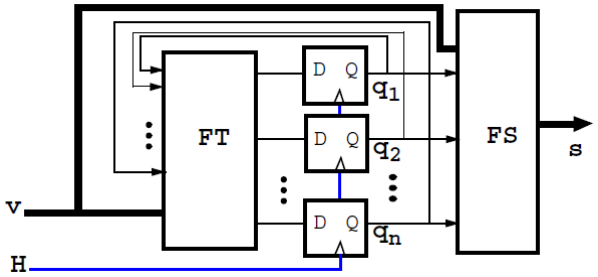
3.6. Exemple de synthèse avec **codage binaire**

Synthèse d'un compteur MODULO4 avec un codage binaire

Cet exercice a pour but d'illustrer la synthèse d'un circuit séquentiel simple à l'aide de bascules D, avec un des codages possibles des états : le codage binaire.
Pour rappel, un système séquentiel peut être modélisé comme sur le schéma suivant, ou FT est la fonction de transition, et Fs la fonction de sortie.



On veut réaliser un compteur modulo 4, qui s'incrémente lorsqu'une commande Inc est à1. Si la commande Inc est à 0, le compteur ne change pas d'état.

1) Définition des entrées
Combien ce système a t-il d'entrée(s) ? Laquelle ou lesquelles ?

1 entrée INC

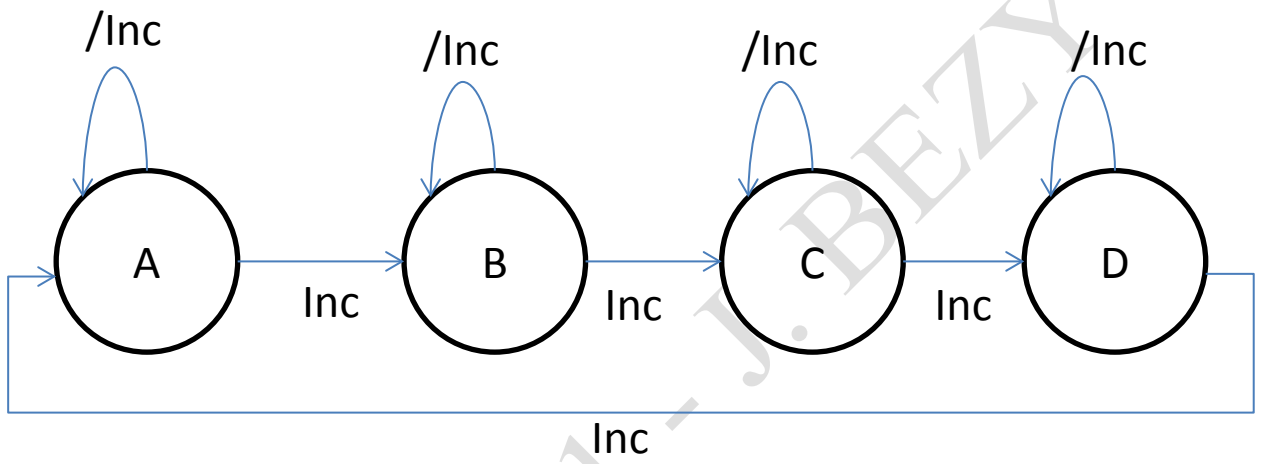
2) Définition des états
Combien ce système a t-il d'états ?
Vous les noterez A, B,..

