ensiie
école nationale supérieure d'informatique
pour l'industrie et l'entreprise

Architecture des ordinateurs

G.BERTHELOT

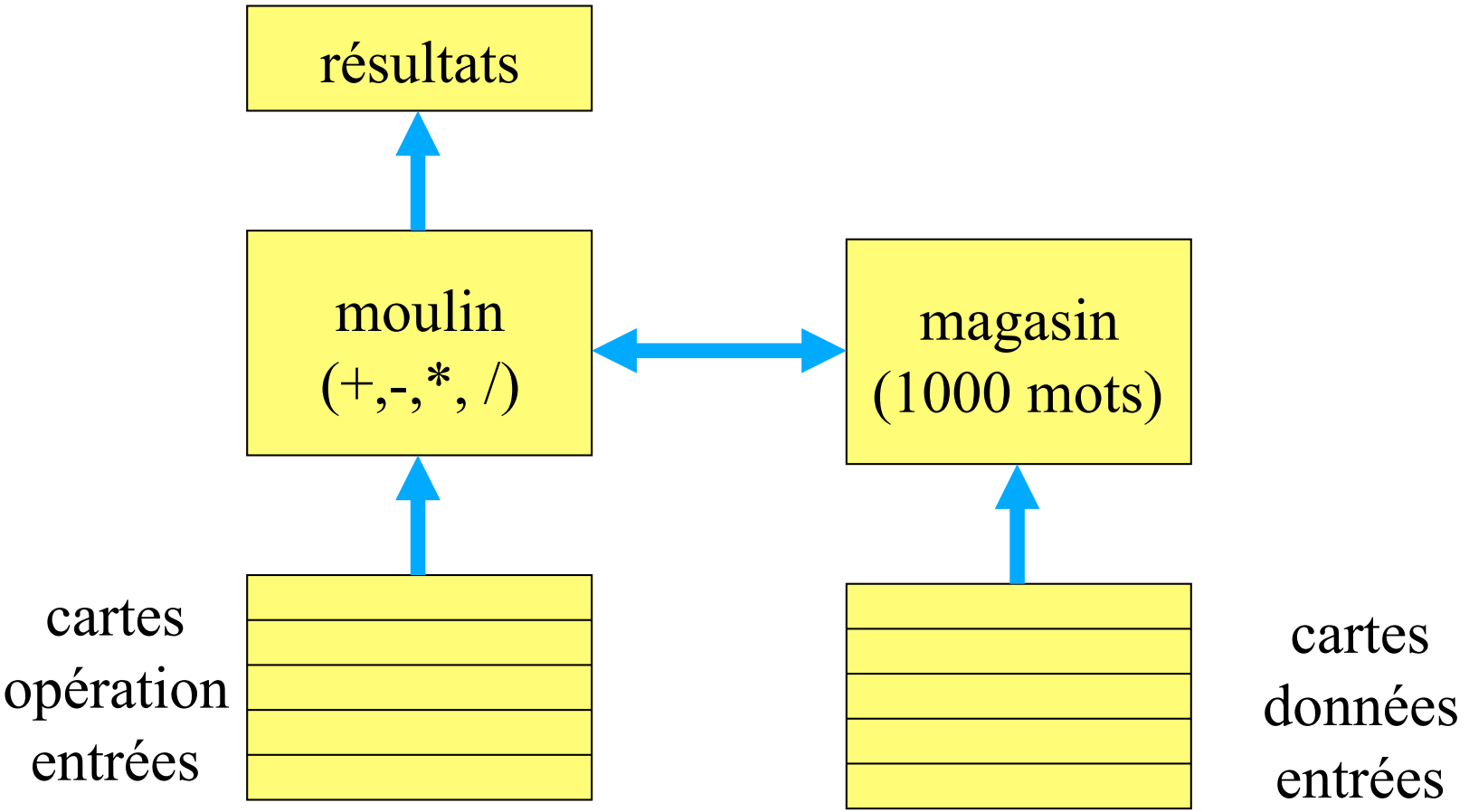
Historique des dispositifs de calcul

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- boulier (antiquité)
- Pascal : machine à additionner
- Liebniz : machine à mutliplier
- Babbage 1822 : differential engine (multiplications)

Historique des dispositifs de calcul

- Babbage 1834 : analytical engine



Historique

- analytical engine (Babbage 1834)

jeu d'instructions			action
code op	sources	destination	
+	W_i W_j	W_k	$W_i + W_j \rightarrow W_k$
-	W_i W_j	W_k	$W_i - W_j \rightarrow W_k$
*	W_i W_j	W_k	$W_i * W_j \rightarrow W_k$
/	W_i W_j	W_k	$W_i / W_j \rightarrow W_k$
adv N	W_i		N cartes $\downarrow$ si $W_i < 0$
ret N			N cartes $\uparrow$

  - construite seulement en 1995 (fonctionne, avec les restrictions technologiques de 1850)
  - le programme n'est pas enregistré en mémoire

Historique

- analytical engine

exemple :

I =10

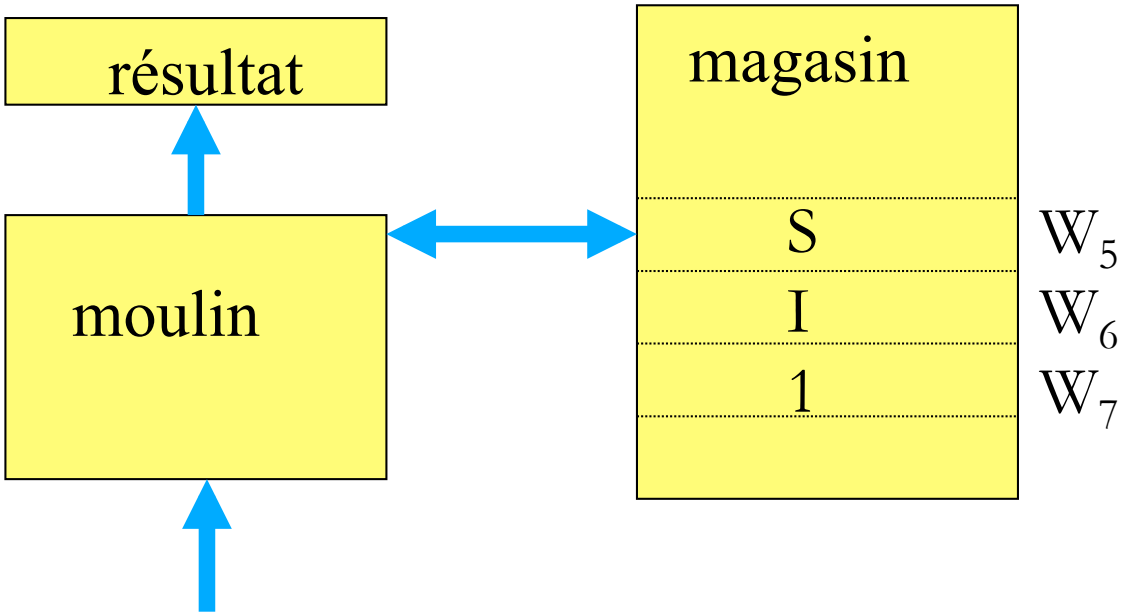
S =0

tant que I ≠ 0 faire

  S = S + 1

  I = I + 1

fait



1	ADV 4	W_6	I < 0 goto 5
2	+	W_5 W_6 W_5	S + 1 → S
3	-	W_6 W_7 W_6	I -1 → I
4	RET 3		goto 1
5	IMP res		

Historique des dispositifs de calcul

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- analogique / digital
  - calcul analogique : les valeurs manipulées sont représentées par des grandeurs physiques (intensité, potentiel)
  - calcul par somme et différence des grandeurs physiques
  - au XXème siècle de nombreux dispositifs de calcul analogiques
  - les premiers ordinateurs comportaient souvent des parties analogiques

