du piratage et au sens cryptographique quand le système traite des informations confidentielles (cas des cartes à puces)

Les types de traitement



Le contrôle peut être modifié par les instructions :
⇒**Flexibilité**
⇒**Temps de développement**

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

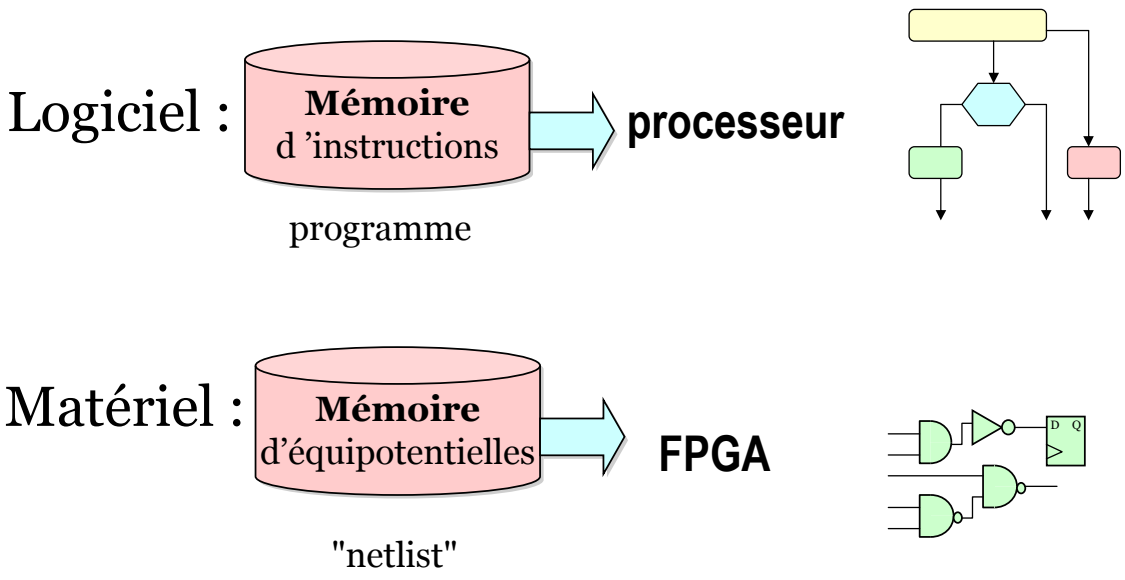
TELECOM PARIS
école nationale supérieure des télécommunications

Les 2 types de traitement reposent avant tout sur des technologies **électroniques**.
Un traitement **logiciel** utilise un processeur alors qu'un traitement **matériel** utilise un contrôleur spécialisé.

Le traitement **logiciel** est par essence très flexible car il est facile de changer les instructions et leur séquence. Le génie logiciel permet des temps de développement très courts.

Le traitement **matériel** permet d'obtenir des performances très supérieures du fait que le contrôle est spécifique et les opérateurs peuvent être utilisés en parallèle.

Programmabilité dans ELEC240



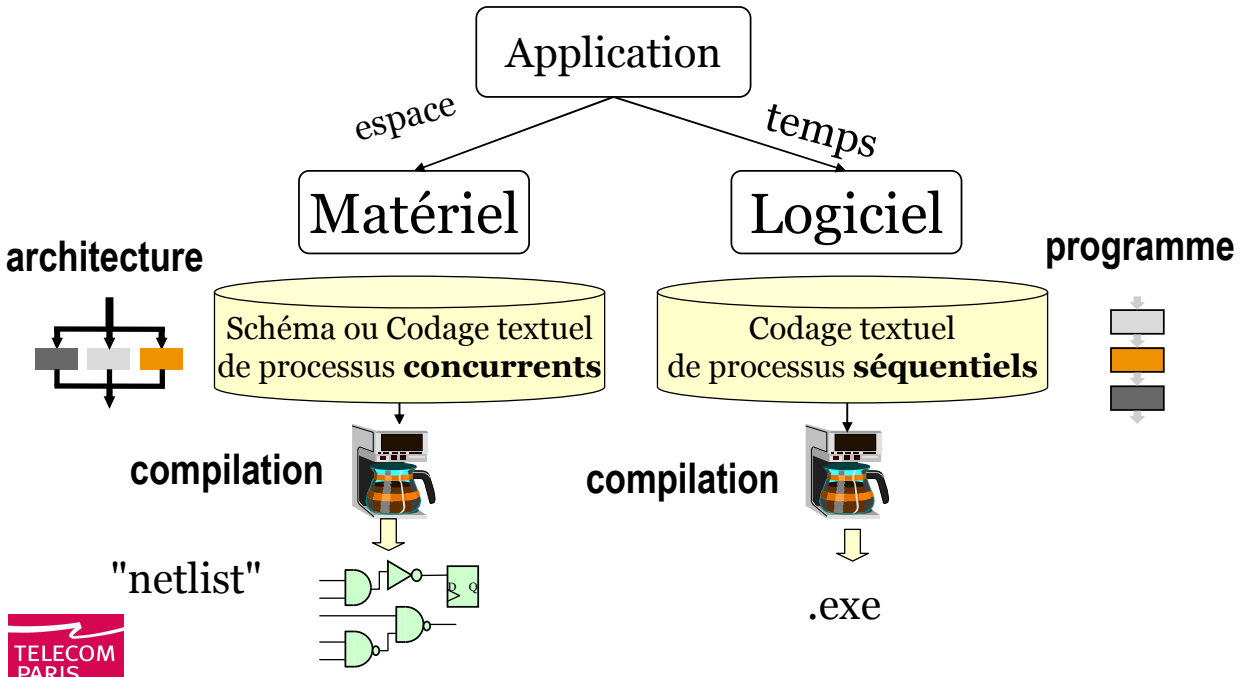
TELECOM PARIS
école nationale supérieure des télécommunications

Nous allons étudier des systèmes programmables.

Au niveau **logiciel**, la programmabilité est naturelle car les instructions constituent déjà des programmes modifiables;

Au niveau **matériel**, nous allons utiliser des FPGAs qui sont des circuits intégrés composés de cellules logiques dont les connexions entre elle ainsi que leur fonctions logiques sont programmables. On pourrait penser que ces circuits font partie du domaine logiciel mais leur programmabilité est statique et le résultat de compilation est une « netlist » et non un programme.

Partitionnement et codage



TELECOM PARIS
école nationale supérieure des télécommunications

Une des difficultés que rencontre tout ingénieur en électronique numérique est le "**partitionnement**" entre le traitement logiciel et matériel. D'une façon générale le logiciel est utilisé pour assurer la flexibilité et l'ergonomie de l'application , donc il satisfait aux besoins des couches « hautes » du traitement. Le matériel est destiné aux opérations compliquées et/ou répétitives qui se situent plutôt au niveau hiérarchique le plus bas des architectures de fonctionnement. Par exemple dans les couches ISO des applications réseaux, la couche physique est réalisée en matériel alors que la pile des protocoles est effectuée en logiciel.

Règles de fonctionnement

Démarrage : **RESET**

Nécessité d'utiliser un signal actif uniquement au démarrage : le "**RESET**"

Evolution : **SYNCHRONE**

L'utilisation d'une "**horloge**" permet de s'affranchir des dispersions temporelles des éléments logiques

TELECOM PARIS
école nationale supérieure des télécommunications

Il est impératif de respecter ces 2 règles de démarrage et d'évolution des systèmes électroniques. Tout système électronique dispose de ces 2 signaux de commandes.

Le **reset** fixe l'état initial des machines à états des contrôleurs, alors que l'**horloge** fixe le cadencement. Le fonctionnement synchrone permet de fiabiliser le traitement par la simple connaissance du chemin critique entre 2 registres (voir la leçon sur la logique synchrone)

Le types de signaux

- **Signaux de contrôle**
  - ***Horloge***
  - ***Reset***
  - ***Commandes***
    - Générées pas des **machines à états**
- **Données**
  - ***Données effectives à traités***
  - ***Pointeurs de données (adresses)***
  - ***Indicateurs (résultat correct, FIFO pleine,…)***

TELECOM PARIS
école nationale supérieure des télécommunications

Ces 2 types de signaux sont associés à 2 traitements différents.

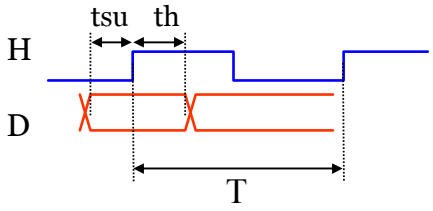
• Les signaux de **commandes** sont générés et utilisés dans les contrôleurs qui sont des machines à états spécifiables par des graphes d'états.

• Les **données** sont générées et utilisées dans les chemins de données « data path » composés d'opérateurs pilotés par des signaux de commandes

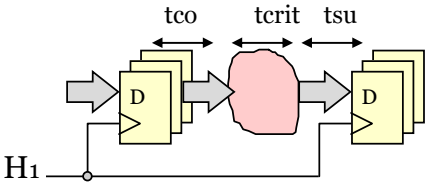
Rappel de la logique synchrone

1 seule horloge de période T
Calcul fiable si :

- **T > tco + tcrit + tsu**
- tco > th (toujours garanti)



Il suffit de connaître Tcrit !



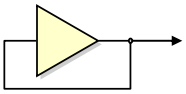
TELECOM PARIS
école nationale supérieure des télécommunications

Le fonctionnement synchrone permet de fiabiliser le traitement par la simple connaissance du chemin critique entre 2 registres (voir la leçon sur la logique synchrone)

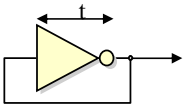
Génération d'horloge

gain A

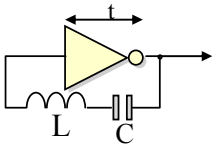
Déphasage φ



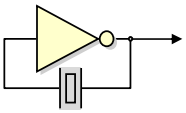
le système oscille si $\begin{cases} A = 1 \\ \varphi = 180° \end{cases}$



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



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



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

TELECOM PARIS
école nationale supérieure des télécommunications

La théorie des systèmes asservis nous dit qu'un système en "boucle fermée"est instable si son gain en "boucle ouverte" est supérieur ou égal à 1 quand sa phase est de 180°. (critère de stabilité de Nyquist). Lorsque le gain est précisément égal à 1 à une certaine fréquence où la phase vaut 180°, le système oscille à cette fréquence. C'est ce principe qui est généralement utilisé pour les oscillateurs.

Lorsqu'on reboucle sur lui-même un inverseur de temps de propagation t, il est facile de montrer sur un chronogramme que le système oscille à la fréquence 1/2t.

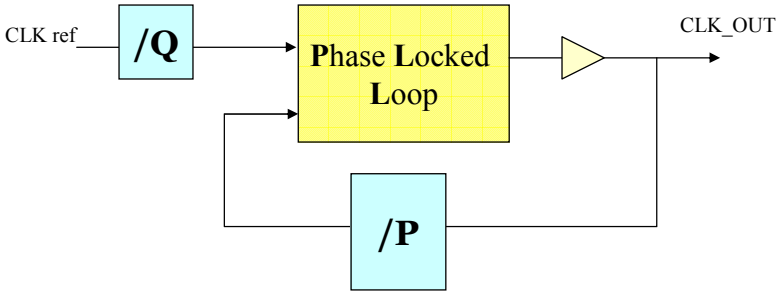
Les temps de propagation étant très dispersifs, il est difficile d'obtenir une fréquence précise. Un circuit résonant LC a une impédance nulle à sa fréquence de résonance, ce qui permet d'osciller à cette fréquence.

Les bobines utilisées pour l'inductance L ont une résistance R non négligeables, ce qui engendre une fréquence de résonance peu stable.

Le cristal de quartz a un modèle électrique correspondant à un circuit électrique LC avec un très bon facteur de qualité L/R. Ce dispositif est très utilisé comme générateur d'horloge dans les systèmes électroniques.

Synthèse d'horloge

Principe : système asservi en phase



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

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

TELECOM PARIS
école nationale supérieure des télécommunications

Les boucles à verrouillage de phase ou Phase Locked Loop **PLL** permettent de synthétiser un signal d'horloge à la fréquence voulue. L'horloge de référence peut être générée à l'aide d'un oscillateur à quartz. Elle est comparée à la sortie de la PLL. Ceci constitue un système asservi ayant pour but de minimiser l'erreur de phase entre l'horloge de référence et l'horloge générée.

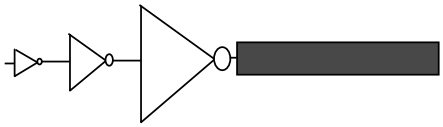
Cette erreur est ensuite moyennée et pilote une VCO « Voltage Control Oscillator » générant l'horloge de sortie.

Quand la PLL est verrouillée, l'horloge de sortie à la même phase que l'horloge de référence. Ceci permet de compenser le « skew », ou déphasage de l'horloge utile, qui est du à sa forte charge pour piloter les bascules D du circuit.

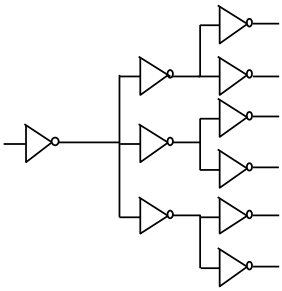
En utilisant des diviseurs de fréquence sur l'horloge de référence (facteur Q) et dans la boucle de retour de l'horloge de sortie (facteur P), la fréquence de l'horloge de sortie est égale à P/Q la fréquence de l'horloge de référence.

Distribution de l'horloge

rail d'horloge



Arbre d'horloge



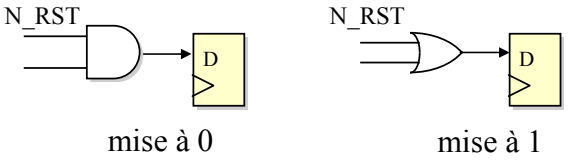
TELECOM PARIS
école nationale supérieure des télécommunications

Une horloge a une charge très élevée dans les circuits. Elle peur atteindre des dizaines de milliers d'entrées de bascule. Les méthodes pour diminuer l'influence de cette charge sont soit :

- Utiliser un amplificateur (buffer) basé sur des paires CMOS surdimensionnées et une équipotentielles d'horloge large donc moins résistive.
- Utiliser un arbre d'horloge générant des équipotentielles différentes mais équilibrées en charge de façon à avoir des « équitemporelles ».

Démarrage du système

RESET synchrone



mise à 0 mise à 1

RESET asynchrone

mise à 0 mise à 1

TELECOM PARIS
école nationale supérieure des télécommunications

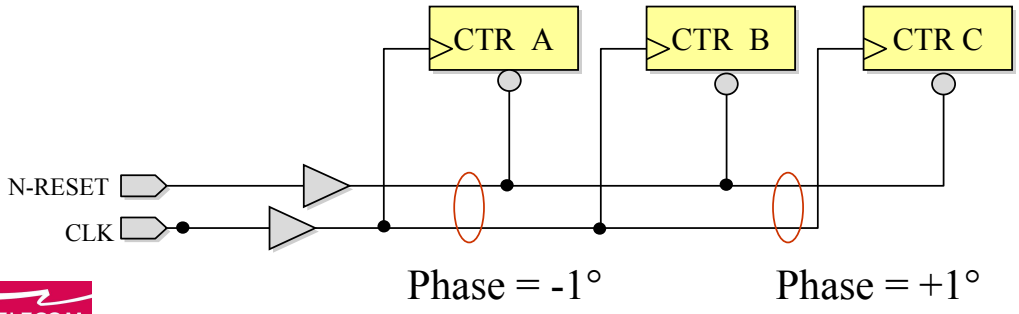
Le système électronique nécessite un signal spécifique d'initialisation : le RESET.