4 états : A, B, C, D

3) Détermination de la fonction de transition (*encore appelée « next-state» car on l'applique à l'entrée des bascules D pour qu'elle fasse passer le système dans l'état suivant*)
Pour obtenir la fonction de transition (FT) vous commencerez par :

a) Construire le diagramme des états
b) Coder les états. Ici on utilisera le codage binaire (états codés sur n bits q_0, q_1, .. q_{n-1}).

a)



b) On utilise le code binaire. Encodage binaire arbitraire : 2 bits nécessaires pour coder 4 codes. Par exemple :

Etat	Code sur 2 bits q1q0
A	00
B	01
C	11
D	10

c) Donner la fonction de transition FT à appliquer sur l'entrée des bascules D, en fonction des n bits du code ($q_{n-1}..q_0$) et des entrées. Il s'agit de déterminer l'état suivant (que l'on peut nommer $q_{n-1}+$, $q_{n-2}+$… q_0+)

FT : q1+q0+

	Inc q1q0	0	1
Etat A	00	00	01
Etat B	01	01	11
Etat C	11	11	10
Etat D	10	10	00

q1+=q1./Inc+q0.Inc
q0+=q0./Inc+/q1.Inc

4) Détermination de la fonction de sortie (FS)

Il y a **deux sorties** à ce système : **S1 et S0, qui donnent la valeur (l'état) du compteur en binaire**.

Après avoir rempli le tableau suivant, donner l'expression des fonctions de sorties en fonction de l'état (représenté par les bits $q_{n-1}..q_0$), et des entrées.

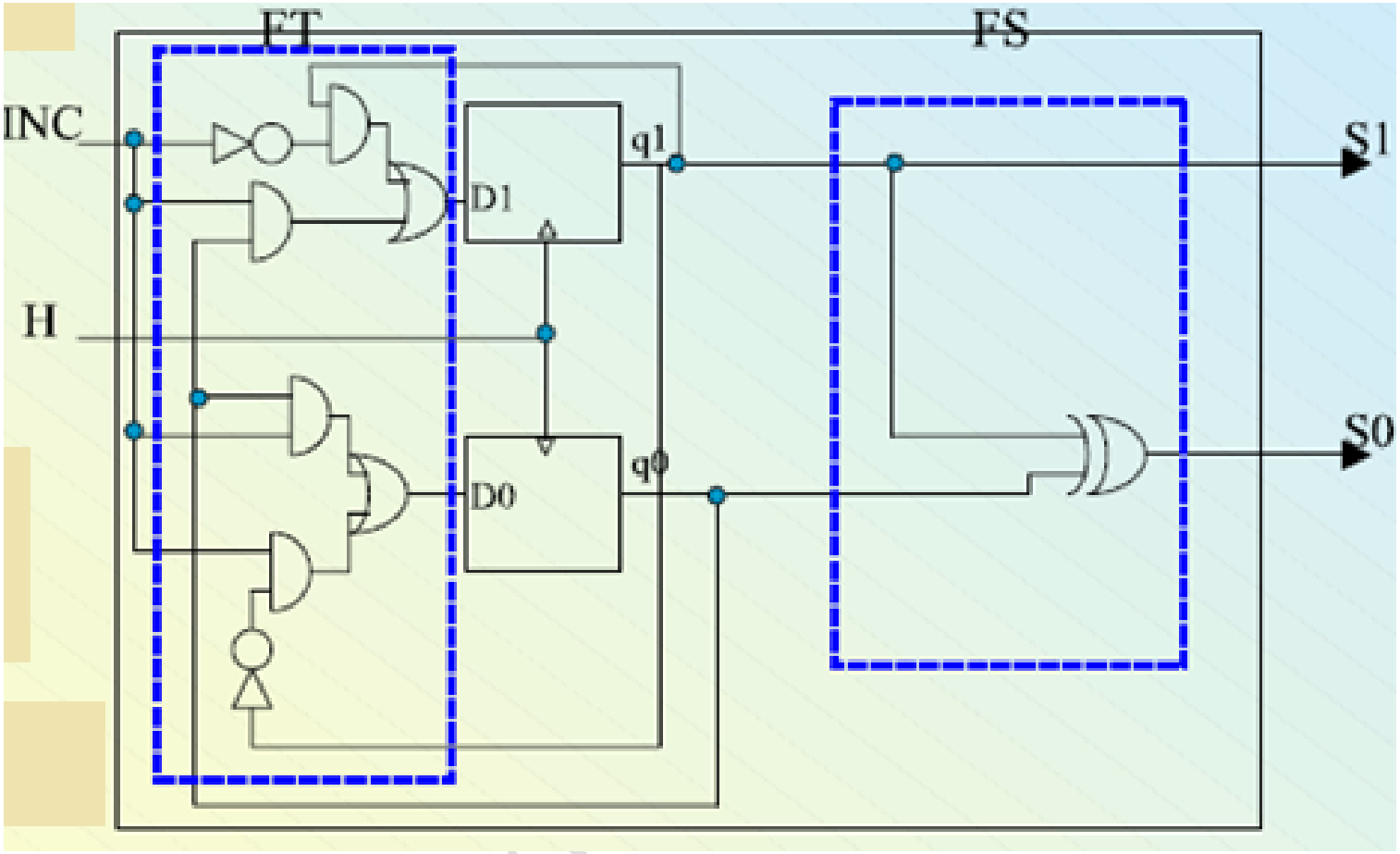
Etat du système	$q_{n-1}……q_0$	S_1 S_0	Valeur du compteur en décimal
A B C D			0 1 2 3