Historique des dispositifs de calcul

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- analogique / digital
  - inconvenient des calculs analogique
    - échelle de valeur restreint : un ordre de grandeur (2 au max)
    - calculs peu précis
  - domaines d'application limités
  - calcul digital : les valeur sont représentées par des nombres; calcul arithmétiques
  - XXI siecle : suprématie totale des calculs digitaux

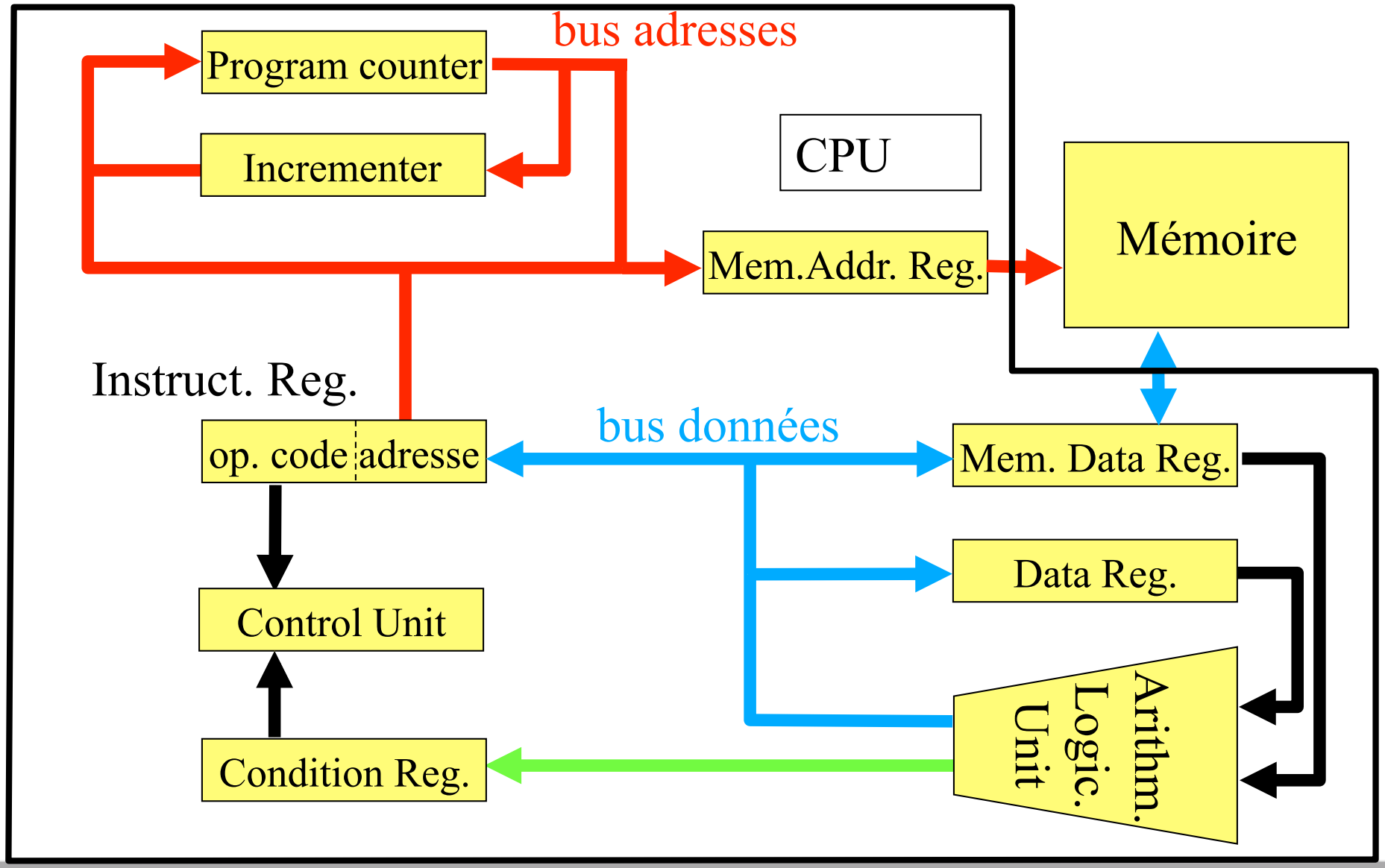
Historique des dispositifs de calcul

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- décimal versus binaire
  - il est beaucoup plus facile de faire un dispositif physique à deux états plutôt qu'à 10 (en dehors des dispositifs mécanique)
  - les calculs sont plus simples sur des valeurs binaires que sur des valeurs décimales

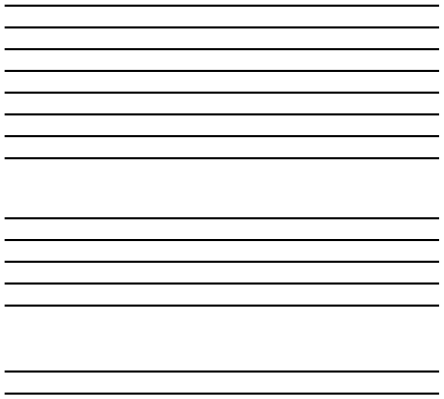
    (cf table d'addition et de multiplication)

Architecture Von Neuman (1943, toujours actuelle)



BUS

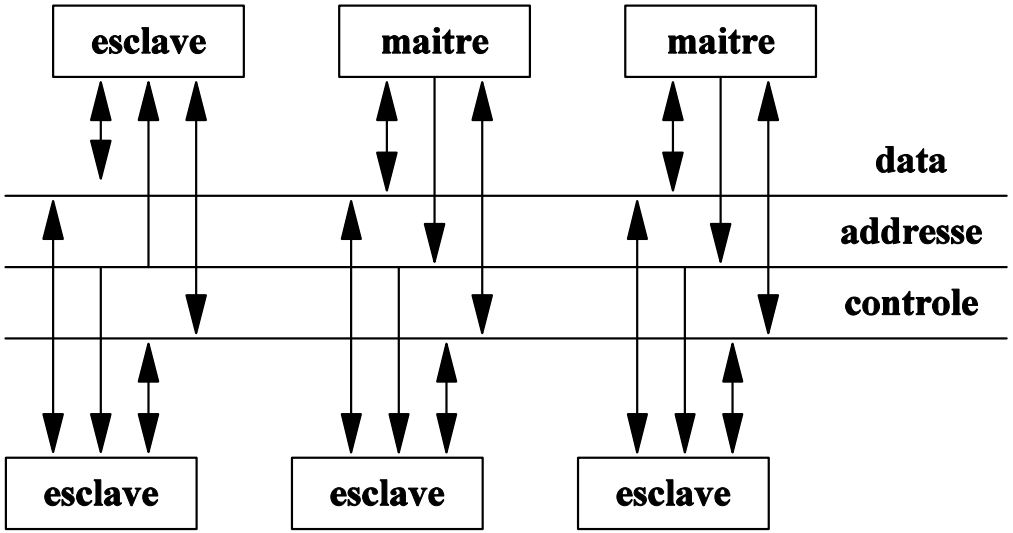
- dispositif pour transférer simultanément plusieurs bits entre deux composants
- le composant destinataire est identifié par une adresse
- les conflits d'accès et les transferts sont gérés à l'aide des lignes de commande (control)