Il correspond à une équipotentielle particulière. Celle-ci peut agir d'une façon synchrone sur l'entrée D de la bascule mais il est plus judicieux d'utiliser une entrée asynchrone (mise à 1 : preset ou mise à 0 : clear) des bascules D. Les bibliothèques ce cellules ASIC et FPGA dispose généralement de ce type de bascule.

L'utilisation des entrées asynchrone est réservée au RESET et il est vivement recommandé de ne pas utiliser de la logique sur ces entrées qui au moindre parasite risquent de réinitialiser la bascule.

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



Phase = -1° Phase = +1°

TELECOM PARIS
école nationale supérieure des télécommunications

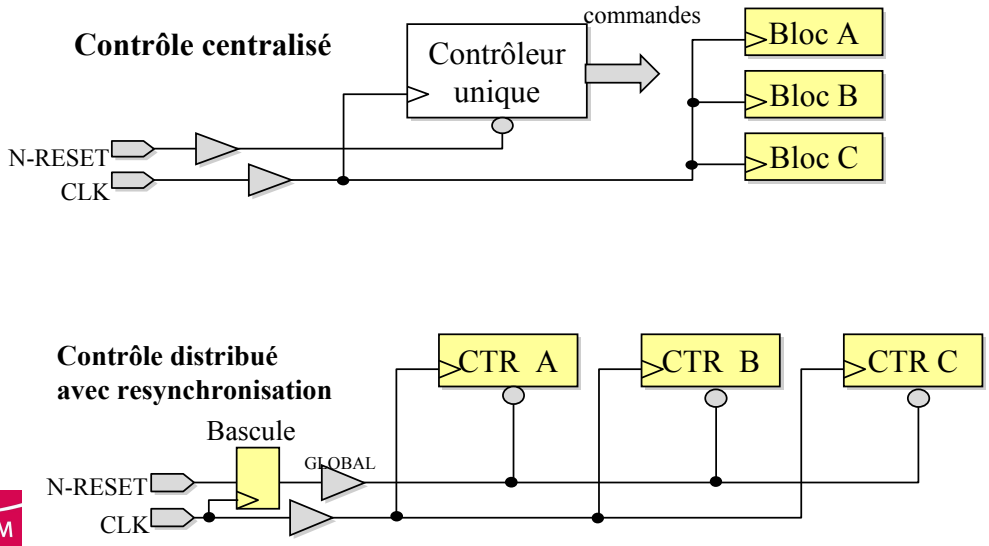
Certaine précautions doivent être prises pour démarrer correctement un système.

Un contrôleur (ou machine à états) doit impérativement être initialisé avec le RESET. Si ce n'est pas le cas, le système risque de démarrer dans un état qui n'est pas fonctionnel ou qui n'est pas celui de l'état initial de la machine à états.

Comme le RESET a une charge très élevée, il existe un risque sur la cohérence du démarrage des différents contrôleurs du système. Le RESET peut arriver juste avant le front d'horloge pour certains et juste après pour d'autres. Il existera alors un cycle d'horloge d'écart entre eux.

Stratégies pour bien démarrer

Utilisation systématique du Reset sur tous les processus de contrôle



TELECOM PARIS
école nationale supérieure des télécommunications

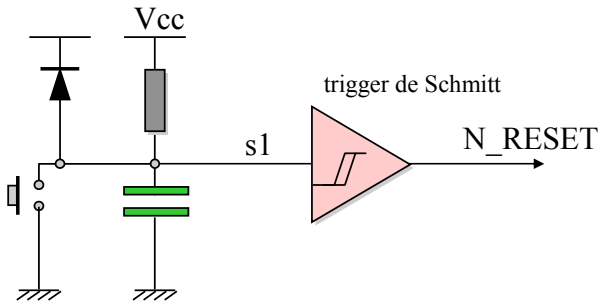
Pour éviter le problème précédent; les solutions consistent à :

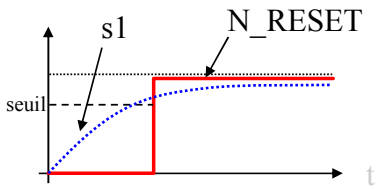
- Centraliser le contrôle, ce qui n'est possible que pour les petits système
- Auto synchroniser les contrôleurs entre eux.
- Synchroniser le RESET avant de l'utiliser. Cette solution simple est souvent utilisée. Il faut toutefois vérifier que le RESET a le temps de se propager durant une période d'horloge car il est très chargé.

Génération du RESET

- au démarrage de l'alimentation
- par bouton poussoir
- externe

exemple :





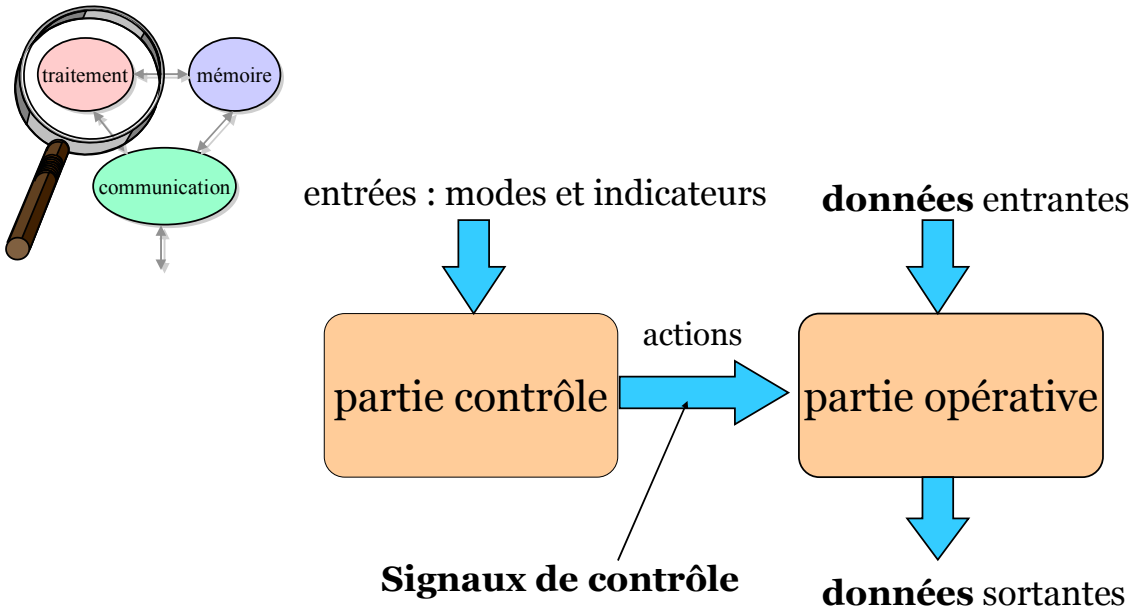
TELECOM PARIS
école nationale supérieure des télécommunications

Le RESET est généré automatiquement à la mise sous tension. Pour régler son temps d'activation nécessaire à l'établissement de l'alimentation, un circuit RC est utilisé.

Un comparateur à hystérésis (Trigger de Schmitt) permet d'avoir un bel échelon sur la sortie RESET. L'hystérésis évite l'influence du bruit thermique qui peut donner lieu à des oscillations lorsque le RESET atteint le seuil de comparaison.

Certains systèmes disposent d'un bouton poussoir pour réarmer manuellement le système. C'est le cas pour certains ordinateurs PC.

Unité de traitement



TELECOM PARIS
école nationale supérieure des télécommunications

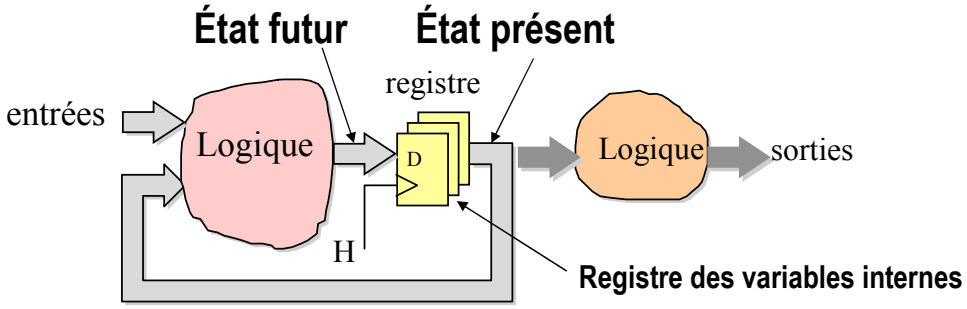
Le traitement est composé de la partie **contrôle** pour générer les signaux de commandes et de la partie **opérative** pour traiter les données.

Les systèmes électroniques de contrôle pur (robots, automates) n'ont pas ou peu de données à traiter. Ils génèrent des commandes pour les actionneurs électromécaniques.

Machine à états

Génère les signaux de contrôle :

Machine séquentielle synchrone à états finis



Machine à états de Moore : sorties = f(état)
Machine à états de Mealy : sorties = f(entrées,état)

TELECOM PARIS
école nationale supérieure des télécommunications

Rappel sur les machines à états qui constituent la partie contrôle des systèmes.

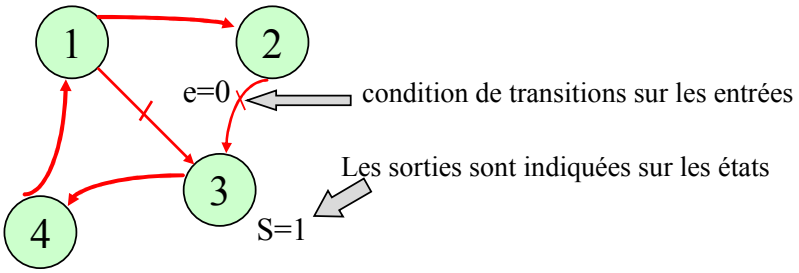
L' « intelligence » du contrôleur vient de son calcul séquentiel qui dépend des entrées et des états passés du circuit.

Une machine à états est un circuit séquentiel possédant des variables internes dont les valeur illustrent l' **état** du circuit. Pour un système synchrone, ces variables internes sont stockées dans un registre d'états et elle servent d'entrées à la fonction de calcul de l'état futur qui va devenir l'état des variables internes au prochain front d'horloge.

Les sorties de la machine à états dépendent des variables internes (Machine de Moore) ou des variables internes et des entrées (Machine de Mealy)

Graphe d'états

spécification d'une machine à états par graphe :



Le graphe doit être :

- **complet : pour chaque état Σconditions = 1**
- **Non contradictoire : $\Sigma_i\Sigma_{j!=i}CiCj=0$**

TELECOM PARIS
école nationale supérieure des télécommunications

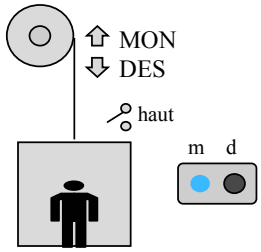
La spécification par graphe nécessite des vérifications fonctionnelles garantes de la spécification du système. D'autre part le graphe doit respecter ces 2 règles :

• La **complétude** : Tout état actif doit donner lieu à un passage à un autre état (ou passage d'un « jeton »). Il faut donc que toutes les combinaisons d'entrée définissent le futur état sinon le jeton risque d'être perdu et la machine peut se bloquer.

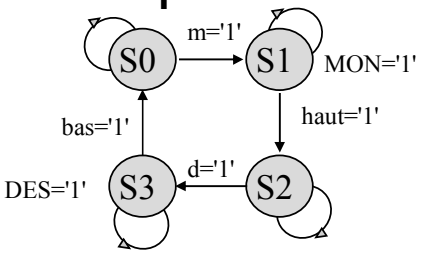
• La **non-contradiction**. La passage d'un état doit se faire vers un unique état. Le fait d'avoir plusieurs jetons simultanément rend le système non fiable.

Dans les systèmes complexes, les machines à états peuvent être architecturées de façon hiérarchique.

Exemple de machine à états



Graphe d'états



Codage des états

"one hot":
1état sur 4 bits Q3 Q2 Q1 Q0
S0=0001
S1=0010
S2=0100
S3=1000

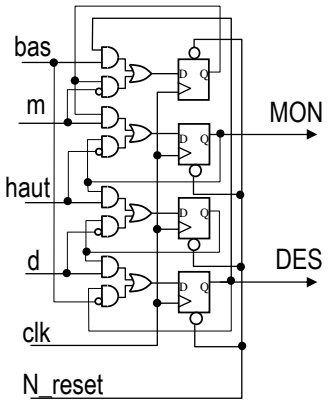
Équations des sorties :

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



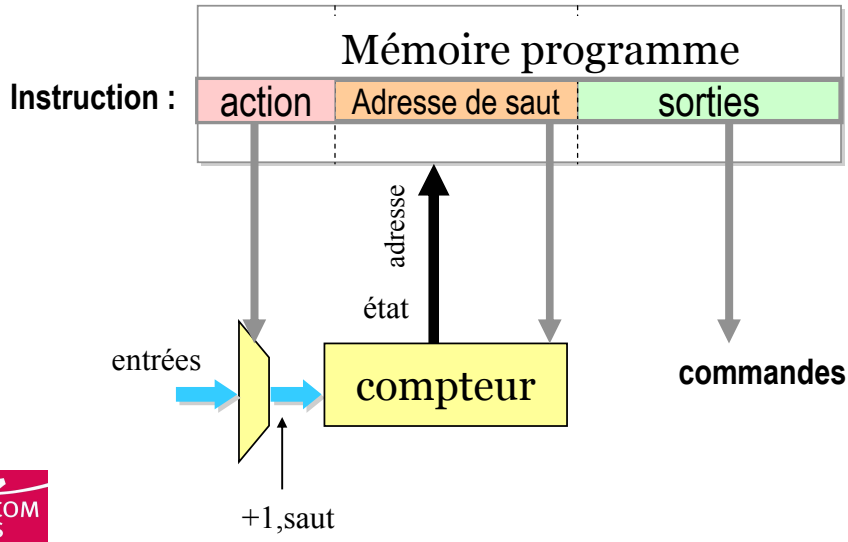
TELECOM PARIS
école nationale supérieure des télécommunications

Dans cet exemple, le codage « One hot » est choisi. Les équations illustrent la facilité d'obtention des résultats avec ce type de codage, au détriment du nombre de bascules.

L'utilisation des langage VHDL ou Verilog avec des outils de synthèse évite au concepteur les étapes de choix des bits et de codage.