Etat du système	q_1q_0	S_1 S_0	Valeur du compteur en décimal
A B C D	00 01 11 10	00 01 10 11	0 1 2 3

On trouve :

S1=q1
S0=q1⊕q0

5) Réalisation du circuit
Réaliser le circuit avec des bascules D.
Vous ferez bien apparaître la fonction de Transition FT et la fonction de Sortie FS, comme sur le schéma ci-dessus.



On rappelle qu'on a trouvé :

q1+=q1./Inc+q0.Inc

q0+=q0./Inc+/q1.Inc

S1=q1

S0=q1⊕q0

3.7. Exemple de synthèse avec **codage un bit par état (machine à jeton)**

Exercice 2 : Synthèse d'un compteur MODULO3 avec un codage Machine à jeton

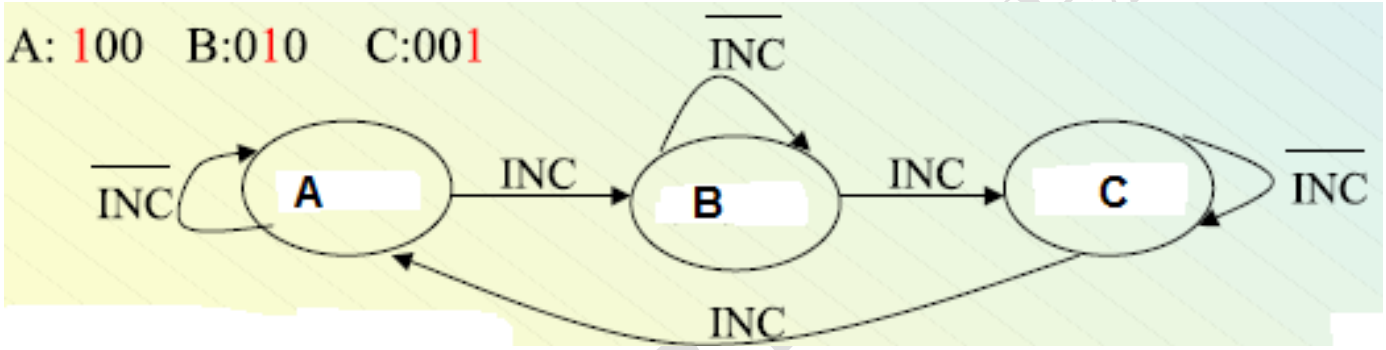
Cette fois-ci, il s'agit d'illustrer un autre type de codage, **menant à une matérialisation différente : le codage Machine à jetons**. Pour rappel, dans ce codage, le nombre de bits est égal au nombre d'états. Le compteur fonctionne encore avec une commande Inc.

1) Définition des entrées
Combien ce système a t-il d'entrée(s) ? Laquelle ou lesquelles ?
Une entrée, Inc.