données(data)
8 bits

adresse

commande



arch. Von Neuman

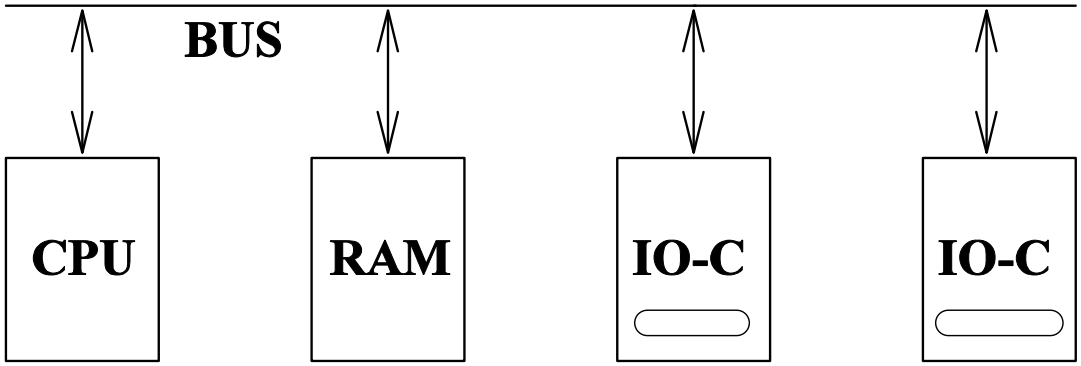
ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- Caractéristique
  - Programme ET données stockés en mémoire
- Conséquences
  - un cycle instruction à besoin de lire
    - une instruction et
    - un ou plusieurs opérandes

  ⇒le bus mémoire est un goulot d'étranglement
- l'unité de commande (contrôl unit):
  - ouvre et ferme les accès aux bus
  - indique à l'ALU quelle opération exécuter

