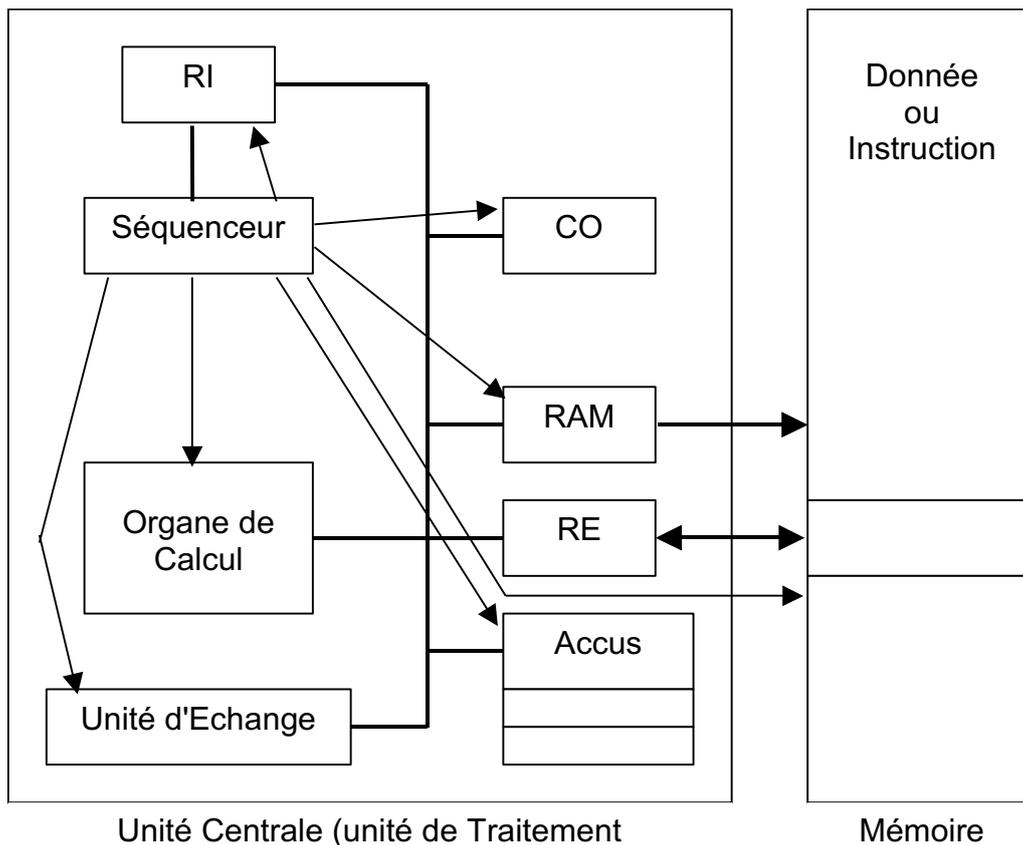


# TD 3&4

## Architecture

### A. RAPPELS DE COURS

#### 1. Unité Centrale



**RI** contient l'instruction en cours d'exécution (il mémorise sa combinaison).

**CO** contient l'adresse de l'instruction à faire ou en cours.

**Accus** contiennent les données en cours de traitement (nombres, caractères, ...).

**RAM** contient l'adresse de l'emplacement du mot échangé entre l'UC et la mémoire.

**RE** contient le mot échangé entre l'UC et la mémoire.

#### 2. Séquenceur

Le séquenceur est l'organe qui envoie les signaux à tous les autres modules en fonction de l'instruction à exécuter.