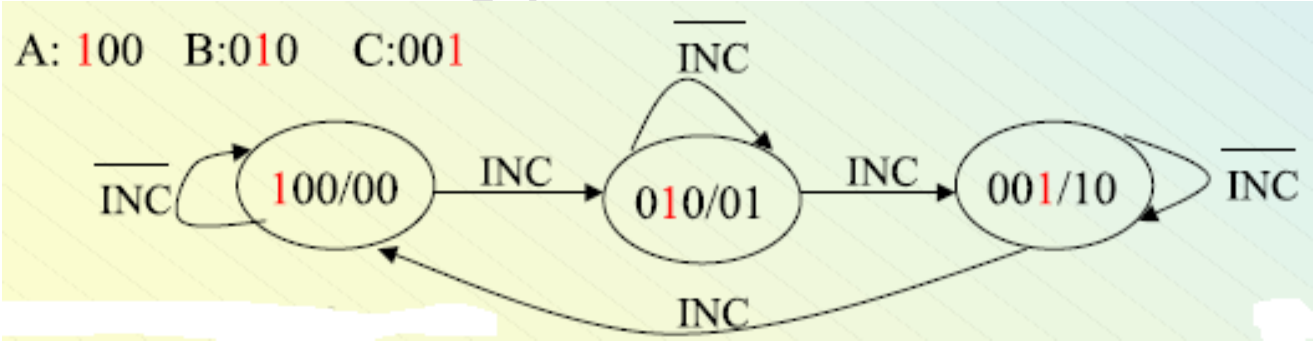
2) Définition des états
Combien ce système a t-il d'états ?
Vous les noterez A, B,..
3 états (car modulo 3, donc compte 0,1,2,0,1,2…)
A, B, C.

3) Détermination de la fonction de transition

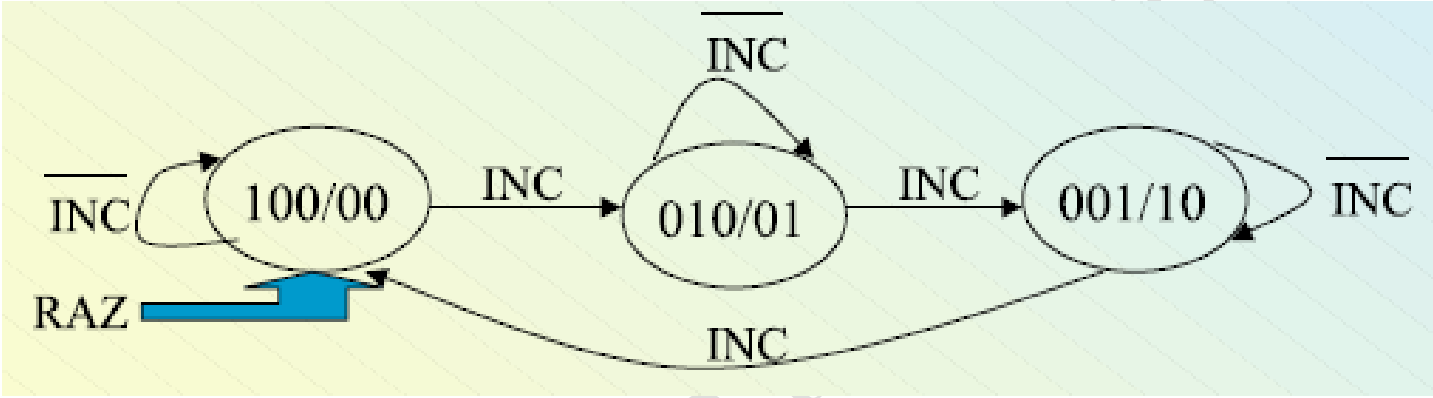
a) Construire le diagramme des états



b) Coder les états. Ici on utilisera le codage Machine à jeton.
3 états alors 3 bits : q3, q2, q1



c) Attention, cette fois il est prévu une commande RAZ qui met le compteur dans l'état 0. Refaites le diagramme des états en tenant compte de cette commande.