contenu d'un boitier

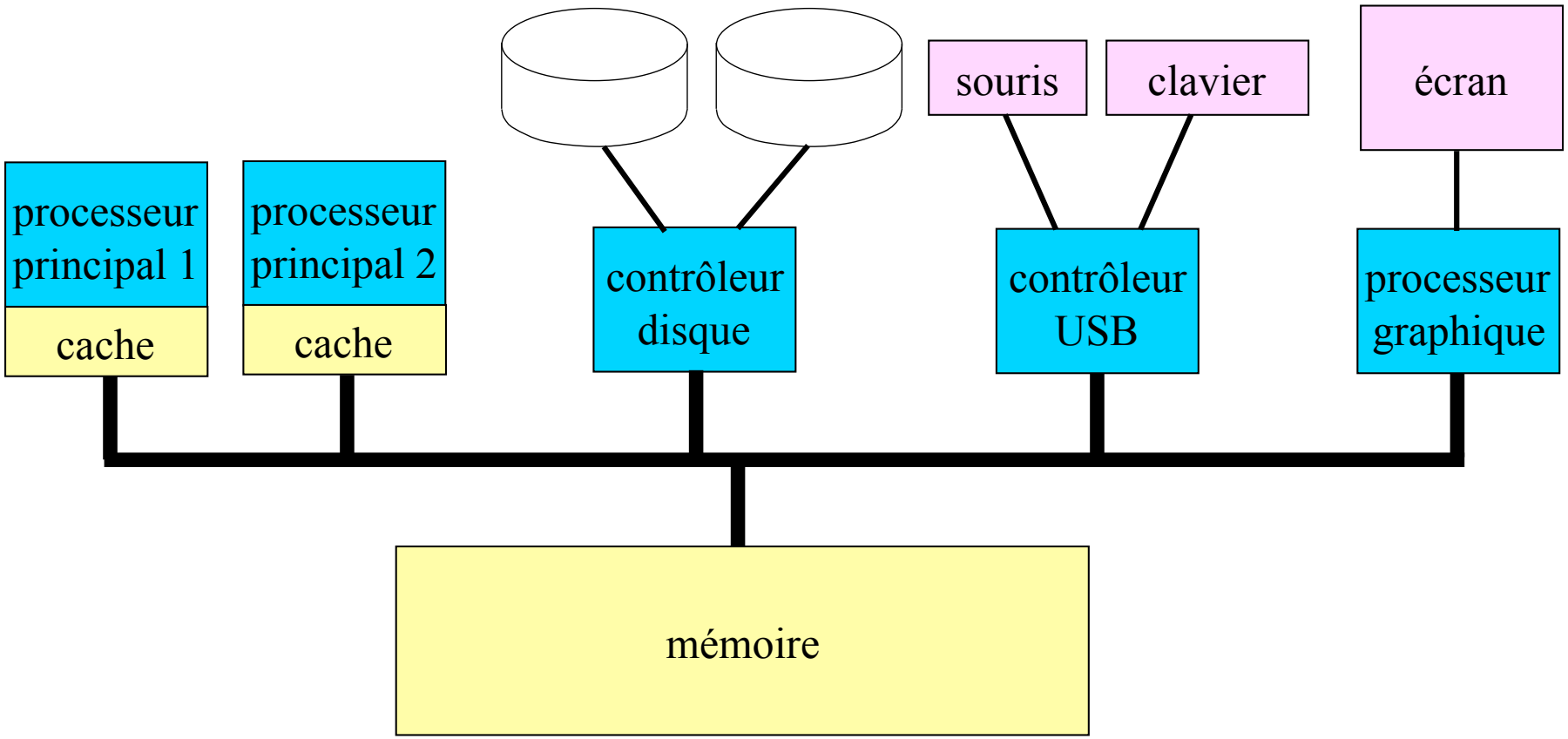
- composants d'un ordinateur



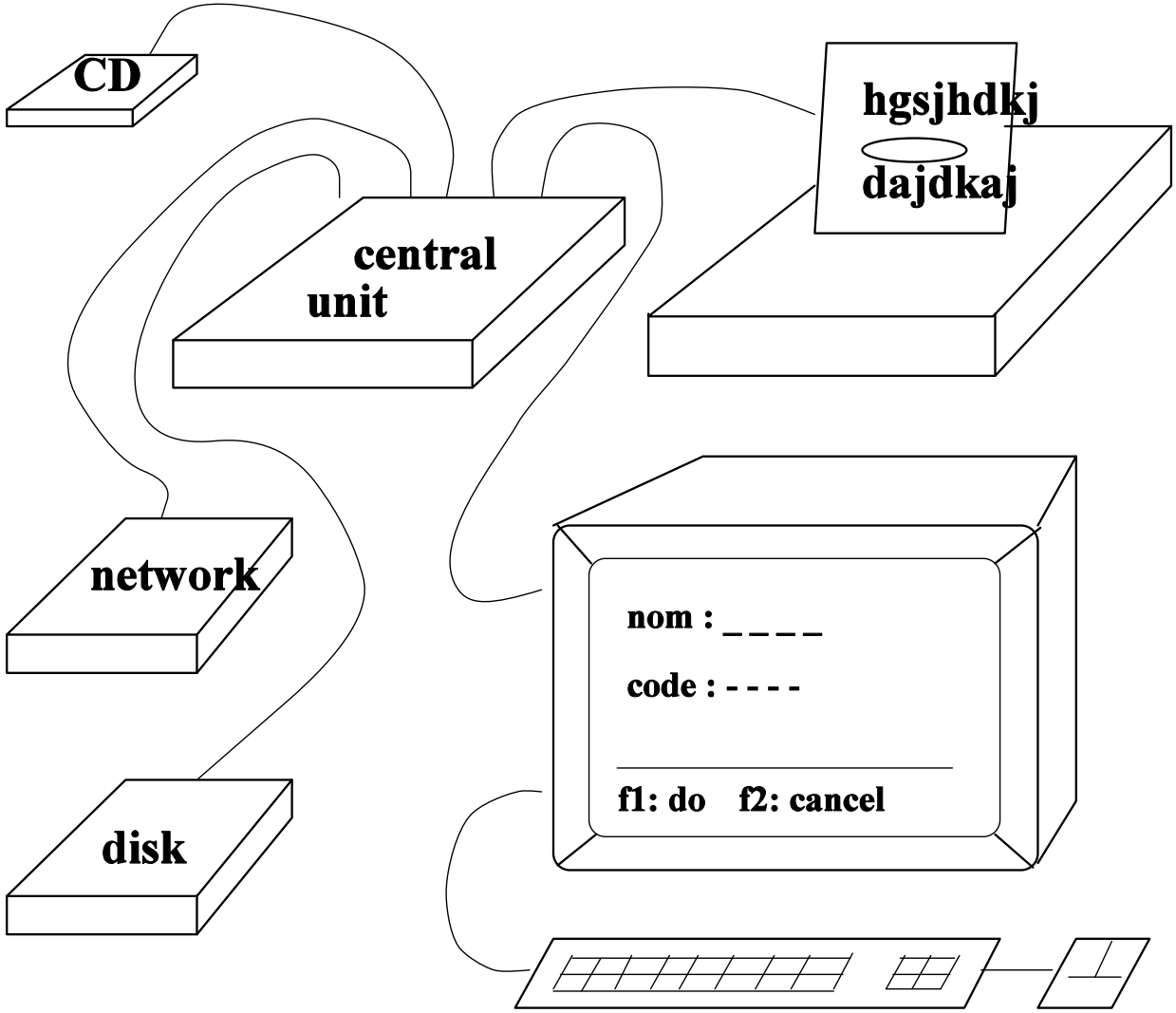
IO-C: I/O controler

Structure

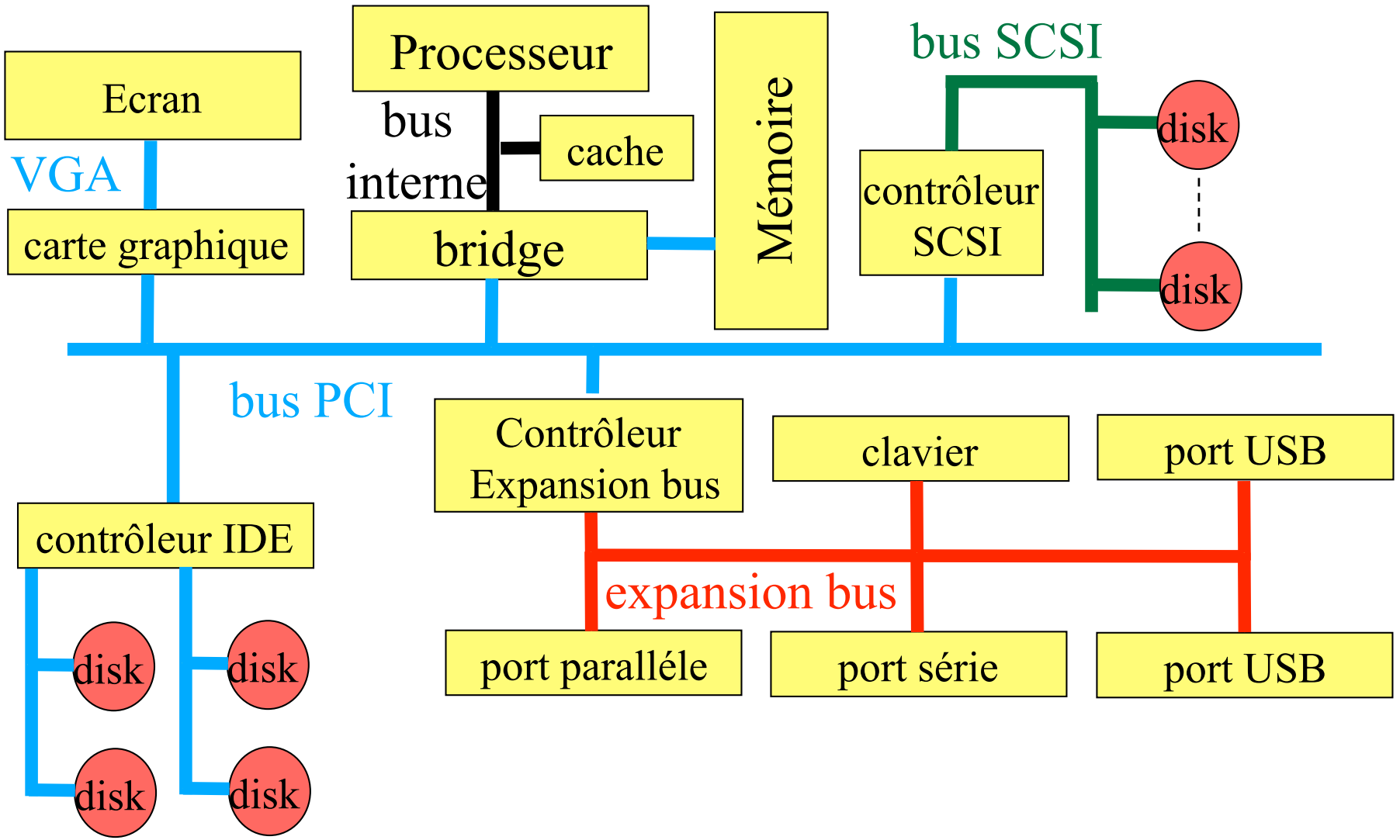
- Structure générique



Organes d'un ordinateur



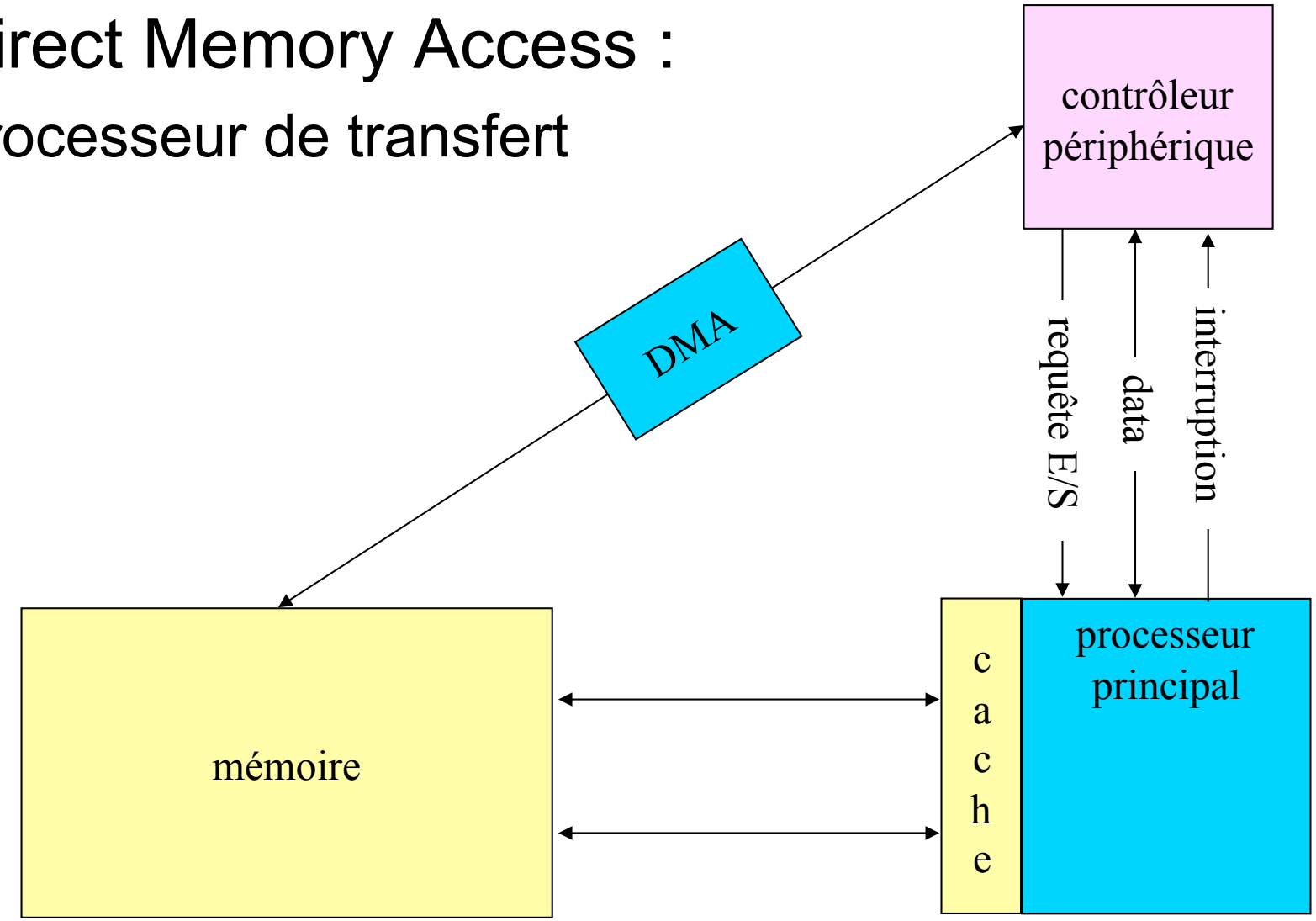
Architecture PC actuel