Séquenceur

Le séquenceur est une machine à états de MOORE où un compteur indique l'état courant



TELECOM PARIS
école nationale supérieure des télécommunications

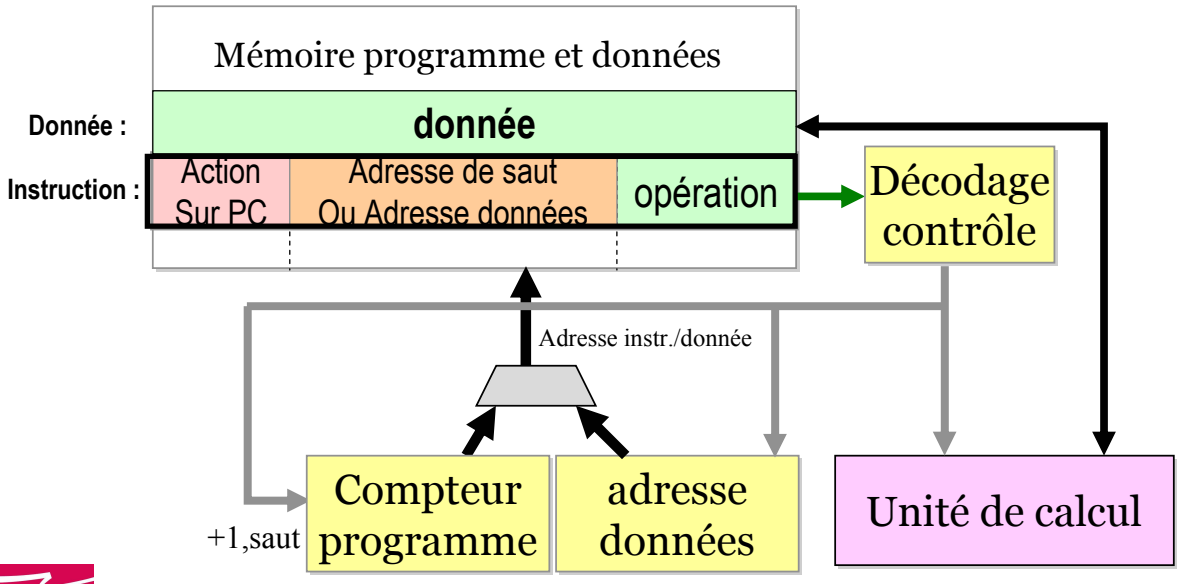
Le séquenceur est une machine à états de Moore générique dont le registre d'état est un compteur qui peut être incrémenté ou chargé par une nouvelle valeur. Ce compteur pointe dans une mémoire sur une « instruction » qui contient l'action à effectuer (+1 ou saut) en fonction des entrées et l'adresse de saut dans le cas d'un saut. L'instruction a également un champ pour les sorties servant à piloter des actionneurs extérieurs.

L'ensemble des instructions s'appelle un programme et le compteur s'appelle le compteur de programme ou PC. Cette architecture s'inspire de la fameuse machine de Turing où le ruban est remplacé par une mémoire électronique.

Cette architecture de séquenceur s'appelle aussi automate programmable. Elle est très utilisée en électromécanique pour piloter des robots industriels.

Architecture "Von Neumann"

Le processeur est un séquenceur pour le traitement de données

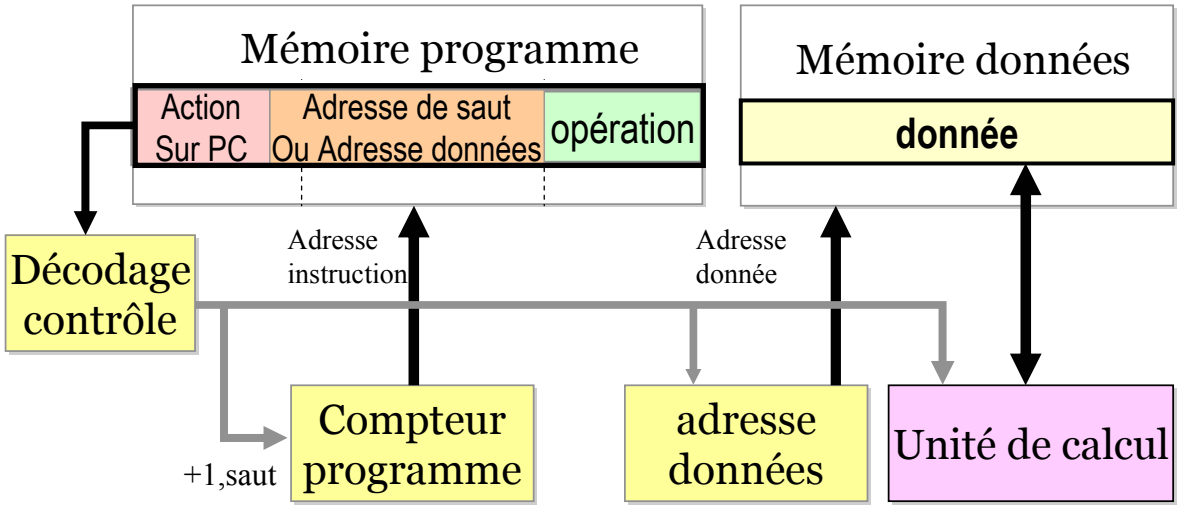


TELECOM PARIS
école nationale supérieure des télécommunications

Un séquenceur servant à traiter des données s'appelle un « processeur » . Le champ des sorties (ou champ «opérations) vient alors commander l'unité opérative ou s'effectue les calculs. Dans l'architecture dite « Von Neuman » du nom d'un des pionniers ayant réalisé le premier ordinateur (l'ENIAC en 1945), les données sont stockées dans la même mémoire que les instructions. Dans le champ instruction, se trouve l'adresse de la donnée à traiter ou à stocker, dans le cas d'un opération, ou l'adresse de saut, dans le cas d'un branchement. Une unité de traitement spécifique à l'adresse est nécessaire pour que la mémoire soit adressée par le compteur de programme ou par l'adresse de la donnée.

Architecture "Harvard"

Les mémoires programme et données sont séparées => rapidité accès x2



TELECOM PARIS
école nationale supérieure des télécommunications

L'architecture Von Neuman oblige à aller chercher successivement en mémoire l'instruction puis la donnée. En disposant de 2 mémoires respectives pour les instructions et les données, il devient possible de paralléliser les accès et donc d'augmenter les débits de calcul. C'est le but de l'architecture Harvard qui est utilisée dans tous les processeurs performants.

Types d'instructions

- **Traitement**
  - opérations arithmétiques, logiques,décalage de bits
  - format virgule fixe, flottant
- **Transferts avec la mémoire**
  - load, store
- **Contrôle**
  - Branchements
  - branchements aux sous programmes
    - Sauvegarde automatique de l'adresse de retour en « PILE »
- **Système**
  - Interruptions logicielles : Appel de fonctions de l '«operating system »
- **Coprocesseur**
  - Instructions spécifiques à un opérateur extérieur « coprocesseur »

TELECOM PARIS
école nationale supérieure des télécommunications

Les instructions servent à effectuer des opérations sur les données et à effectuer des sauts de programme conformément aux algorithmes de calcul. De façon à effectuer les calculs en interne et ne pas à aller chercher systématiquement les données en mémoire, les instructions LOAD et STORE servent respectivement à précharger des registres données et à sauvegarder les résultats en mémoire.

Certains processeurs disposent d'instructions facilitant la mise au point ou l'utilisation de système d'exploitation multi-tâches "OS" . C'est la cas des exceptions logicielles permettant au processeur de changer de mode. Par exemple les appels systèmes pour effectuer des processus du noyau de l'OS peuvent utiliser ce type d'instruction.

D'autre part les processeurs peuvent être aidés de coprocesseurs qui sont des circuits de calculs externes permettant d'accélérer le traitement. Pour éviter de faire de écritures/lectures avec le coprocesseur , le processeur communique avec lui par un canal de commandes dédié permettant au coprocesseur de se synchroniser et saisir à la volée les données à traiter ou à stocker

Modes d'adressage

indiquent la façon d'accéder à la donnée

exemples courants :

Mode d'adressage	Contenu de l'instruction
Immédiat	donnée
Absolu	adresse donnée
Registre	numéro de registre
Registre Indirect	numéro de registre contenant l'adresse donnée
Indirect	adresse d'une adresse donnée

TELECOM PARIS
école nationale supérieure des télécommunications

La donnée peut être directement indiquée dans l'instruction, ce qui correspond à l'adressage immédiat. Ce mode d'adressage présente les inconvénients suivants :

- Mot d'instruction long pour rajouter le champ donnée
- Un même programme ne peut pas être utiliser pour plusieurs listes de données.
- Une même liste de donnée ne peut pas être traitée par différents programmes

Pour les 2 derniers points, il est préférable d'indiquer l'adresse de la donnée dans l'instruction (mode d'adressage absolu). Pour réduire la taille de l'instruction, certains microprocesseurs considèrent la différence d'adresse qui les sépare avec le compteur programme de l'instruction (adressage relatif).

Si ce sont des registres internes qui contiennent les données, le mot instruction est encore réduit car il suffit de donner l'indice du registre.

Si la donnée est en mémoire, un registre peut servir de pointeur ou de registre d'adresse (mode d'adressage indirect). Dans ce cas l'instruction ne contient que l'indice de ce registre d'adresse et le processeur va effectuer un cycle supplémentaire pour accéder à la donnée en mémoire adressée par ce registre d'adresse.

Modes de fonctionnement

Chaque mode correspond à des ressources propres

=> permet la mise en place de protection pour les OS

Il existe au moins 2 modes : Superviseur et Utilisateur

- Les ressources propres sont généralement des registres : PC, pointeur de pile
- Le passage d'un mode à un autres s'effectue par les exceptions (interruption logicielle ou matérielle)

TELECOM PARIS
école nationale supérieure des télécommunications

De façon à pouvoir exécuter des programmes de supervision ou systèmes d'exploitation multi-tâches et multi-utilisateurs, certains processeurs ont différents modes de fonctionnement.

Ces modes permettent de délimiter les ressources de façon à ce qu'un utilisateur ne puisse bloquer le système. Le passage d'un mode à l'autre s'effectue par un mécanisme d'exceptions qui est généré soit par une instruction, soit par une entrée du microprocesseur (interruption, reset, alerte,…)

Exceptions

Permettent de signaler un événement particulier au processeur pour qu'il effectue un traitement spécifique.

Elles peuvent être générées par :

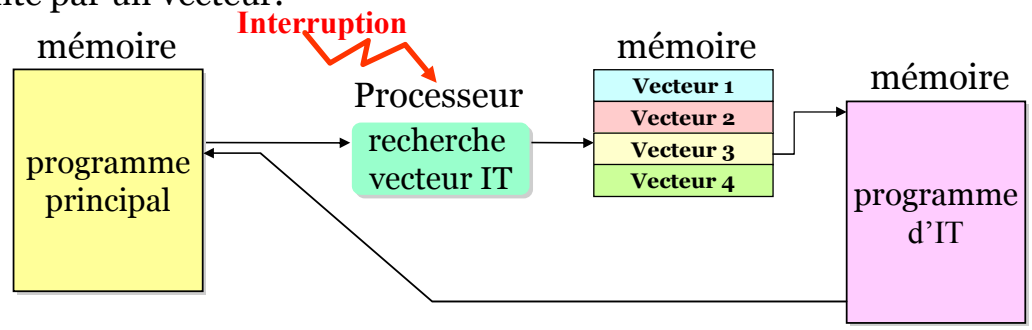
- interruption logicielle (instruction d'appel système)
- interruption matérielle (un signal d'interruption extérieur devient actif)
- dysfonctionnement du système (échec d'un accès mémoire, instruction non reconnue,...)

TELECOM PARIS
école nationale supérieure des télécommunications

Les exceptions permettent d'interagir aux évènement extérieurs de façon à dérouter le programme vers un traitement plus propice. Par exemple , les programmes d'entré/sortie peuvent nécessiter le passage en mode "superviseur" par le biais d'un appel système qui effectuer une fonction sécurisée par le noyau du système. De même un système de surveillance de cuve peut réagir s'il reçoit une "**interruption**" de l'extérieur.

Traitement des exceptions

Une exception donne lieu à l 'exécution d'un programme de traitement pointé par un vecteur.



- Les vecteurs d'exception sont regroupés dans une table située à un endroit spécifique de la mémoire.
- Dans le cas d'une interruption ils peuvent être communiqués par l'élément générateur de l'interruption
- Les exceptions simultanées sont traitées par ordre de priorité

TELECOM PARIS école nationale supérieure des télécommunications

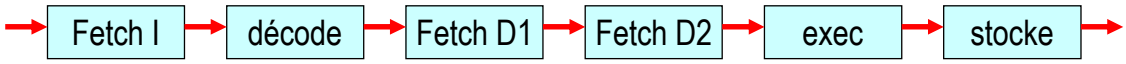
Lors de la réception d'un exception (une interruption dans cet exemple), l'exécution du programme principal est interrompu par le processeur qui va sauvegarder la valeur du PC (adresse de retour) de façon à revenir au programme principal.

Le processeur doit alors déterminer le générateur de l'exception et scruter en conséquence la table des vecteurs d'exception de façon à connaître l'adresse du programme de traitement En début de ce programme, une sauvegarde des registres peut être faite comme lors d'un appel à un sous–programme. La sauvegarde est effectuée dans une zone mémoire particulière : la "pile" de façon à ne pas écraser le contexte des registres du programme principal. Ce contexte est ensuite "dépilé" avant le retour au programme principal qui s'effectue en rechargeant la valeur du PC avec l'adresse de retour.

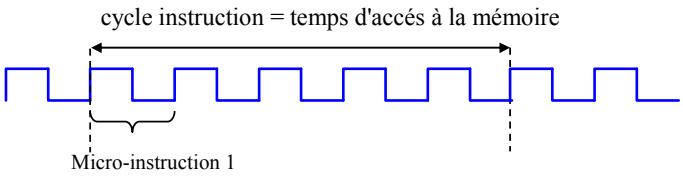
Les interruptions pouvant être nombreuses, un processeur est associé à un contrôleur d' interruptions qui permet d'affecter des priorités en cas de simultanéité et de savoir quel élément est le générateur.

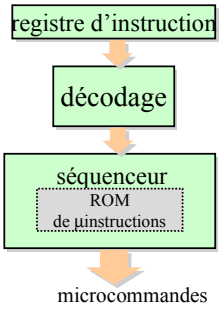
Le passé : Processeur CISC

- L'accès à la mémoire extérieure est relativement lent
- L'exécution d'une instruction nécessite plusieurs étapes



▪ Idée : Utiliser le temps d'attente mémoire pour exécuter des instructions compliquées avec des modes d'adressage variés





TELECOM PARIS
école nationale supérieure des télécommunications

Partie contrôle 70% (sous utilisée)
Partie opérative 30%

Dans les années 1970, après la naissance des premiers microprocesseurs, les accès mémoires étaient relativement lents et le processeur disposait de plusieurs cycles d'horloge (au moins 8 pour le 68000) pour aller chercher, décoder et exécuter l'instruction. Ce nombre de cycles permettait d'effectuer des calculs micro-séquencés qui donnait lieu à des instructions très compliquées mais peu souvent utilisées, ce qui justifie le nom de ce type de processeur : "Complex Instruction Set computer".