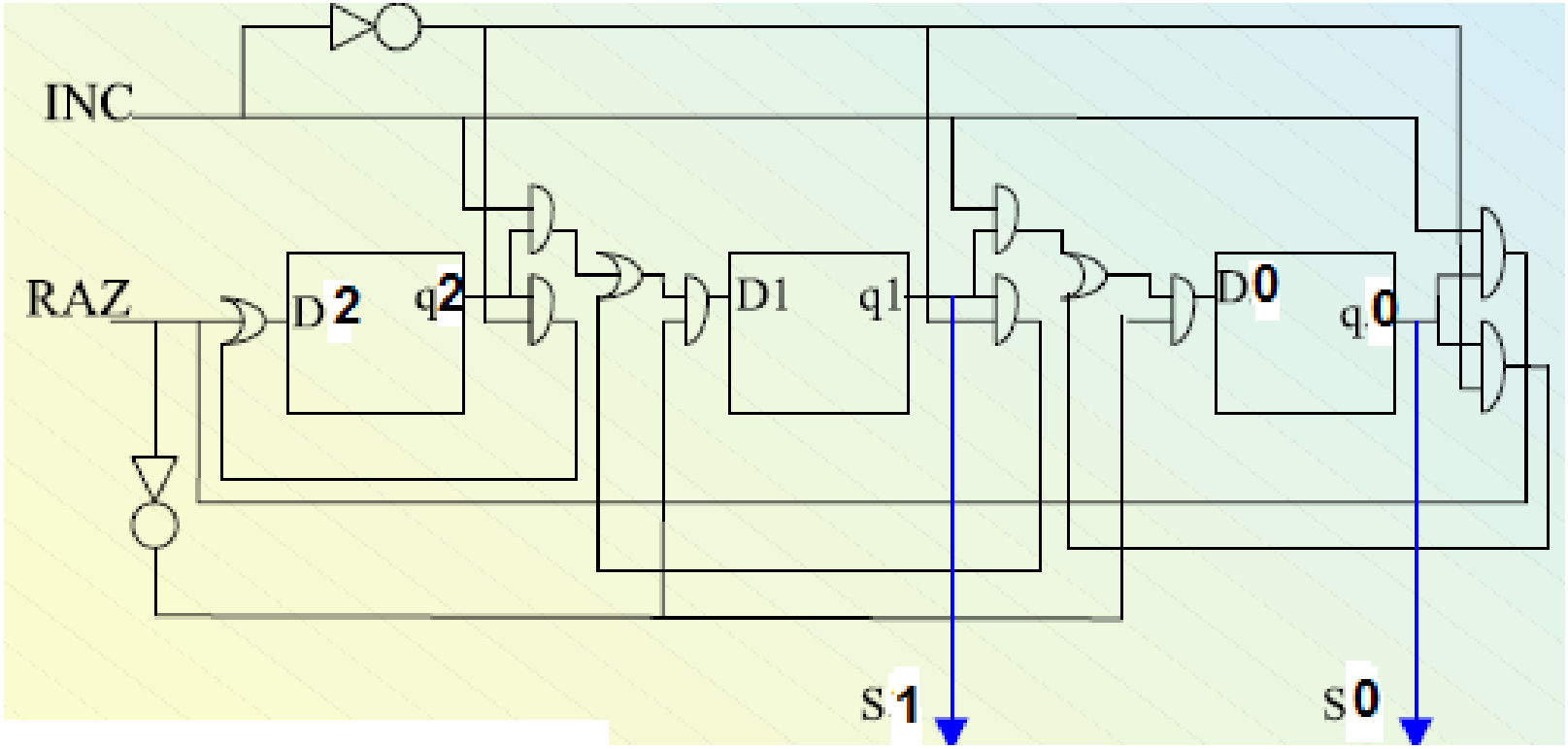


4) Détermination de la fonction de sortie (FS).
Il y a deux sorties à ce système : S2 et S1, qui donnent la valeur (l'état) du compteur en binaire. Après avoir rempli le tableau suivant, donner l'expression des fonctions de sorties en fonction de l'état (représenté par les bits $q_1..q_n$), et des entrées.