Architecture d'un PC actuel

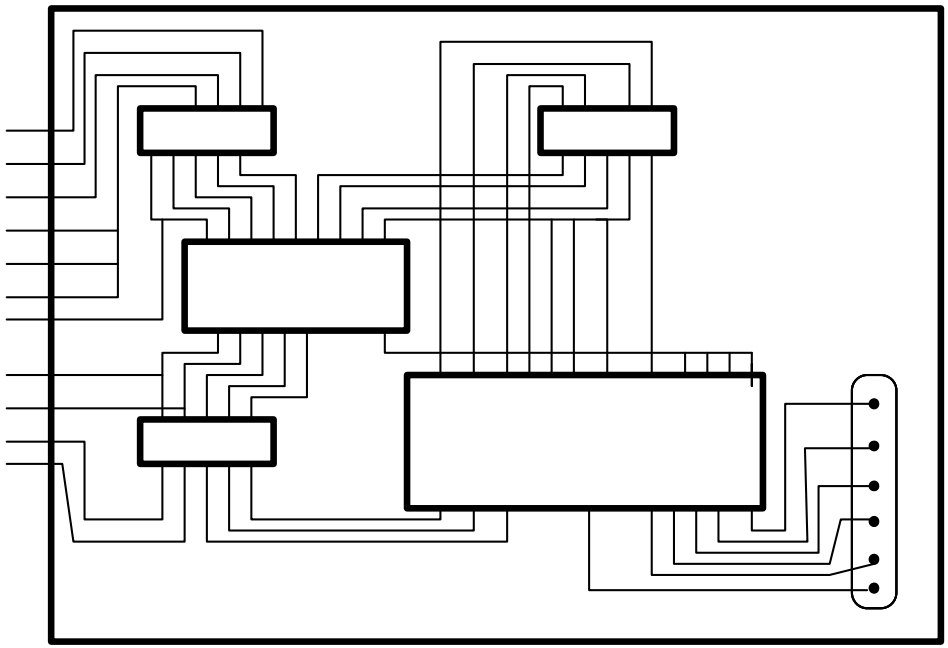
- Direct Memory Access :

  Processeur de transfert



Composants: cartes

- base: porte logique (gate)
  - transistors déposés sur des circuits intégrés (CI)
  - CI fixés sur des cartes
  - deux types NPN et PNP (phosphore : impuretés dans le silicium)