Processeur RISC

- Utiliser des registres, de la mémoire locale rapide ou « cache » pour éviter d'attendre
- Paralléliser les instructions : techniques pipeline

Dans les années 80, cette approche devint possible grâce aux progrès technologiques . Le gain en performance doit tendre vers « une instruction en 1 cycle d'horloge ». Le jeu d'instruction est réduit pour simplifier l'architecture et faciliter la mise en pipeline :

- Instructions simples au format fixe
- 2 instructions pour l'accès à la mémoire : LOAD STORE

TELECOM PARIS
école nationale supérieure des télécommunications

Dans la fin des années 1980, il est devenu plus facile d'intégrer de la mémoire rapide sous forme de bancs de registre, mémoire locale ou mémoire cache au sein même du processeur. Le micro-séquencement des instructions ne se justifiait plus mais les instructions devait être assez simples pour être exécutées le plus vite possible, c'est à dire en 1 cycle. Le concept "Reduced Instruction Set Computer" était né.

Il n'en demeure pas moins qu'il faut toujours les mêmes étapes pour exécuter une instruction RISC :

• Aller la chercher en mémoire

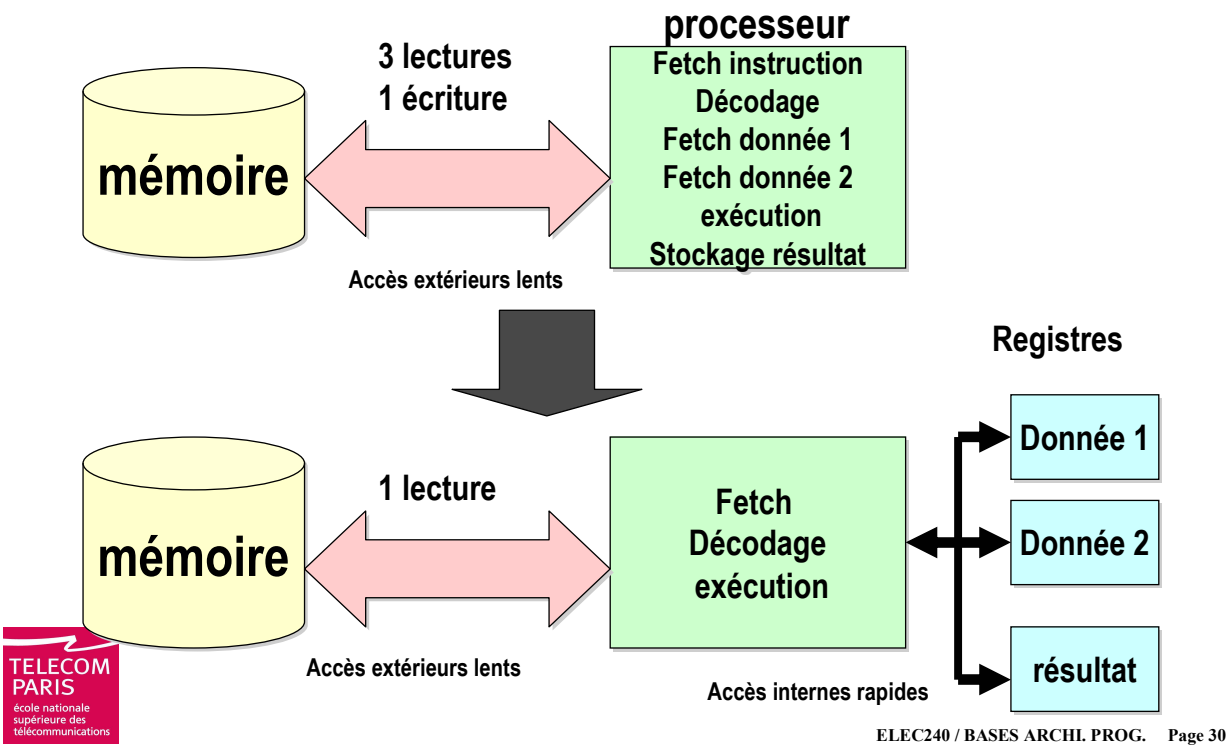
• La décoder

• L'exécuter

Un façon simple de "paralléliser" ces étapes est de les mettre en pipeline comme il va être vu par la suite.

D'autre part les calculs effectués sur des registres permettent de simplifier l'architecture et de faciliter le respect des performances. Pour charger ces registres ou sauver leur valeurs en mémoire, il suffit d'avoir respectivement 2 instructions simples LOAD et STORE.

Utilisation de registres locaux

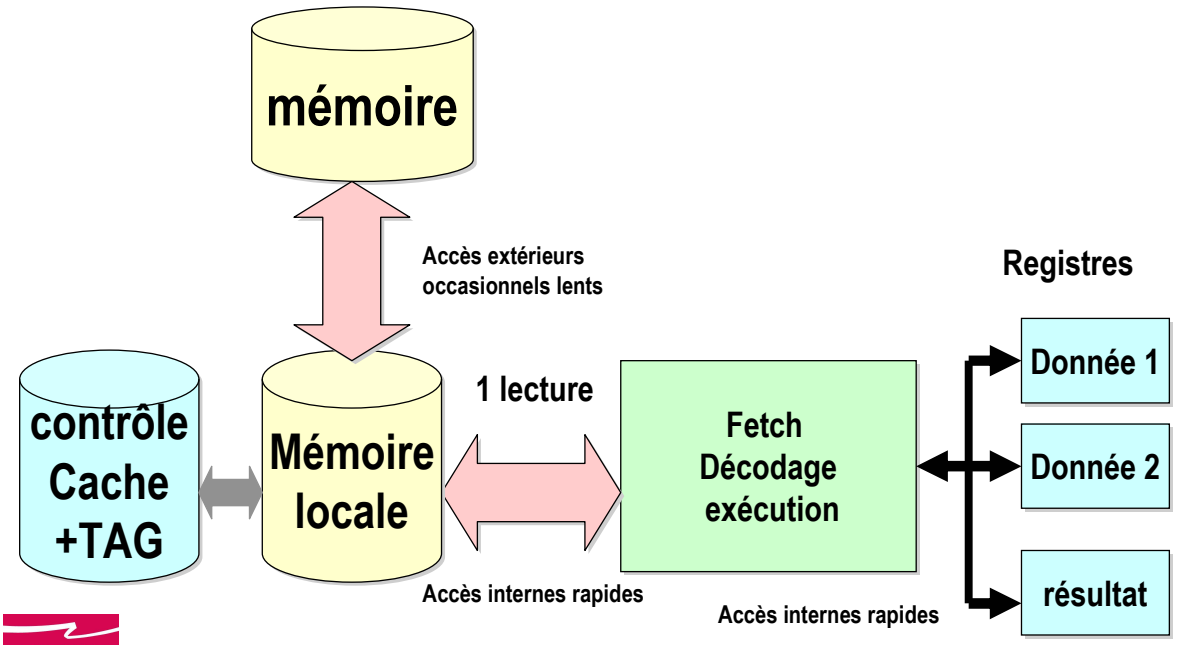


Dans l'exemple du haut, les données sont stockées en mémoire extérieure avec le programme (Von Neuman). Il faut 3 cycles de lecture (1 instruction et 2 données) et 1 cycle d'écriture (résultat).

Dans l'exemple du bas , les données sont préalablement stockées dans les registres. Dans ce cas il suffit d'une lecture de l'instruction.

Bien sûr les registres doivent être préalablement chargés ou lus avec des LOAD et des STORE, mais si on considère des opérations chaînées et un grand nombre de registres, il y a une perte de temps faible pour préparer le calcul.

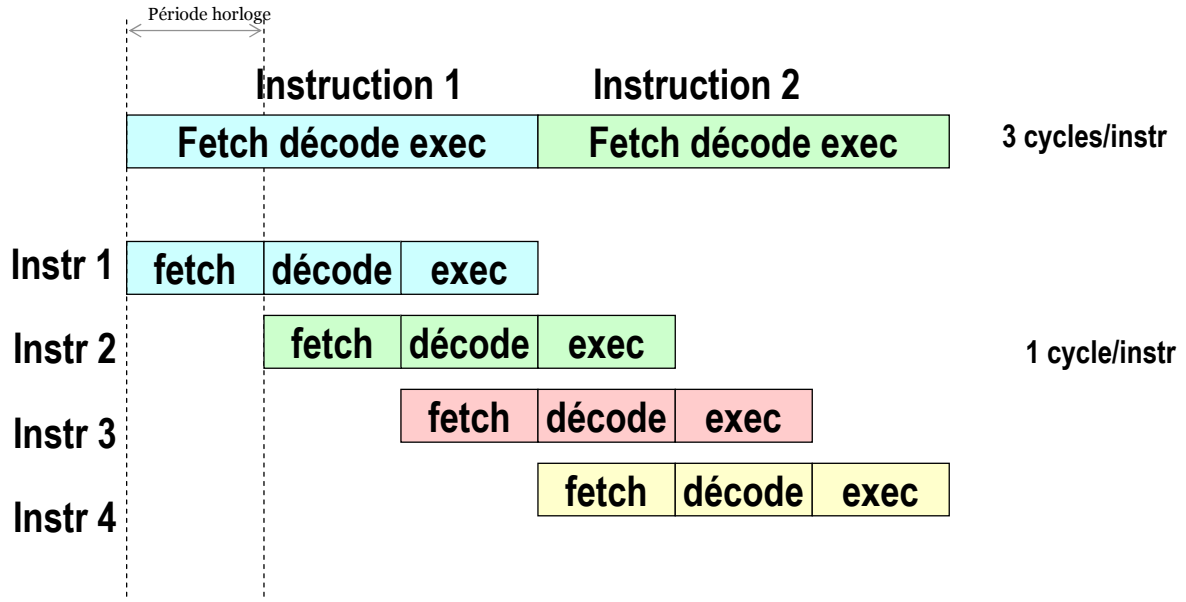
Utilisation de mémoire locale



TELECOM PARIS
école nationale supérieure des télécommunications

Les accès à la mémoire extérieure peuvent être accélérés avec l'utilisation d'une mémoire cache qui est par essence rapide et n'introduit pas de cycles d'attente.

Utilisation du pipeline



TELECOM PARIS
école nationale supérieure des télécommunications

Le pipeline d'instruction permet d'exécuter une nouvelle instruction à chaque cycle d'horloge.

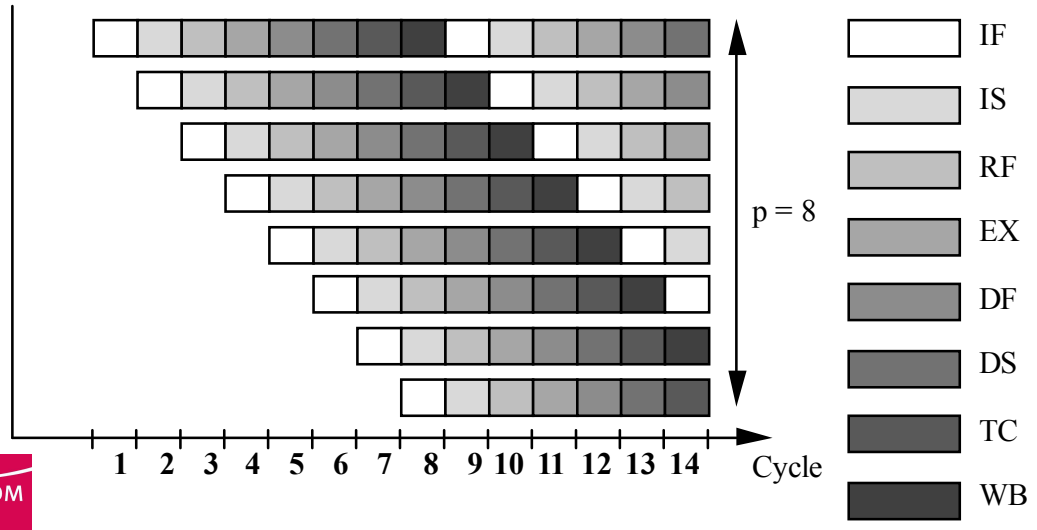
Dans cet exemple, il y a 3 étapes : **Fetch**, **Decode** et **Execute**.

Lors de l'étape 3, l'instruction 3 est dans la phase Fetch, alors que l'instruction 2 est décodée et l'instruction 1 est exécutée. Au cycle d'après c'est l'instruction 2 qui sera exécutée.

Les instruction ont toujours besoin de 3 étapes mais le débit d'instruction est au rythme de l'horloge.

Architectures Super-pipeline

- Très grandes fréquences d'horloge



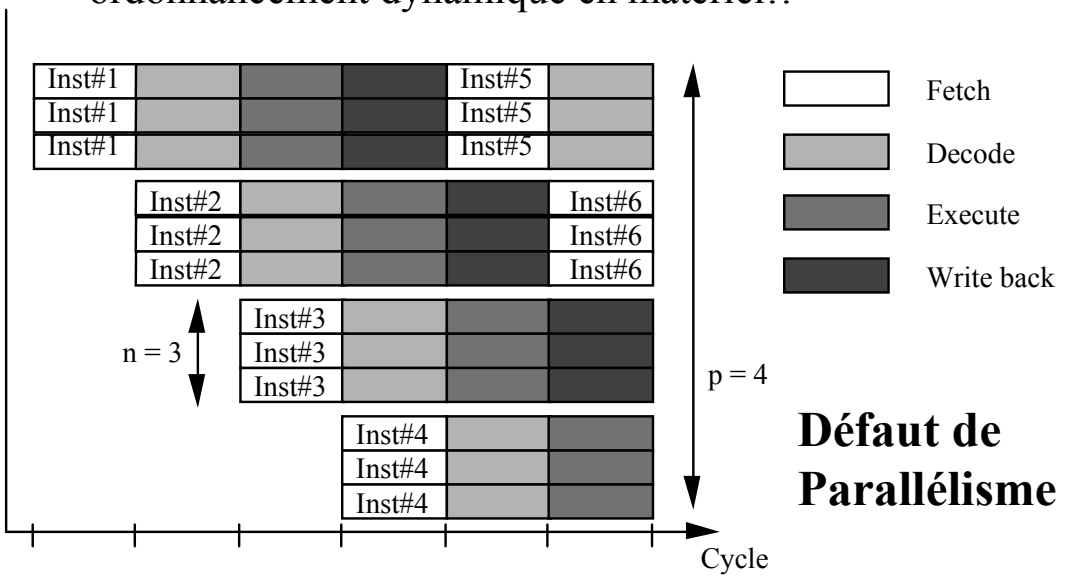
TELECOM PARIS
école nationale supérieure des télécommunications

Certains processeurs comme le pentium ont de nombreux niveaux de pipeline (>20). Cette technique "superpipeline" permet de diminuer les temps critiques et donc d'augmenter la fréquence de fonctionnement.

En revanche comme nous allons le voir par la suite, la rupture du pipeline est pénalisante.

Architectures Super-scalaires

- n unités d'exécution
- n instructions délivrées par cycle
- ordonnancement dynamique en matériel!!