Etat du système	q2 q1 q0	S_1 S_0	Valeur du compteur en décimal
A	100	00	0
B	010	01	1
C	001	10	2

S1=1 pour l'état C donc S1=q0
S0=1 pour l'état B donc S0=q1

5) Réalisation du circuit
Réaliser le circuit avec des bascules D.



4. Mémoires

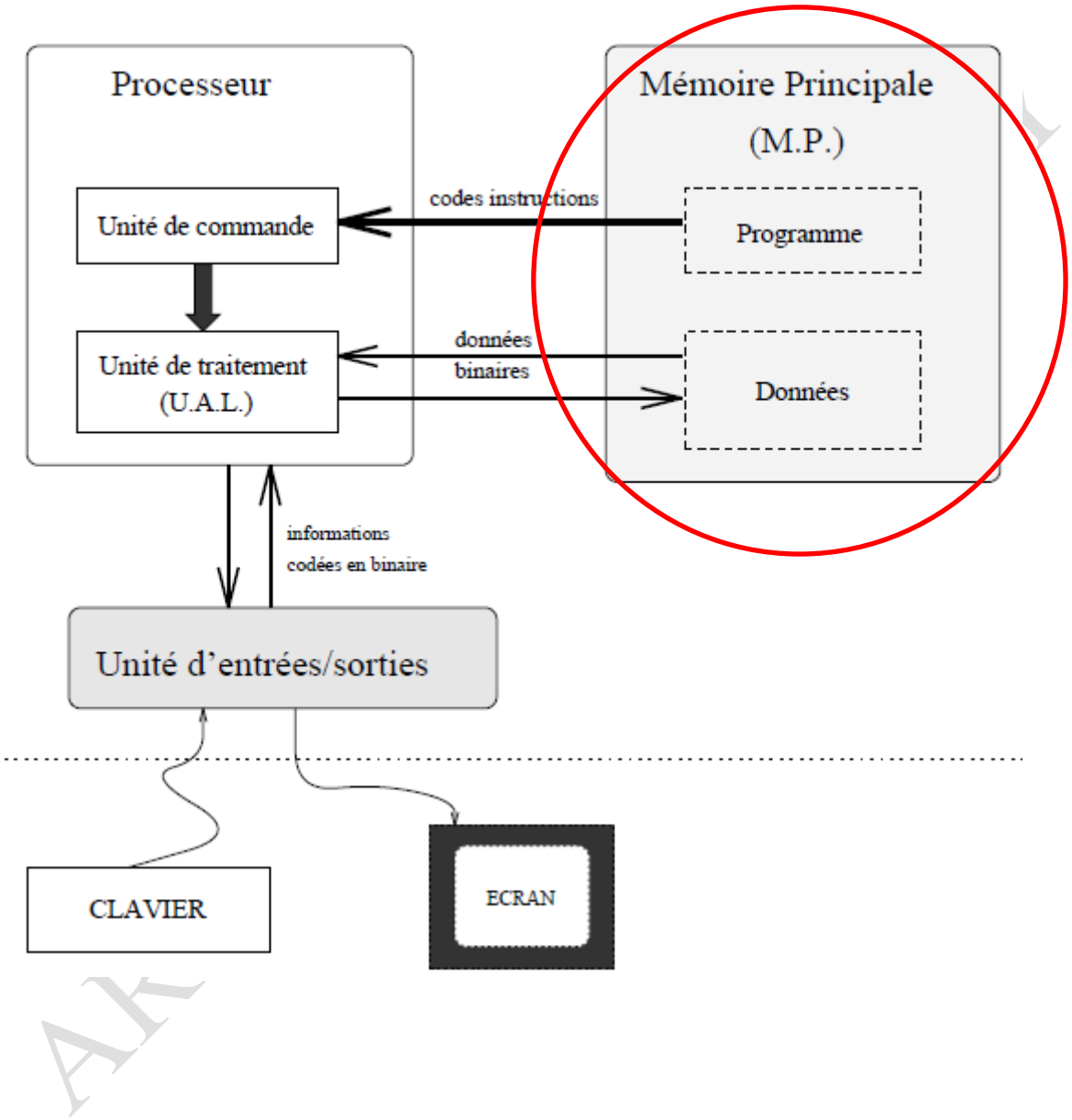
- Mémorisation données : avantage systèmes numériques / systèmes analogiques