Composants : vue silicium

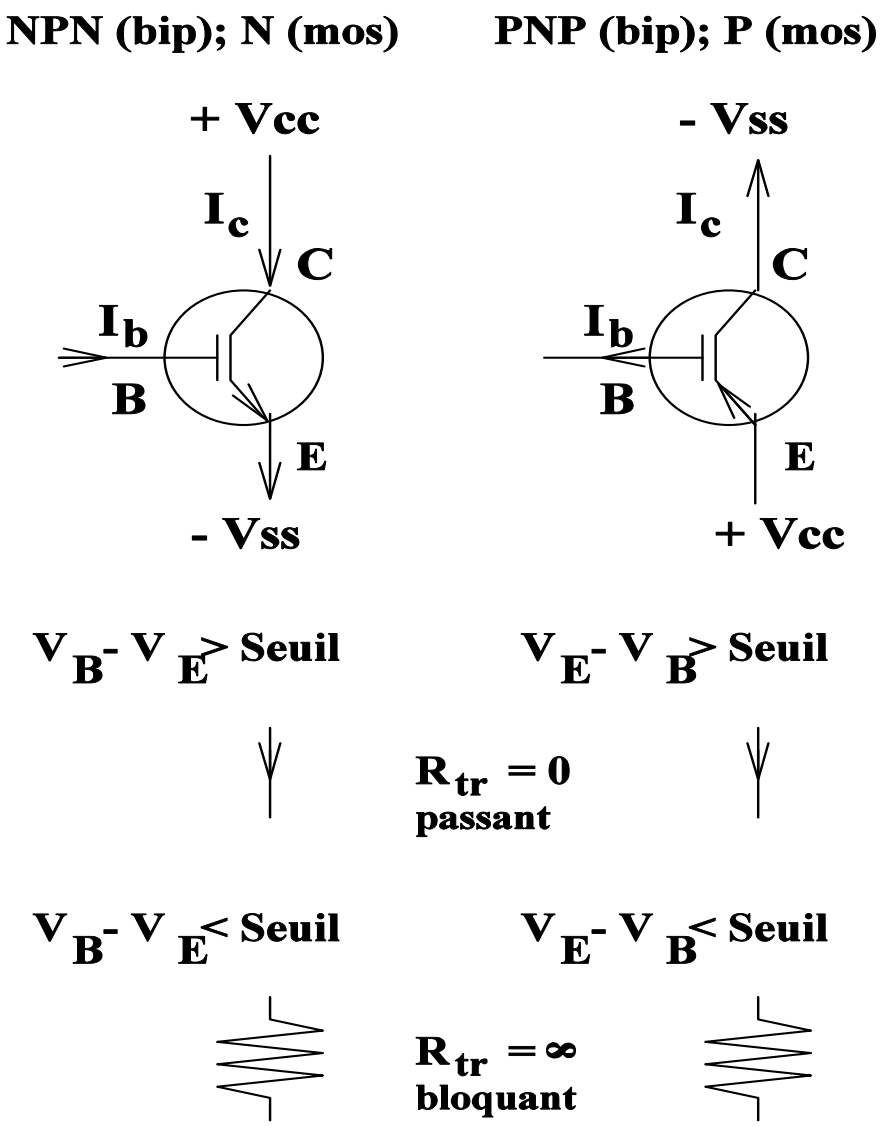
composants : vue fonctionelle

- Transistor :

NPN : si la différence de potentiel V_B - V_E entre la base et l'émetteur est supérieure à un seuil donné alors le transitor est passant sinon il est bloquant

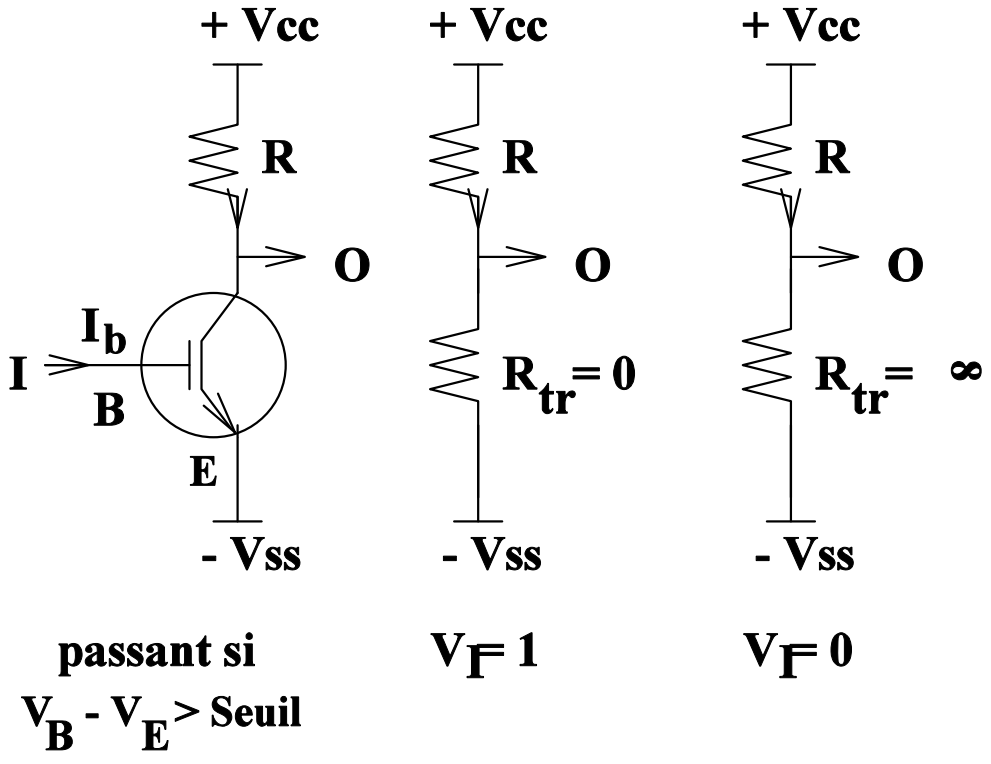
⇒ possibilité de bloquer ou d'autoriser l'information venant du collecteur

PNP : idem dans l'autre sens



Composants: différentes fonctions

- Porte logique Non :



V_I	V_E	R_{tr}	V_O
0	0	$+\infty$	+
+	0	0	0

V_O = non V_I

Composants: différentes fonctions

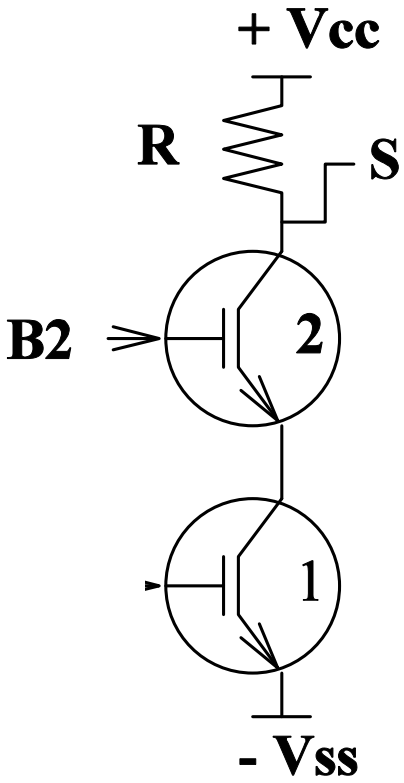
- porte logique NAND

si B2 -Vss > seuil ET B1-Vss> seuil
alors les deux transitors sont non bloquants et Vcc est envoyée sur Vss.

Sinon Vcc est redirigée sur S.