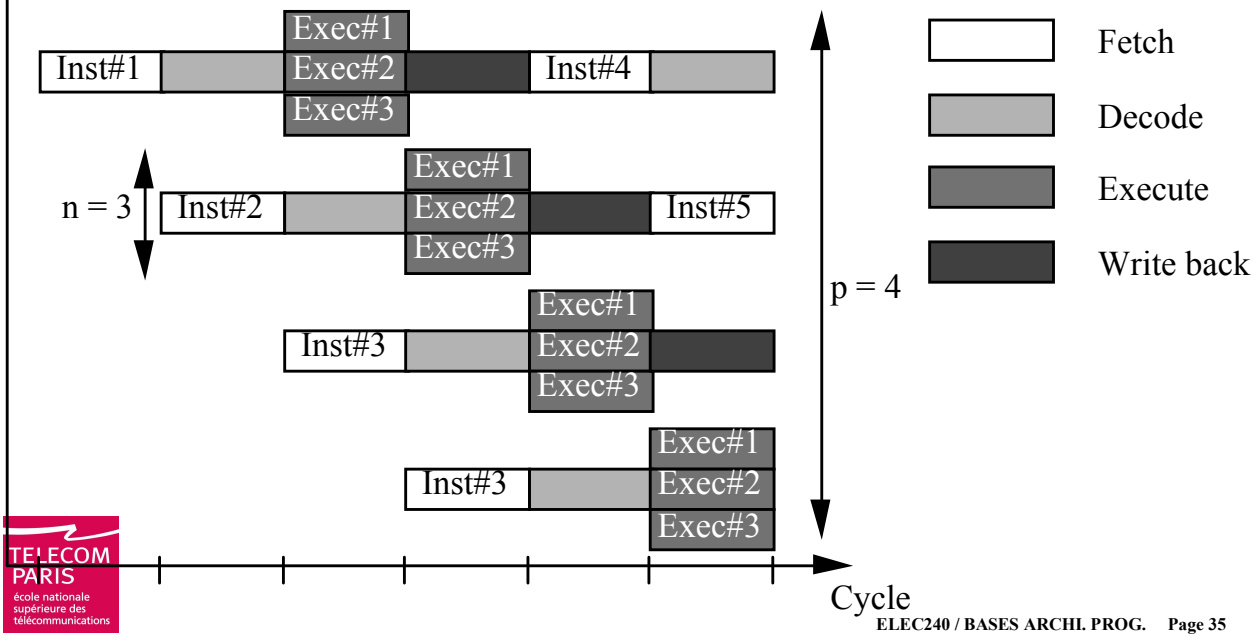


Défaut de Parallélisme ?

Cette technique vise à mettre en parallèle plusieurs processeurs en parallèle . Cette technique "superscalaire" repose sur une unité interne d'ordonnancement dynamique des instructions de façon à profiter pleinement du parallélisme.

VLIW

•Une seule unité de contrôle, une instruction très longue par cycle.



Une autre technique consiste à disposer d'unités d'exécution en parallèle mais d'agencer les instructions respectives au niveau du compilateur de telle sorte qu'une longue instruction correspondant à plusieurs instructions concaténées soit présentée au processeur. Cet agencement statique à la compilation permet en théorie d'atteindre des performances élevées par rapport aux techniques "superscalaires". Cependant l'écriture de tels compilateurs demeure très difficile et les résultats restent perfectibles, à moins d'utiliser des fonctions optimisées écrites en langage machine.

Architecture VLIW

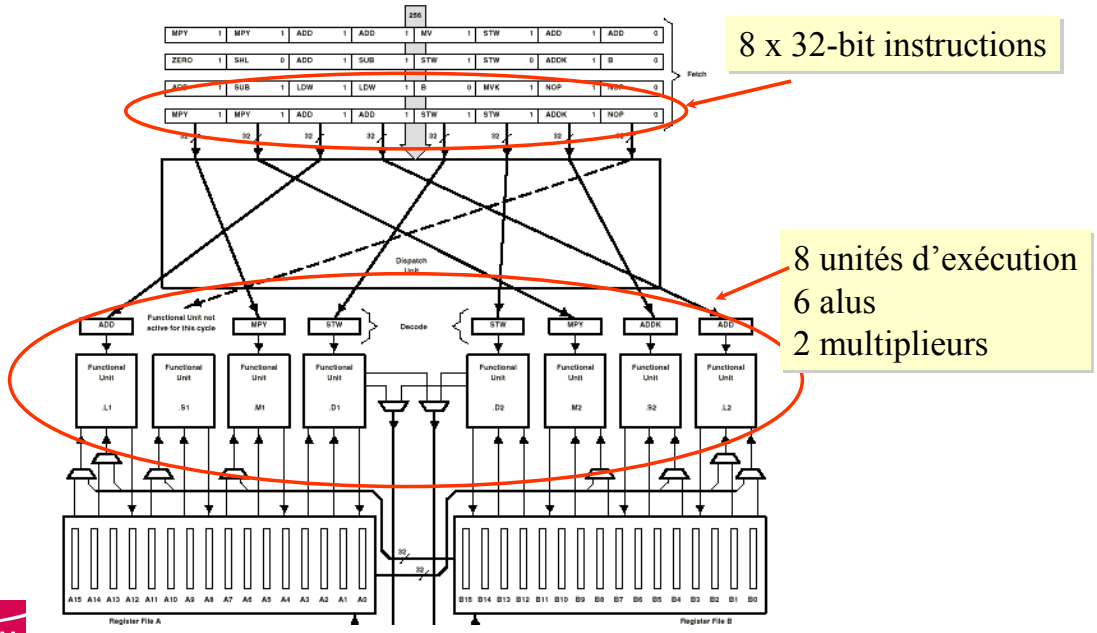
- **Formats fixes de l'instruction (champs indépendants)**
- **Ordonnancement statique à la compilation**
- **Les + :**
  - *densité d'intégration accrue : surface de silicium difficile à exploiter avec le superpipeline/scalaire*
  - *complexité "exponentielle" du contrôle dynamique de grands pipeline et/ou d'un parallélisme instruction important*
  - *croissance des applications de type DSP.*

TELECOM PARIS
école nationale supérieure des télécommunications

La technique VLIW est très utilisées dans les processeurs de traitement du signal car elle permet de réduire la surface silicium, donc le coût, et offre des performances très rapides si on dispose de bibliothèques écrites en assembleur pour pouvoir profiter du parallélisme.

Les processeurs généralistes (pentium, AMD) sont soumis à des fortes contraintes de compatibilité et de généricité permettant difficilement ce type d'approche.

Exemple : TMS320C6201



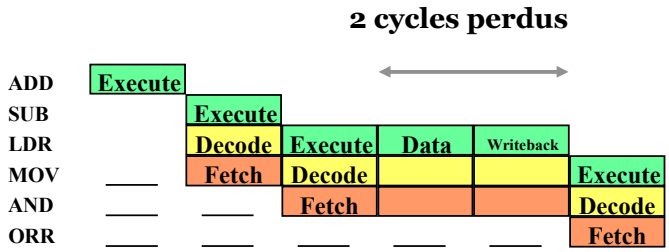
TELECOM PARIS
école nationale supérieure des télécommunications

Le DSP C6X de Texas Instruments sont les précurseurs des architectures VLIW. Ils présentent 8 unités de calculs en parallèle.

Problème : rupture du pipeline

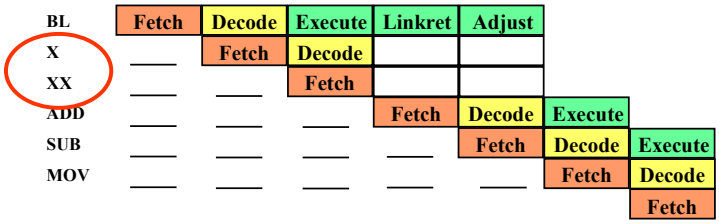
2 causes principales :

Accès mémoire extérieure



branchements

2 instructions perdues



Solutions : branchement retardé
prédiction de branchement

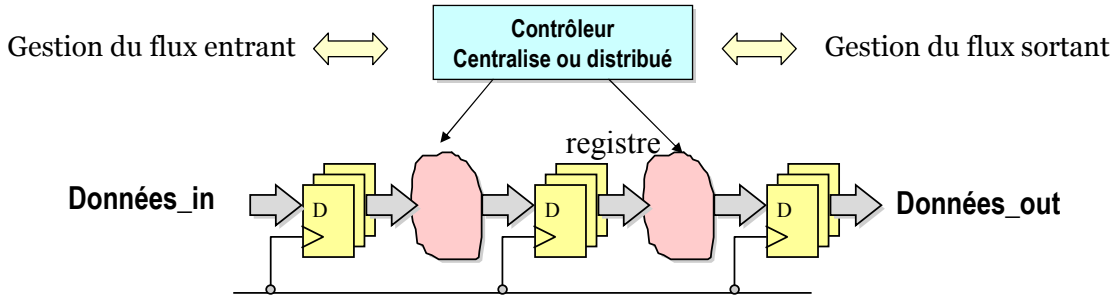
TELECOM PARIS
école nationale supérieure des télécommunications

Le pipeline peut rompre pour différentes raisons. Voici quelques cas importants :

L'exécution d'un LOAD ou STORE : Dans une architecture Von Neuman, il faut un cycle pour présenter l'adresse de la donnée de façon à ce que celle-ci soit transférée du processeur vers la mémoire. D'autre part il faut 1 cycle en plus car le temps entre la mémoire et le registre est trop long. Plutôt que de réduire la fréquence d'horloge, il est préférable de rajouter un registre tampon et donc 1 cycle .

L'exécution d'un branchement : Le branchement s'effectue à l'issue du cycle d'exécution. Les N-1 instructions (N étant le nombre d'étages de pipeline), qui suivent le branchement ne sont pas exécutés si le branchement a lieu. Certains processeurs utilisent l'unité d'exécution pour effectuer des opérations de sauvegarde de l'adresse de retour dans un registre, dans le cas d'un appel à un sous programme.

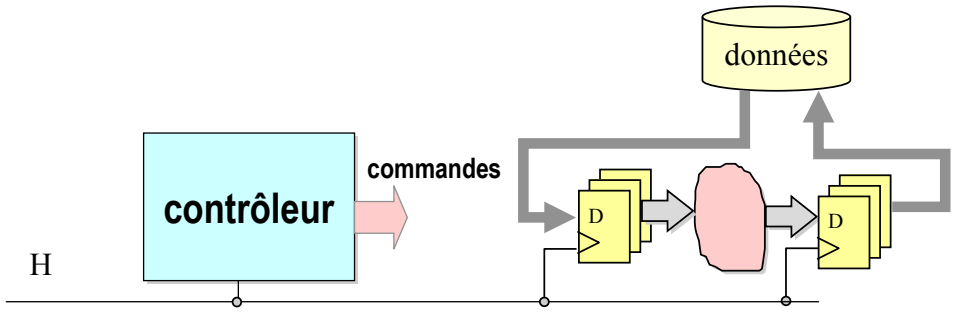
Partie opérative « data flow »



- Traitement en pipeline
- Contrôle simple

Dans le cas d'un traitement d'un flux de données, les structures en pipeline sont appropriées. Le contrôle est en général assez simple. Il peut lui-même être distribué en pipeline et être associé à chaque opérateur.

Partie opérative « Control Driven »



- Contrôleur centralisé complexe
- séquentialisation des opérations

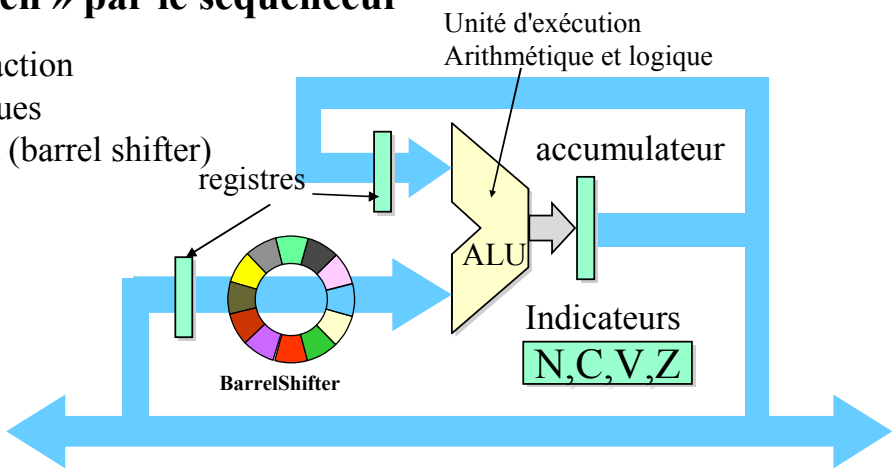
TELECOM PARIS
école nationale supérieure des télécommunications

L'approche "control driven" consiste à piloter le cœur opératif par un contrôleur unique qui décide du type d'opération et de l'aiguillage des données. Cette approche est celle qu'on retrouve dans les calculateurs génériques et les microprocesseurs avec l'unité arithmétique et logique.

Partie opérative d'un processeur

« Control Driven » par le séquenceur

- Addition-soustraction
- opérateurs logiques
- Décalages de bit (barrel shifter)



opérateurs plus spécifiques :

- Multiplication-accumulation (DSP)
- ALU + Multiplication en réel
- Calculs vectoriels

TELECOM PARIS
école nationale supérieure des télécommunications

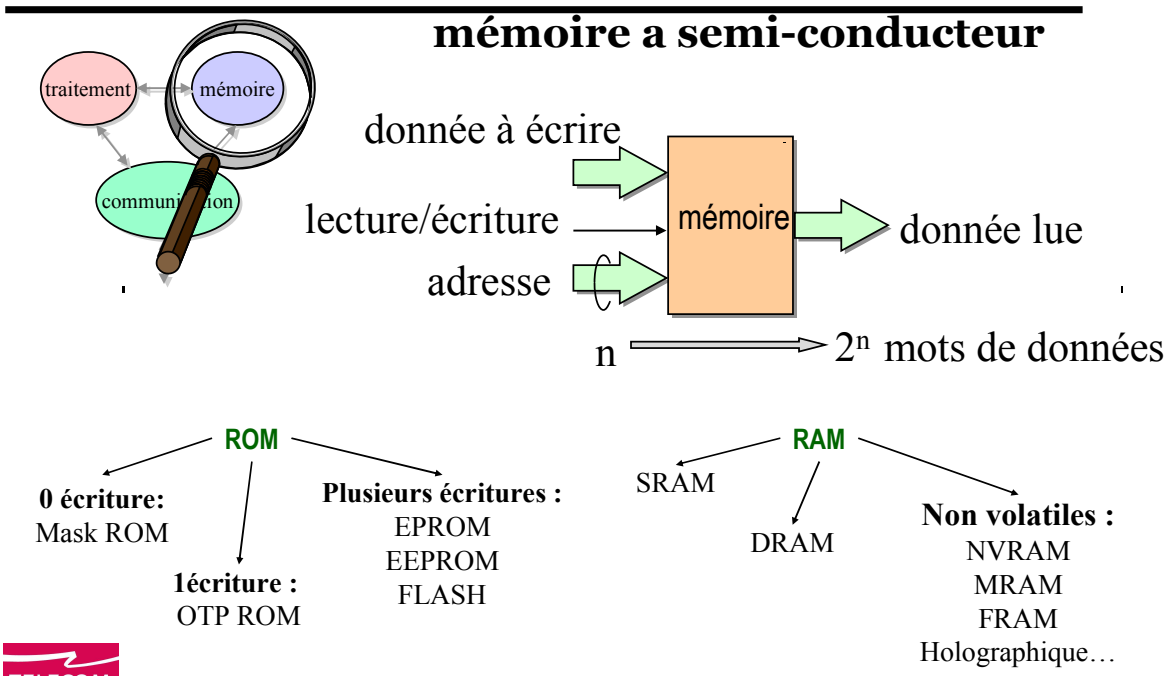
L'architecture classique de l'Unité arithmétique et logique UAL contient un registre accumulateur qui sert lui-même d'entrée à l'UAL pour la prochaine opération. Les microprocesseurs RISC ont la possibilité d'utiliser un registre quelconque comme accumulateur. D'autre part un des opérandes peut passer préalablement dans un "barrelshifter" avant de rentrer sur l'UAL. Cet opérateur effectue une rotation sur les bits de l'opérande, ce qui permet des multiplications ou divisions par des puissances de 2.