- **Mémoire** interne de l'ordinateur contient instructions permettant le fonctionnement de l'ordinateur dans toutes les circonstances (intervention humaine minimum)

- Deux principaux composants d'un ordinateur :
  - ✓ **Mémoire principale** (interne) : stocke de l'information (programmes et données)
  - ✓ **Processeur** : exécute pas à pas les instructions composant les programmes
  - ✓ Fonctionnement processeur : pour chaque instruction, lecture en **mémoire** de l'instruction, réalisation du traitement correspondant, passage à l'instruction suivante
  - ✓ Processeur constitué de 2 parties :

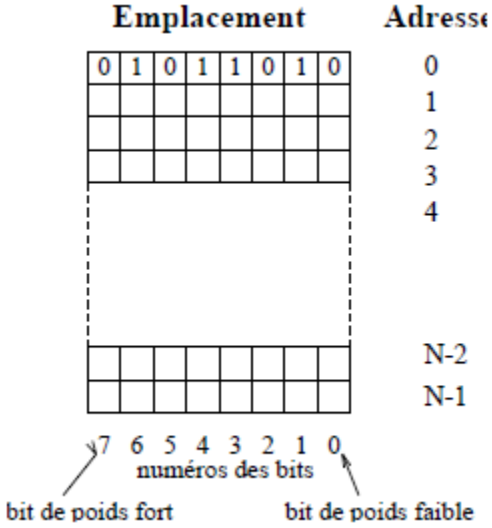
- unité de commande : lecture en **mémoire** et décodage des instructions
- unité de traitement (encore appelée UAL) : exécution des instructions qui manipulent les données

- La mémoire : un des deux principaux composants de l'ordinateur, avec le processeur :


Processeur
Unité de commande
Unité de traitement
(U.A.L.)
Mémoire Principale
(M.P.)
Programme
Données
codes instructions
données
binaires
informations
codées en binaire
Unité d'entrées/sorties
CLAVIER
ECRAN

- La mémoire principale :

  Divisée en emplacements de taille fixe (ex : un octet) pour stocker des données et des instructions



Structure de la mémoire principale :

Mémoire de taille N = N emplacements

Chaque emplacement est numéroté (adresse) : 0 à N-1

Adresse écrite en hexadécimal

Capacité = nombre d'emplacements exprimée en Giga, Tera octets

Ex :

1 G (Giga)= 2^{30} = 1 073 741 824 octets

- Les 2 grands types de mémoires :
  - ✓ Mémoires vives (utilisables pour écrire ou lire des informations – constitue la **plus grande partie de la mémoire principale de l'ordinateur**)
  - ✓ Mémoires mortes (accessibles uniquement en lecture, écriture faite une seule fois à la conception – ROM, ou à l'initialisation en usine – PROM, ou sur site - EPROM)

- Classement des mémoires :