V_{B1}	V_{B2}	R_{tr1}	R_{tr2}	V_S
0	0	+∞	+∞	+
0	+	+∞	0	+
+	0	0	+∞	+
+	+	0	0	0

$S = \text{non}\ (B_1 \wedge B_2)$

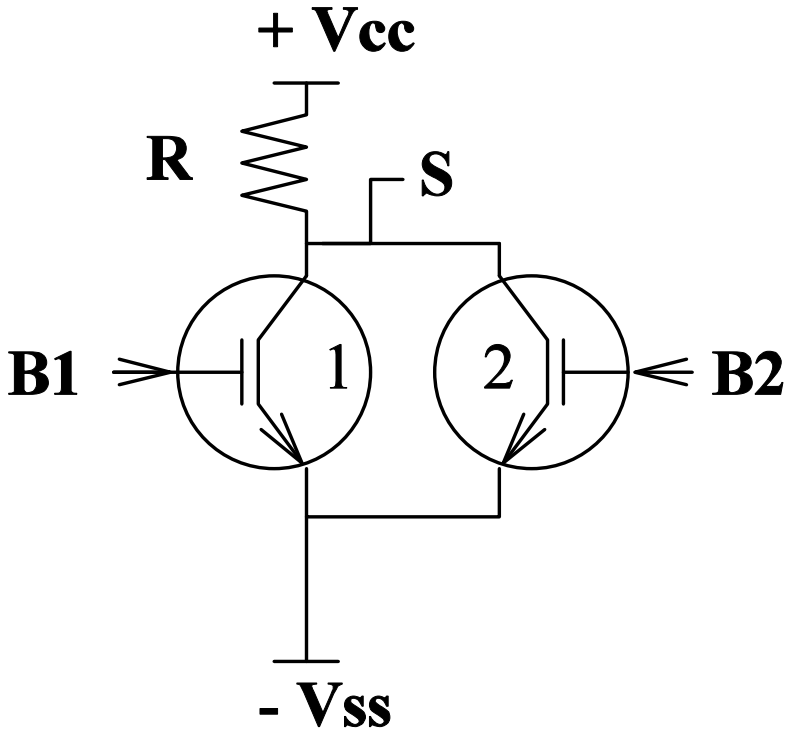


Composants: différentes fonctions

- porte logique NOR

V_{B1}	V_{B2}	R_{tr1}	R_{tr2}	V_S
0	0	$+\infty$	$+\infty$	+
0	+	$+\infty$	0	0
+	0	0	$+\infty$	0
+	+	0	0	0

$S = \text{non}\ (B_1 \vee B_2)$



Composants: différentes fonctions

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- A retenir :
  - tous les opérateurs logiques unaires, binaires, ou n-aires peuvent êtres construits à partir de l'opérateur NOR ou de l'opérateur NAND

Technologie des composants

- “Loi” de Moore : observation empirique : nombre de transistors sur une puce de silicium double tous les deux ans (2e loi de Moore, 1975).
- Observation commune: la « puissance », la « vitesse », double tous les 18 mois.
- Le niveau d’intégration atteint aujourd’hui ses limites. L’augmentation de la puissance de calcul passe par le parallélisme (notamment multi-cœurs).

Calculabilité

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- Définition: ce qui peut être déterminé avec un algorithme.
- Question: quelles sont les fonctions calculables ? (Ceci donne les limites de ce qu'un ordinateur peut faire).
- Plusieurs réponses vers 1930-40: lambda-calcul, fonctions récursives. La machine de Turing: plus simple dispositif permettant d'exprimer un algorithme.

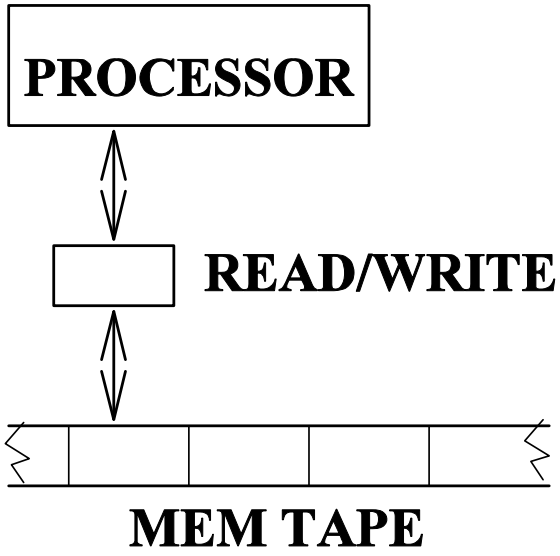
Notion de fonction calculable

- Machine de Turing (Alan)
  - dispositif de calcul abstrait M
    - X : alphabet X= {0,1,◊ } (◊ : symbole vide)
    - Q : états Q = q_0,q_1, ...
    - F : régles d'évolution Q,X→ X,G/D,Q
  - le processeur lit un caractère

sur la bande puis réagit suivant

la régle d'évolution qui s'applique

à son état actuel et au caractère lu



Notion de fonction calculable

- Machine de Turing (1936)
  - dispositif de calcul abstrait M composé de
    - bande de calcul (contient les données)
      (infinité de cellules contenant des symboles)
    - tête de lecture/écriture (déplacement gauche ou droite)
    - Fonction de contrôle F (spécifie les règles d'évolution, c.à.d un programme)
      - X,X' : alphabets par exemple : X= {0,1,◊ } ◊ : symbole vide
      - Q : états Q = q_0,q_1, ...
      - régles d'évolution F : Q x X→ X' x {G/D} x Q

Notion de fonction calculable

- Machine de Turing (1936)
  - dispositif de calcul abstrait M composé de
    - bande de calcul (contient les données)
      (infinité de cellules contenant des symboles)
    - tête de lecture/écriture (déplacement gauche ou droite)
    - Fonction de contrôle F (spécifie les règles d'évolution, c.à.d un programme)
      - X : alphabet $X = \{0,1,\Diamond\}$ $\Diamond$: symbole vide
      - Q : états $Q = q_0, q_1, \ldots$
      - régles d'évolution $F : Q \times \{0,1\} \rightarrow \{0,1\} \times \{G/D\} \times Q$

Notion de fonction calculable

- Machine de Turing

exemple

$X = \{0,1,\Diamond\}$

$Q = \{q_0,q_1\}$

$F : \{\{q_0,1 \rightarrow 1,D, q_1\}, \{q_1,1 \rightarrow 0,D, q_0\}\}$

état initial de la bande :

$\Diamond$	$\Diamond$	$\Diamond$	1	1	1	$\Diamond$	$\Diamond$

Pas à Pas

position de la tête : ↑						règle utilisée
q_0	$\Diamond$	1 ↑	1	1	$\Diamond$	$q_0 , 1 \rightarrow 1, D, q_1$
q_1	$\Diamond$	1	1 ↑	1	$\Diamond$	$q_1 , 1 \rightarrow 0, D, q_0$
q_0	$\Diamond$	1	0	1 ↑	$\Diamond$	$q_0 , 1 \rightarrow 1, D, q_1$
q_1	$\Diamond$	1	0	1	$\Diamond$ ↑	$q_1 , \Diamond \rightarrow$ arrêt

état résultat

$\Diamond$	$\Diamond$	$\Diamond$	1	0	1	$\Diamond$	$\Diamond$

Que peux-t-on calculer avec une machine de Turing ?

Notion de fonction calculable

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- fonction Turing-calculable
  - une fonction partielle F est T-calculable s'il existe M telle si la donnée d'entrée codée par $x_1,x_2,...x_k$ appartient au domaine de définition de F alors M s'arrête en un nombre fini de pas et produit le résultat $F(x_1,x_2,...x_k)$
  - si la donnée d'entrée n'appartient pas au domaine de définition alors M ne s'arrête jamais
  - Remarques
    - Il existe des fonctions non T-calculables
    - les fonctions Turing-calculables sont les fonctions récursives de Kurt Gödel

Notion de fonction calculable

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- Machine de Turing universelle
  - Une machine de Turing universelle est une machine de Turing qui sur des données d'entrées décrivant d'une part une machine de Turing M donnée et d'autre par une donnée d'entrée x pour cette machine simule l'action de M sur x (c.à.d calcule M(x))
  - U(M,x) = M(x)
  - on peut lui donner son programme sur sa bande d'entrée (pas obligé de construire une machine par algorithme).