En sortie de l'UAL des indicateurs indiquent une propriété du résultat. Les indicateurs les plus courants sont :

- Z = résultat à 0
- N= 1 résultat négatif
- V = débordement
- C = retenu sortante

Les indicateurs sont utilisés par le processeur pour conditionner les instructions de saut.

Mémoire



TELECOM PARIS
école nationale supérieure des télécommunications

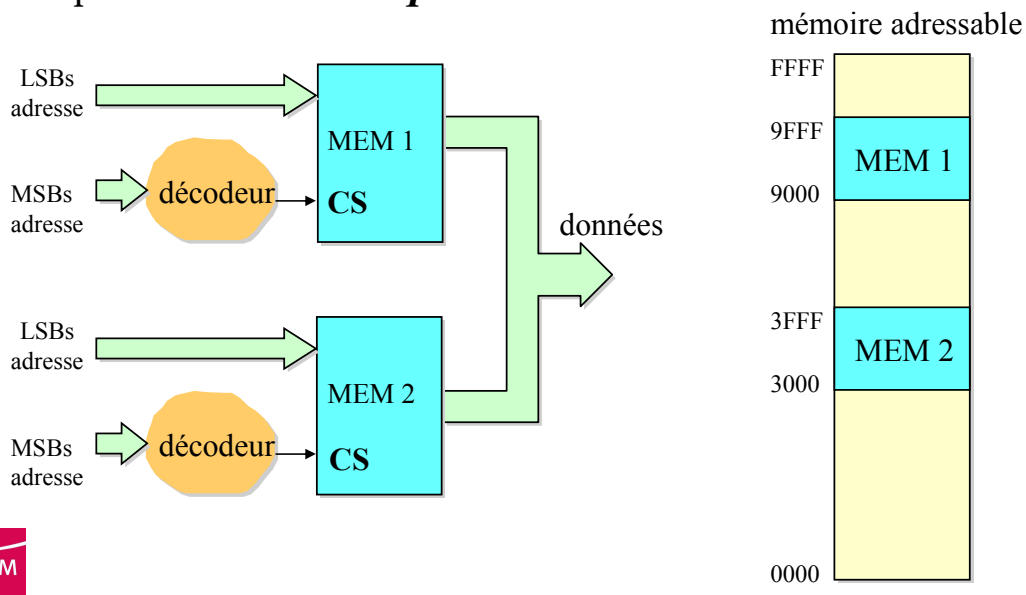
La mémoire électronique stocke 2^n mots de m bits. Un seul mot est accédé à la fois. Chaque mot est donc associé à une adresse. La mémoire agit comme une fonction combinatoire sur les bits d'adresse en lecture. En écriture il faut de plus activer le signal de commande d'écriture et lui fournir la donnée à écrire.

Les mémoires récentes sont synchrones et ont une horloge en entrée. Cela simplifie beaucoup le contrôle qui doit juste satisfaire la contrainte du chemin critique par rapport à la période d'horloge : Tcrit < Th

Mappe mémoire

indique les champs d'adresse mémoire dans un système

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



TELECOM PARIS
école nationale supérieure des télécommunications

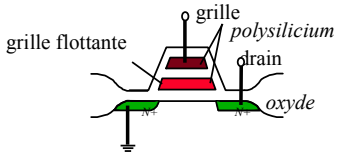
Un système électronique utilisant un champ mémoire, comme un microprocesseur, doit avoir une mappe mémoire correspondant aux zones d'adresse des différents éléments.

Une façon de dissocier 2 éléments, comme par exemple des boîtiers mémoire, est de piloter différemment un signal d'entrée souvent appelé "chip select" CS. Ce CS fait l'objet d'un décodage sur les bits de fort poids MSBs de l'adresse alors que le boîtier reçoit les bits de faible poids.

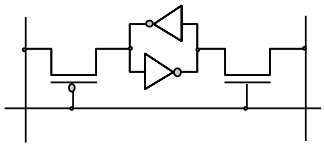
Si on considère un système disposant de 16 bits d'adresse, la mappe mémoire se situe entre 0x0000 et OxFFFF. Si on veut qu'une mémoire de 4K octets (donc 12 bits d'adresse) soit située dans la zone 0x9000 à 0x9FFF, Le CS doit correspondre au décodage des 4 MSBs à la valeur 9.

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

Les principales technologies sont la mémoire **ROM** ou plus récemment **FLASH** , la RAM statique **SRAM** et la ram dynamique **DRAM**.

La mémoire FLASH utilise des transistors double grille avec grille flottante. Les charges injectées dans la grille flottante sont capturées et peuvent rester quelques années. La mémoire est dite non volatile.

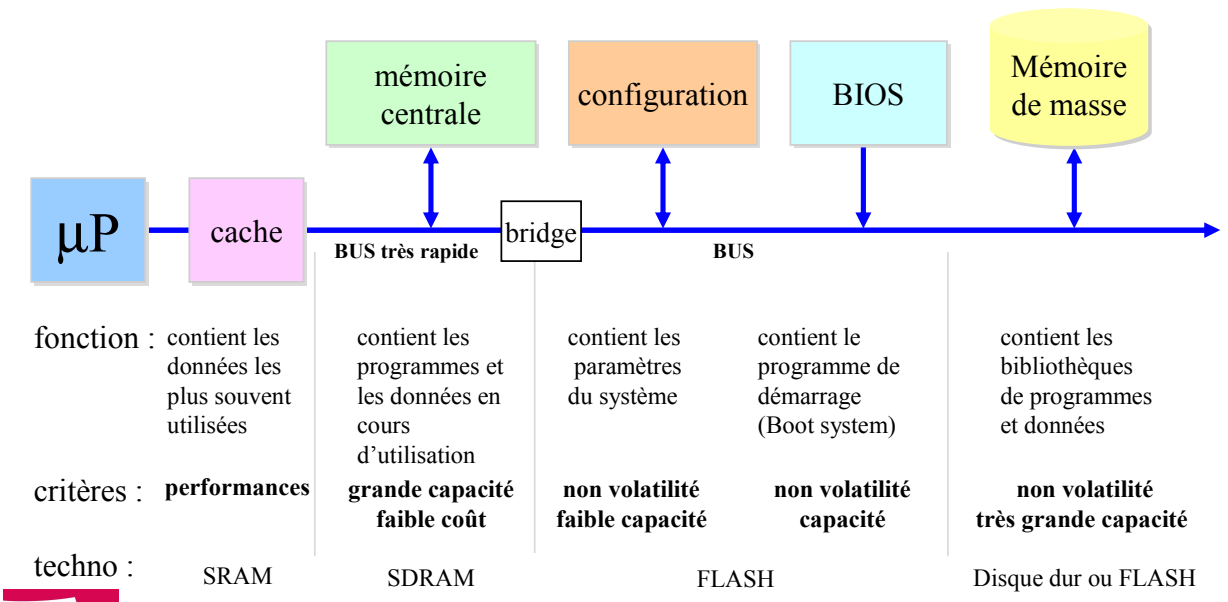
La mémoire SRAM repose sur un système stable assuré avec 2 inverseurs rebouclés sur eux-mêmes. 2 transistors de commande de part et d'autre permettent l'écriture et la lecture.

La mémoire DRAM est la capacité grille d'un transistor MOS. Cette petite capacité se décharge assez vite et un système de rafraîchissement est nécessaire.

Les mémoires SRAM et DRAM sont volatiles. La mémoire DRAM est optimale en terme de taille (1 point=1 transistor) mais est plus lente.

De gros progrès ont été fait sur les interfaces des mémoires de façon à accélérer les débits. Par exemple la mémoire FLASH est écrite par secteurs de données. De même la mémoire DRAM, assez lente en accès aléatoire devient rapide si toute une rangée est préchargée dans une mémoire rapide et accédée en rafale.

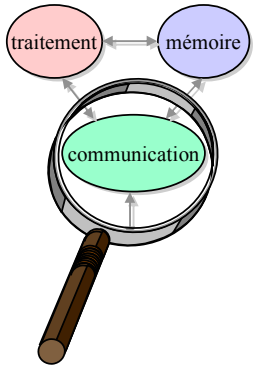
Mémoire dans un système informatique

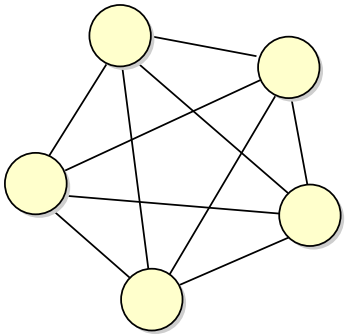


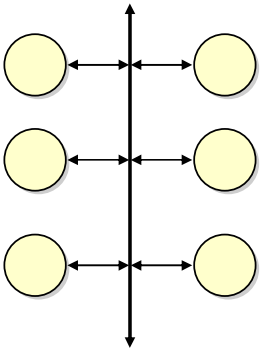
TELECOM PARIS
école nationale supérieure des télécommunications

Un système électronique peut utiliser différentes variétés de mémoires en fonction des besoins. Par exemple dans un ordinateur la SRAM rapide est utilisée comme cache et est gérée par un contrôleur de cache. La DRAM, de taille plus importante, est utilisée comme mémoire de programmes, la mémoire FLASH contient le BIOS pour le démarrage. La mémoire de masse utilise des technologies magnétiques , optiques ou électroniques avec l'augmentation en capacité des mémoires FLASH de type NAND FLASH

Transfert d'information local







point à point

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

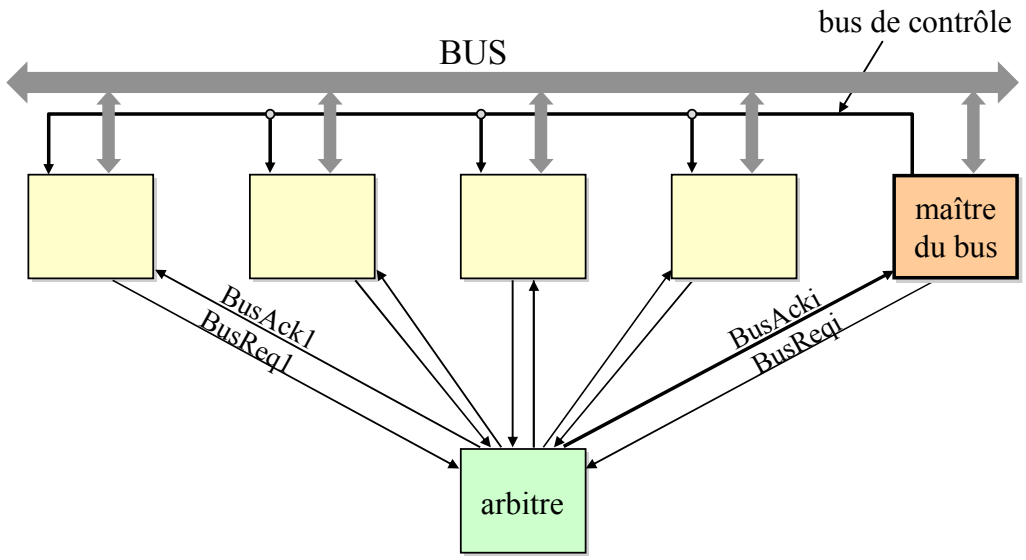
Bus

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

TELECOM PARIS
école nationale supérieure des télécommunications

Il existe de nombreuses façons de faire communiquer les éléments d'un système électronique entre eux. Les liaisons "**point à point**" sont simples à mettre en œuvre mais il faut de nombreuses liaisons pour chaque élément. La contrainte de modularité et d'intégration favorise les liaisons de type "**BUS**" ou un ensemble d'équipotentielles est commun à tous les éléments. Le prix à payer pour ce gros avantage est le fait qu'une seule communication est possible à la fois. D'autre part si plusieurs éléments veulent prendre le bus, il faut un arbitre qui décide des priorités à donner et indique le "**Maître**" du bus.