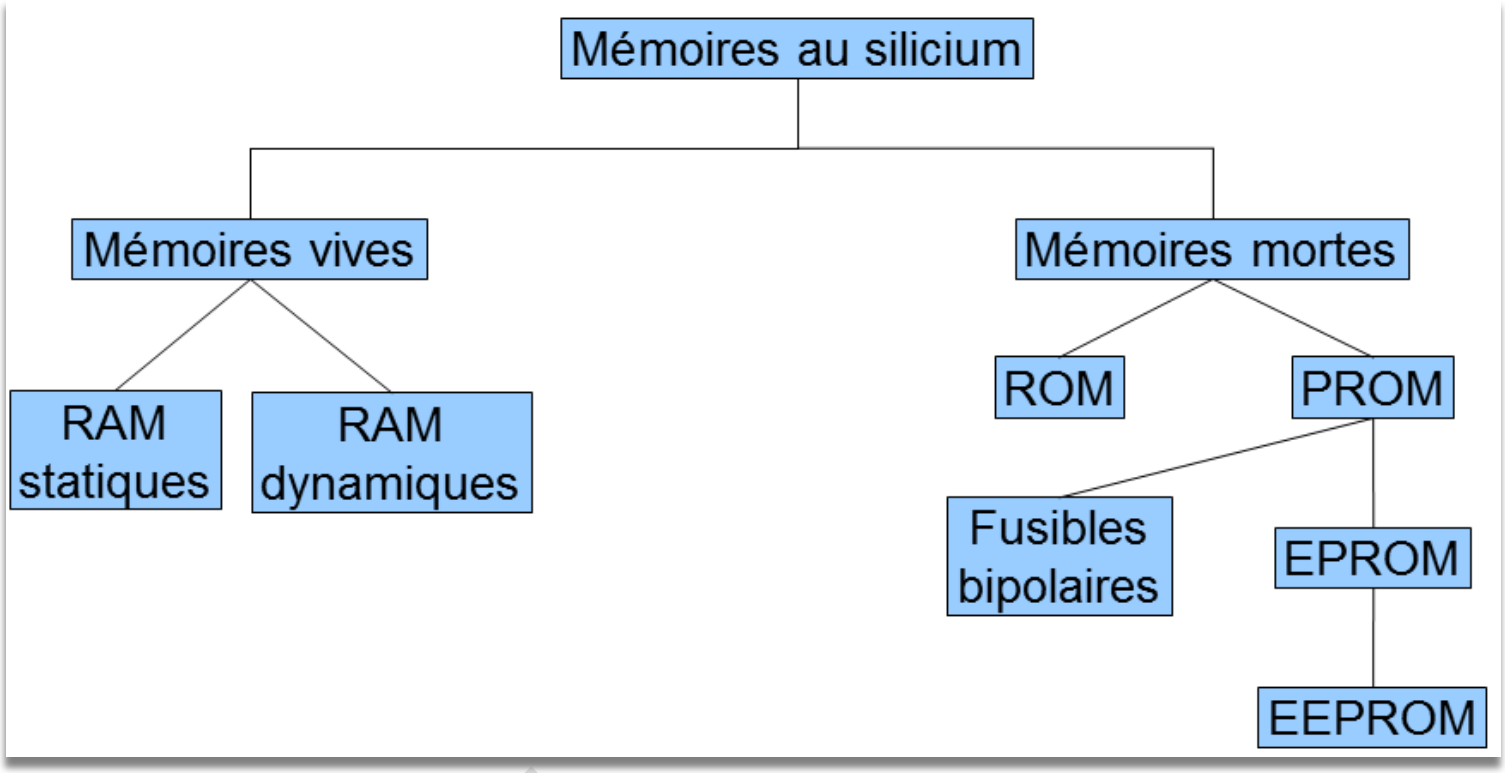


Figure extraite du cours de Laurent Jeanpierre – IUT Caen
RAM : Random Access Memory / ROM : Read Only Memory / PROM : Programmable Read Only Memory / EPROM : Electrically Programmable Read Only Memory / EEPROM : Electrically Erasable Programmable Read Only Memory

- Différentes technologies : microfusible pour PROM, **bascules** pour RAM statique, transistor+condensateur pour RAM dynamique…
- Propriétés variables : rapidité, intégrations, consommation