Notion de fonction calculable

- Machine de Turing universelle
  - problème de l'arrêt :
    - Soit une machine M1 qui calcule une valeur F(x)
    - Après plusieur millions d'étapes, elle tourne toujours
    - faut-il l'arrêter ou continuer encore un peu pour obtenir le résultat ?

    - Existe-t-il une machine M2 qui ayant comme donnée M1 et x pourrait se terminer par vrai si M1 s'arrêtera et se terminer par faux sinon

Notion de fonction calculable

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- Décidabilité
  - Un problème est décidable si l'ensemble des solutions est T- calculable.
  - L'arrêt de la machine de Turing est indécidable

- Thèse de Church (1950)

  toute fonction mécaniquement calculable est T-calculable
  - tout les processeurs sont fonctionnellement équivalents
  - savoir si un programme s'arrête est indécidable

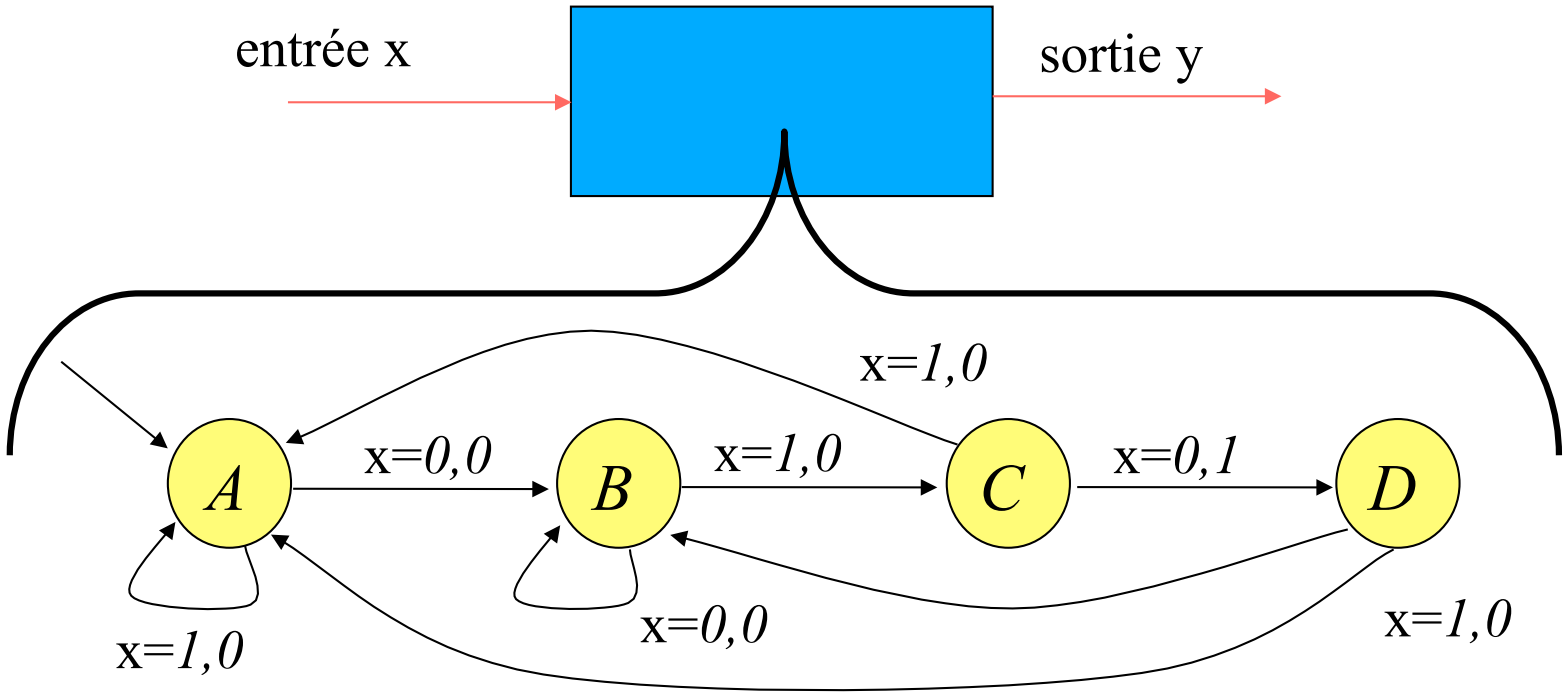
Comment spécifier un algorithme ?

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- En donnant l'ensemble de règles d'évolution d'une machine de Turing
- En donnant une machine à état (automate)
- Plusieurs types d'automates
- Automate fini
  - Un automate fini est un quintuplet $A=(q_0,Q,F,X,f)$ où
    - Q est un ensemble fini d'états et q_0 est l'état initial
    - $F \subset Q$ est l'ensemble des états finaux
    - X (alphabet) est un ensemble fini de symboles
    - $f : Q \times X \rightarrow Q$ est la fonction de transition

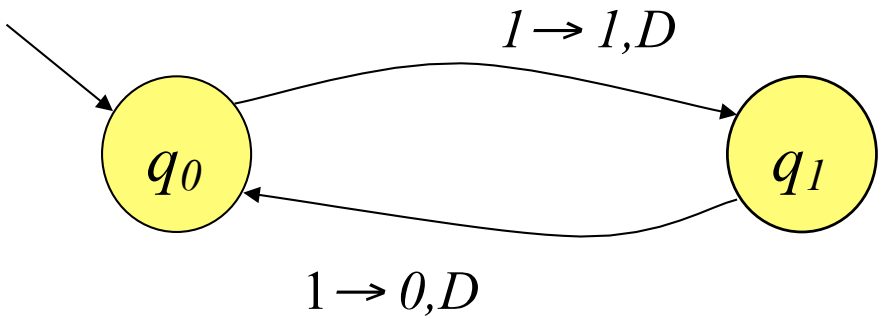
Spécification d'algorithme

- automate fini reconnaissant la séquence d'entrée 010 (envoie 1 sur y lorsqu'il reconnait 010)



Spécification d'algorithme

- automate de la machine de Turing qui remplace 111 par 101



Rappels d'algèbre de Boole

- (p => q) ⇔ !p or q
- !(p or q) ⇔ !p and !q Loi De Morgan
- !(p and q) ⇔ !p or !q Loi De Morgan

Rappels algèbre booléenne

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- Tableaux de Karnaugh: méthode de simplification expressions booléennes.
- méthode « graphique », qui utilise notre capacité à reconnaître des motifs géométriques.
- difficile à pratiquer au-delà de 6 variables.

Rappels algèbre booléenne

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- Principe:
  - répartir les variables en ligne et colonne sur un tableau 2^n cases pour n variables
  - décliner les valeurs des variables selon code binaire réfléchi (Gray) (ex: 00,01,11,10).
  - faire des regroupements de 1 de forme carrée ou rectangulaire de 2^n cases (toroïdal permis, éléments peuvent être pris plusieurs fois)
  - on s'arrête quand on a pris tous les 1
  - on ne garde que les variables qui ne prennent pas une valeur constante.