Arbitrage de bus

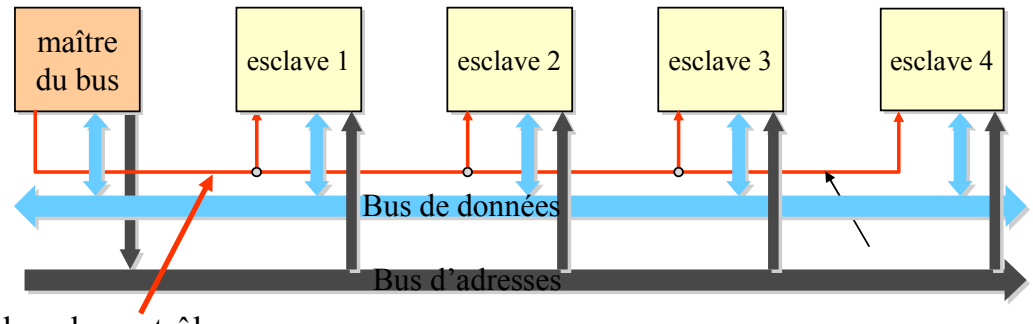


TELECOM PARIS
école nationale supérieure des télécommunications

L'arbitre reçoit des requêtes des maîtres potentiels et leur renvoie une autorisation s'il leur accorde le bus. Une seule autorisation ou "jeton" est donné. De nombreuses stratégies existent pour avoir des accès équitables et éviter les "famines". L'arbitre peut faire tourner le jeton, change le niveau de priorité si un maître accapare trop souvent le bus,…

Transfert sur un bus

Nécessité d'associer une adresse ou un champ d'adresse à un élément.
⇨ chaque élément a son propre décodeur d'adresse générant un "CS"



bus de contrôle :
- horloge
- activité du cycle de transfert
- direction des transferts (lecture/écriture)
- gestion du protocole d'échange

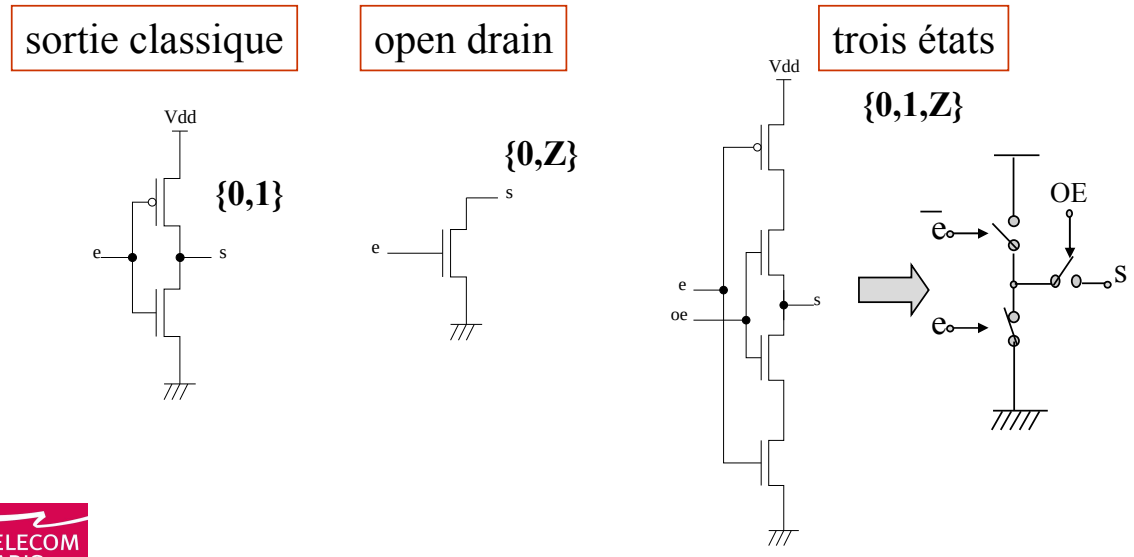
si maître = processeur
esclaves = périphériques

TELECOM PARIS
école nationale supérieure des télécommunications

Les éléments sur un bus ont leur propre champ d'adresse correspondant à l'activation de leur "Chip Sélect" . Quand un maître dispose du bus, il doit présenter l'adresse de l'"esclave" visé sur le bus d'adresse, le type de transaction sur le bus de commandes, et les données en écriture sur le bus de données (ou attendre les données s'il s'agit d'une lecture).

Sorties 3 états et "open drain"

Modèles fonctionnels des sorties de portes logiques :



TELECOM PARIS
école nationale supérieure des télécommunications

Les sorties d'un système électronique se font d'une façon traditionnelle en CMOS avec un réseau de transistors P tirant à 1 et un réseau dual de transistor N tirant à 0.

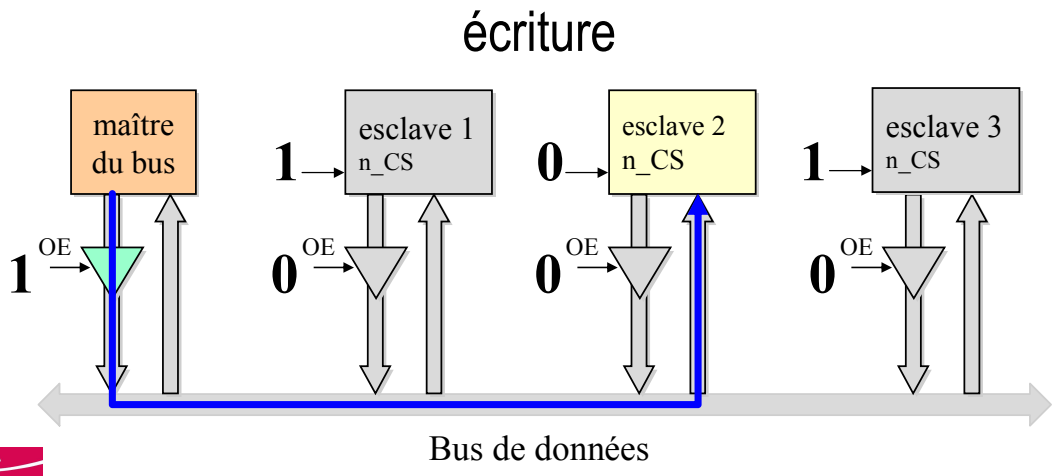
Pour relier les sorties entre elles, ce qui n'est pas possible en sortie CMOS, il peut être intéressant d'enlever le réseau de transistor P. La sortie prend 2 états 0 et Z; Le Z symbolise l'état "haute impédance" où la sortie n'est forcée à aucun potentiel. Ce type de sortie s'appelle sortie "**Open Drain**".

Une autre type de sortie est la sorties **3 états** qu'on peut symboliser comme un sortie CMOS avec un transistor supplémentaire en série pour permettre un 3ème état qui est Z.

Cette sortie a donc un signal de commande "Output Enable" OE autorisant soit la sortie CMOS classique à 2 états, soit la mise en état haute impédance Z. Ce type de sortie se dit aussi "buffer 3 états"

Intérêt du 3 états pour les bus

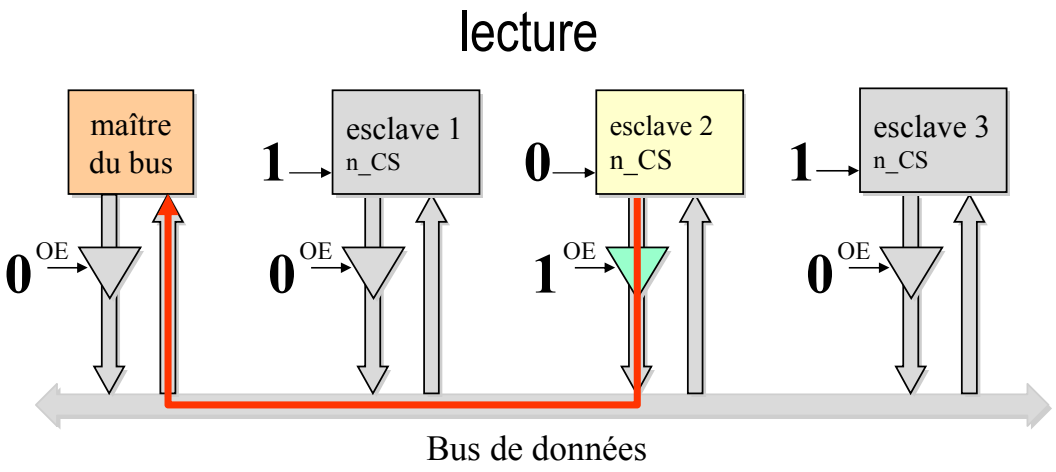
Nécessité d'une logique 3 états pour assurer la bidirectionnalité du bus :
⇒ 1 seul buffer doit être actif sinon conflit.



TELECOM PARIS
école nationale supérieure des télécommunications

Les sorties 3 états permettent les transferts bidirectionnels de données sur un bus.
Quand un maître écrit à l'esclave n°2, seul le sign al OE du maître doit être actif.

Intérêt du 3 états pour les bus



en résumé :

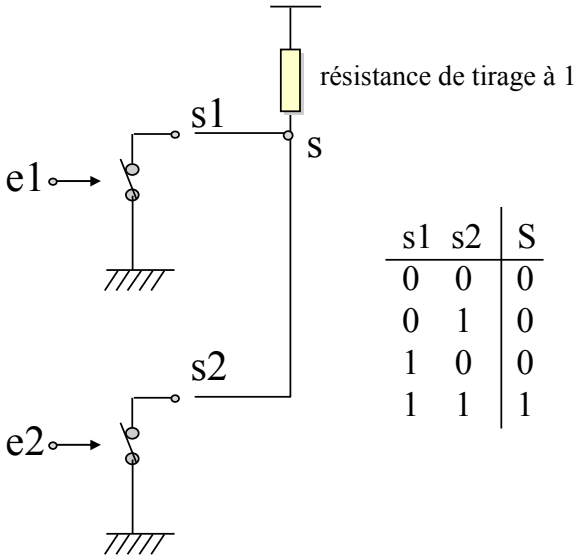
OE maître = écriture
OE esclave_i =lecture ET CS_i

TELECOM PARIS
école nationale supérieure des télécommunications

Quand le maître lit l'esclave n°2, seul l' OE de la sortie de l'esclave 2 est actif.

Intérêt de l' open drain

possibilité de connecter les sorties entre elles :



s1	s2	S
0	0	0
0	1	0
1	0	0
1	1	1

porte ET cablée

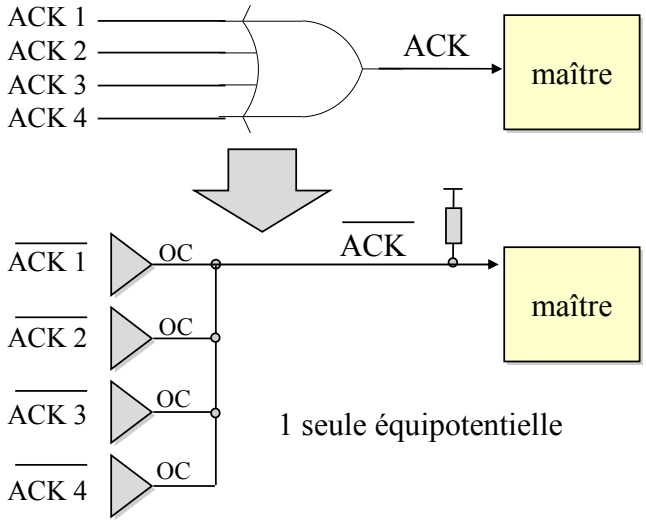
TELECOM PARIS
école nationale supérieure des télécommunications

Les sorties "Open Drain" permettent des connexions de plusieurs sorties. Si on relie la sortie à une résistance connectée à la tension d'alimentation Vdd, l'état Z correspondra à un potentiel Vdd (donc niveau logique '1') par le biais de cette résistance de "tirage" ou "pull up".

Le signal S résultant de cette connexion est équivalent à une porte ET dont les entrées sont S1 et S2. Cette propriété est très intéressante car elle permet de construire une porte ET avec beaucoup d'entrées sans porte CMOS. On parle de "ET câblé" pour le différencier de la porte CMOS.

Exemple de sorties en open drain

signal d'acquittement d'un bus



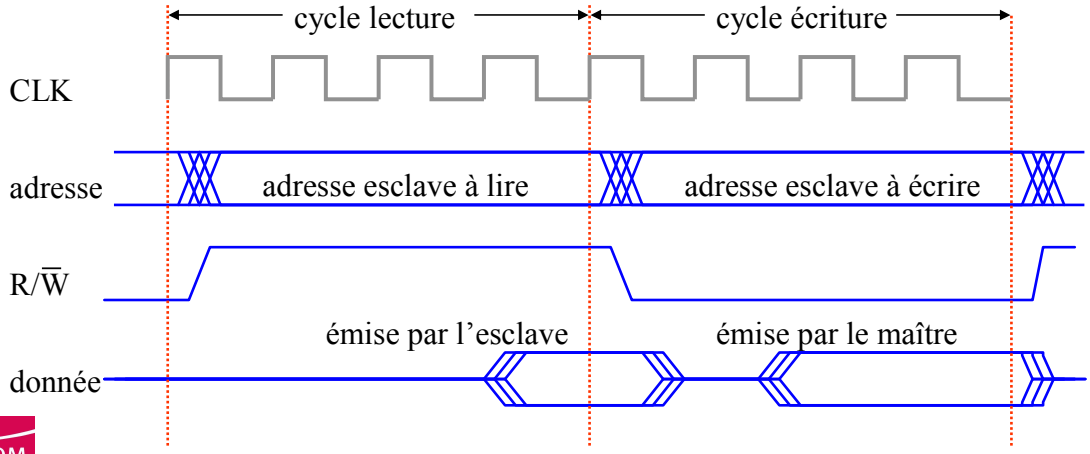
TELECOM PARIS
école nationale supérieure des télécommunications

Dans un bus ayant un protocole asynchrone, le nombre de signaux "accusé de réception" peut être très grand. De même un processeur peut avoir une ligne d'interruption générée par un grand nombre d'éléments périphériques. La fonctionnalité demandée est un OU logique faisant la somme des signaux issus de chaque élément. Ce OU doit avoir un nombre d'entrées variable et important.

L'utilisation des sortie "Open Drain" permet de réaliser un ET câblé qui devient un OU câblé si tous les signaux sont en logique inverse (utilisation du théorème de De Morgan).

Protocole de communication synchrone

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



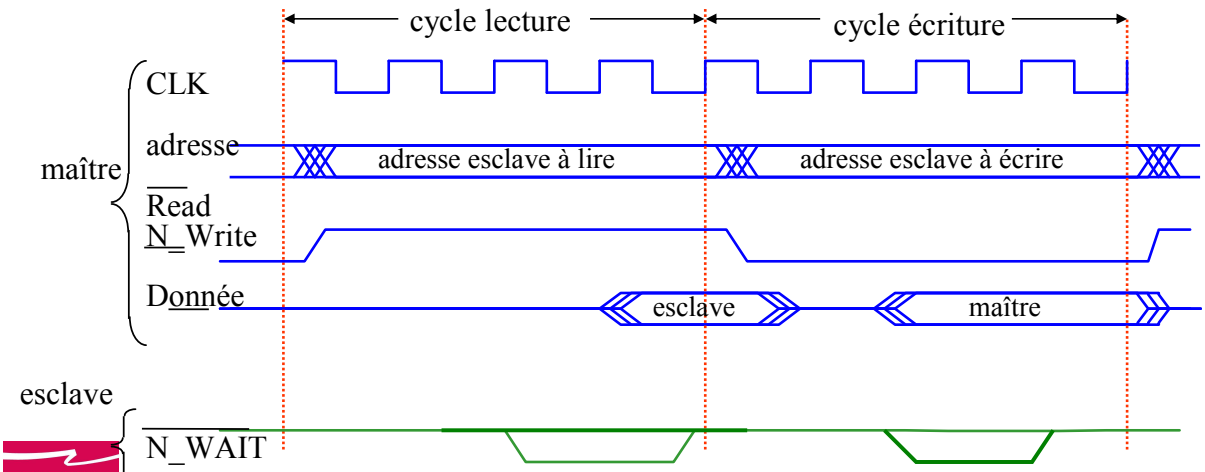
TELECOM PARIS
école nationale supérieure des télécommunications

Le protocole utilisé sur un bus est celui qu'on retrouve dans tous les systèmes de communication. Il est soit direct , c'est à dire sans préoccupation d'acceptation de l'élément adressé, ou **asynchrone** c'est-à-dire avec un échange "poignée de main" ou "Handshake" pour s'assurer de la fiabilité de la transaction.

Dans le cas de l'échange direct, on parle plutôt d'échange **synchrone** en électronique car l'accès dure un nombre constant de cycles d'horloge. Les 2 éléments en communication se sont mis d'accord à l'avance sur le nombre de cycles d'horloge nécessaire à la transaction.

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.



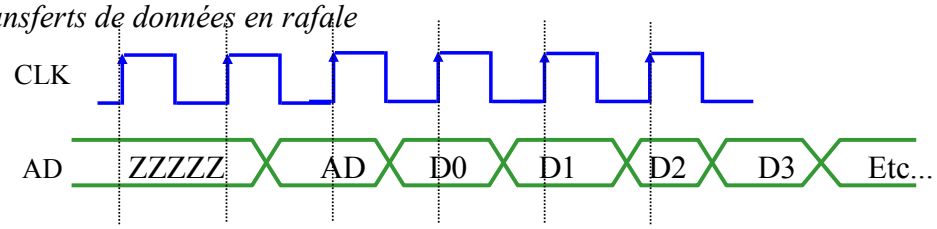
Le protocole asynchrone nécessite un signal "accusé de réception" ou "feu vert" de l'élément adressé vers le maître de façon à s'assurer que la transaction va bien se passer. Il permet à certains éléments lents de faire attendre le maître. Il existe 2 philosophies; Soit ce signal en retour dit : "Je suis prêt" , auquel cas il faut toujours que l'esclave génère ce signal à chaque transaction. Ou bien le signal en retour signifie "attendez !" auquel cas il n'est pas nécessaire que l'esclave génère le signal s'il est assez rapide.

Attention les protocole Asynchrone ne veut pas dire qu'il n'y a pas d'horloge. Dans l'exemple ci-dessus, il existe toujours une horloge car la logique est Synchrone mais la communication est Asynchrone.

Caractéristiques de bus

- Protocoles
- Débits max
  - *Transferts de données en rafale*

Exemple : PCI



- Liaison Série ou Parallèle
- Caractéristiques électriques (tension, courant, impédance)
- Configuration automatique au démarrage ou "à chaud"
- Isochronisme
- Interruptions

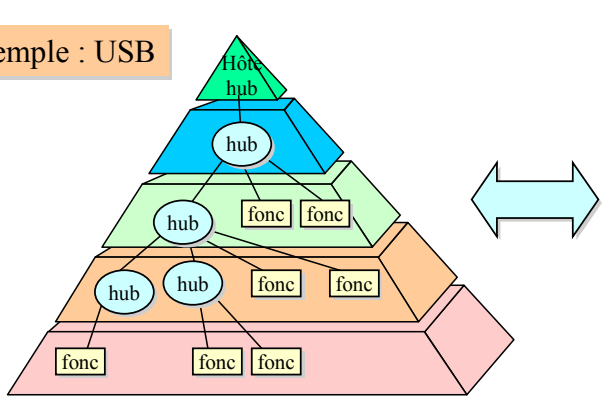
TELECOM PARIS
école nationale supérieure des télécommunications

Les bus se différencient par leur débits, leurs caractéristiques électriques, mécaniques, protocolaires,… Pour augmenter le débit, le mode rafale ou "burst" est communément employé. Il offre la possibilité de transmettre des paquets de données à raison d'une donnée par cycle d'horloge. Le bus peut respecter des contraintes d'isochronisme", c'est-à-dire que l'arbitre peut s'engager à maintenir un débit de communication. Il peut aussi y avoir des signaux d'interruption pour que certains éléments critiques signalent un évènement important. Ceci évite au maître du bus de scruter périodiquement cet élément (polling).

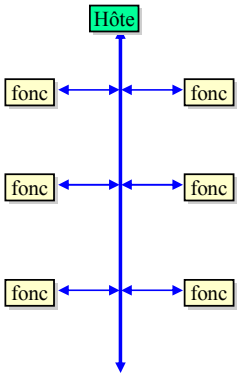
Topologie de bus

Certains bus peuvent avoir une topologie point à point au niveau physique et une topologie de type Bus au niveau "liaison"

Exemple : USB



Niveau physique



Niveau liaison

TELECOM PARIS
école nationale supérieure des télécommunications

Le bus USB a une topologie virtuelle de bus au niveau "liaison" de la communication mais sa couche physique est composée de liaisons point à point. Il n'existe qu'un seul maître qui est le processeur.

Quelques standards d'interface

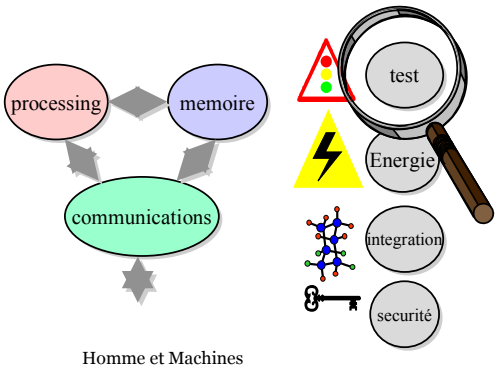
INTERFACE	FONCTION	DEBIT	DISTANCE	STANDARDS
Homme -Machine	rentrée de données	10^2 b/s	10^{-1} m	RS232, I^2C, USB, SPI
	lecture résultats	10^8 b/s	10^0m	DVI, IEEE1394
Machine -Machine	stockage	10^9 b/s	10^{-1} m	SATA, Fiber channel, IEEE1394
	réseau local	10^8 b/s	10^2 m	WIFI, ethernet
	réseau distant	10^7 b/s	10^4 m	ADSL

TELECOM PARIS
école nationale supérieure des télécommunications

Les interfaces de communication tirent parti des avancées technologiques aussi bien dans le domaine des semi-conducteurs que ceux des algorithmes de traitement du signal et de communications.

Par exemple l'interface visuelle homme-machine bénéficie des algorithmes de compression d'image et de synthèse d'images avec des processeurs graphiques vidéo disposant d'une très grande puissance de calcul. Les communications distantes entre machines se font à très haut débit grâce aux technologies CDMA (téléphone portable, WIFI) OFDM (WIFI, TNT, ADSL) et codage canal (Turbo-codes, LDPC) .

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*

TELECOM PARIS
école nationale supérieure des télécommunications

A côté des 3 blocs fonctionnels : Traitement, Mémoire et Interfaces, les blocs annexes ont d'un système électronique numérique visent à améliorer la fiabilité, la consommation faible, la compatibilité et la sécurité.

Le bloc de TEST est indispensable. Il permet de garantir la qualité du produit pour :

•, Faire des économies au niveau de la maintenance

• Avoir une image de produit de qualité (impact marketing fort)

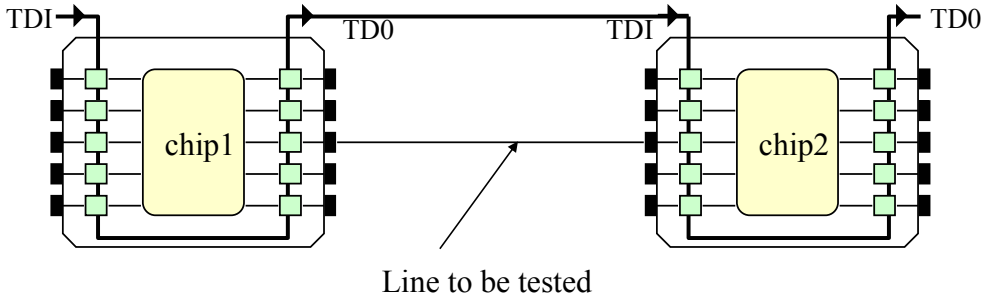
Pour un système électronique, l'architecture peut être modifiée de façon à rajouter des blocs de test spécifiques. Les stratégies de test reposent à la fois sur des tests fonctionnels (très longs) et sur des tests d'intégrité électrique vérifiant que chaque équipotentielle ne soit pas coupée en ou court-circuit. Une technique courante de vérification de l'intégrité électrique consiste à chaîner les bascules D de façon à obtenir un grand registre à décalage "scan chain". Par ce biais il est possible de rentrer des valeurs au circuit à tester puis de récupérer les résultats sur les bascules échantillonnant le résultat. Les outils de CAO sont capables de générer automatiquement les vecteurs de test pour qu'un maximum d'équipotentielles passe d'un niveau logique à l'autre et que le test soit le plus exhaustif possible.

Port JTAG

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

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

exemple de Boundary Scan Test :

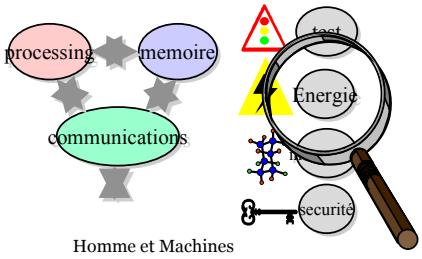


TELECOM PARIS
école nationale supérieure des télécommunications

Le port JTAG "Joint Test Action Group" est une contrôleur gérant un flux de données série (TDI en entrée et TDO en sortie) permettant d'effectuer diverses opérations dont celle du test pour lequel il a été conçu à l'origine.

L'opération de test appelée "Boundary Scan test" permet de tester la connectivité des composants assemblés entre eux. Chaque composant a ses broches d'entrée/sortie reliées par un grand registre à décalage qui est lui même relié par les broches TDI et TDO à un autre circuit. Le tout constituant un énorme registre à décalage. Les contrôleurs JTAG ou "TAP controller" effectuent des opérations d'écriture ou de lecture dans ces registres. Il est donc possible de forcer un niveau logique sur une broche de sortie et de lire le niveau sur la branche d'entrée de la même équipotentielle d'un autre composant. Si les niveaux coïncident, l'équipotentielle est considérée intègre.

Contrainte de consommation d'énergie



Consommation dynamique

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

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

Vdd ↘ f ↗ C ↗ ⇨ P ↗

Consommation statique

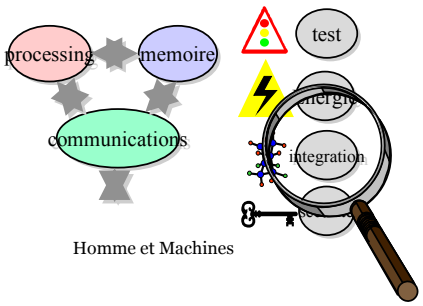
- ***Due aux courants de fuite. Quasi nulle dans les technologies >130nM mais devient non négligeable***

TELECOM PARIS
école nationale supérieure des télécommunications

La consommation est majoritairement due à la décharge et charge des capacités lorsque les transistors d'une paire CMOS commutent. La puissance moyenne dissipée est donc proportionnelle à la fréquence moyenne d'activité des équipotentielles donc à la fréquence d'horloge du système synchrone.

La puissance statique est quasi-nulle. Dans les technologies inférieures à 100nm la proportion de la consommation statique due aux courants de fuite devient perceptible.

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 (réfléxion)*

TELECOM PARIS
école nationale supérieure des télécommunications

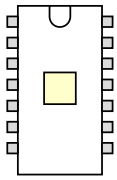
L'intégration dans un environnement se fait à tous les niveaux : mécanique, électrique, protocolaire, logiciel,… Ces contraintes sont à considérer depuis la spécification du système jusqu'à l'assemblage final.

Concernant l'assemblage sur circuit imprimé qui est constitué de plusieurs couches de routage en cuivre, il existe de fortes contraintes de place, d'intégrité du signal et de rayonnement électromagnétique (CEM). Il est donc très important de vérifier l'assemblage. Il faut d'abord effectuer le placement et le routage des composants puis analyser statiquement les critères de transmission sur ligne, d'interférences et de rayonnement pour vérifier que le niveau de perturbations n'est pas critique.

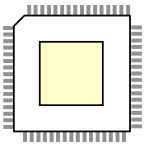
Ce sont les mêmes problèmes rencontrés dans les circuits intégrés mais avec des dimensions moindres.

Encapsulation des circuits

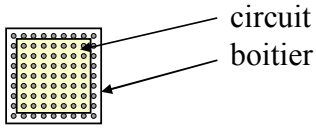
- **Plus d'E/S* en moins d'espace**
  - *Réduire l'espace entre broches "pitch". : 2,54mm => 0,5 mmm*
  - *Utiliser une matrice de broches plutôt que sur la périphérie du boitier*
  - *Utiliser les circuits "FlipChip" où les E/S sont internes à la puce*



DIP (Dual In Line Package)



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



BGA (Ball Grid Array)
pitch=1mmm

- **Dissipation de puissance**
  - *Boitier céramique*
  - *Système de refroidissement*

*** 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~0,6

TELECOM PARIS
école nationale supérieure des télécommunications

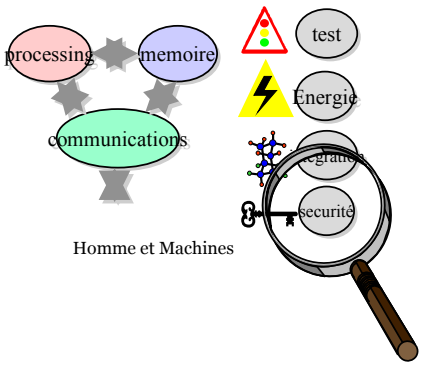
Les boîtiers ont beaucoup évolué pour tenir compte de l'augmentation de la densité de portes et de l'augmentation du nombre d'entrées/sorties E/S, comme l'indique la loi empirique de Rent.

Les premiers boîtiers étaient rectangulaires et disposaient de broches sur 2 côtés (boîtiers DIP Dual In Line Package). Le fait que les broches traversent l'isolant du circuit imprimé diminue la "routabilité" car toutes les couches du circuit imprimé sont pénalisées. Pour remédier à cela les boîtiers ont évolué pour avoir des broches soudables uniquement en surface (composants montés en surface CMS , Surface mounted devices SMD)

Avec l'augmentation des E/S, les broches se sont installées sur la périphérie de boîtiers carrés (Quad Flat Package, 240 broches maximum), donc avec un meilleur facteur de forme. La faible distance entre broches de l'ordre de 0,5mm diminue la fiabilité des connexions.

Les boîtiers récents disposent de broches disposées matriciellement sous le circuit permettant à la fois une augmentation du nombre de broches et une distance entre broches supérieure à 0,5mm (BGA : Ball Grid Array)

Contrainte de sécurité



- **Eviter le copiage**
  - ***Utiliser du matériel dédié***
- **Protéger les communications**
  - ***Cryptographie : Algorithmes très efficaces mais cryptanalyse aussi. Matériel vulnérable car fuit de l'information par le biais de canaux physiques.***
  - ***Utiliser du matériel résistant aux "attaques"***

TELECOM PARIS
école nationale supérieure des télécommunications

La sécurité est à 2 niveaux :

• La protection contre les copiages

• La protection contre l'intrusion ou l'espionnage des messages véhiculés

Pour éviter le copiage , le simple fait de choisir la conception d'un circuit personnalisé constitue une protection par rapport à une solution logicielle. Pour les FPGAs, il existe la possibilité de chiffrer la netlist.

Pour la protection des communications, les algorithmes cryptographiques sont largement utilisés. Il existe cependant des vulnérabilités potentielles au niveau matériel du fait que les fonctions électroniques dissipent de l'énergie reflétant une partie du secret à protéger. Ces "attaques" sur le matériel sont difficiles à réaliser et peuvent être contrecarrées par des architectures résistantes en cours de mise au point dans des laboratoires de recherche.