Exemple de séquence :

- Charger le RI avec le mot mémoire dont l'adresse est donnée par CO.
  - Suivant la valeur du mot (code instruction), activer les ordres de sélection sur les registres, l'organe de calcul, ...
  - Positionner le CO à l'adresse de la prochaine instruction à exécuter.
- Cycle ←
- Fetch ←

### 3. Format des instructions

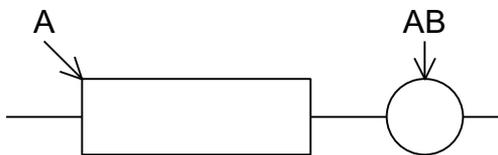


### 4. Deux concepts importants

- L'instruction ne doit agir que sur les accus qui la concerne et doit conserver les autres intacts. Ce n'est pas toujours le cas car suivant les structures il peut arriver que peu de registres soient disponibles.
- Par contre les autres registres peuvent ou non être modifiés selon leur nature (CO, RE, RAM)

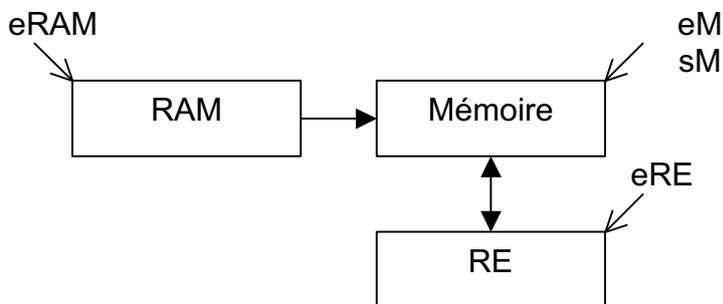
### 5. Présentation des structures

#### Registre



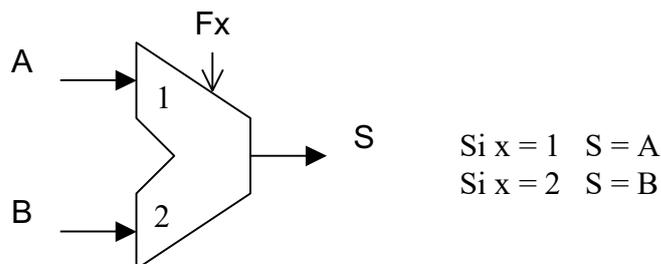
A : Ecriture du contenu du bus sur le registre  
 AB : Libération du contenu du registre sur le bus.

#### Echange avec la mémoire

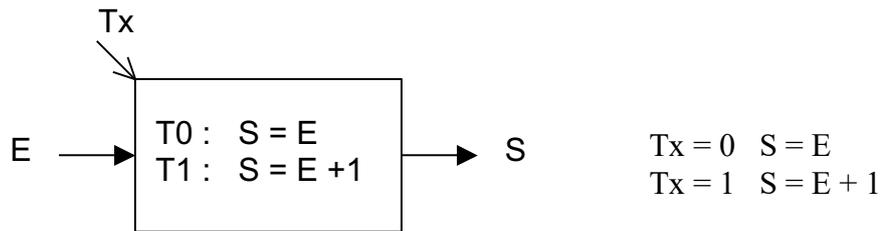


sM : Lecture de la mémoire et écriture du contenu dans RE.  
 eM : Ecriture du contenu de RE dans la mémoire.

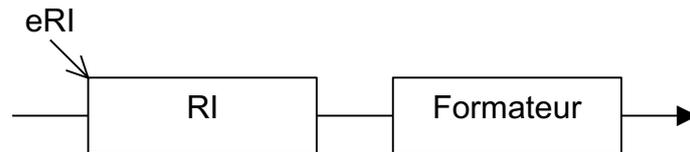
#### Multiplexeur



## Opérateur



## Formateur



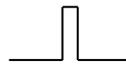
Il permet d'accéder directement à la référence d'adressage (RA). Comme les mots mémoire ont un nombre de bits fixé, les bits relatifs au COP et au MA sont tous à 0 par exemple (suivant le constructeur)



## 6. Signaux

### Pulse à T/2

- Ecriture des registres
- Signaux de commande de la mémoire



Structure S1 : Si eCO signal à niveau, COB à niveau et T1, le registre CO va subir une perpétuelle incrémentation et on ne pourra pas accéder à l'adresse de l'instruction désirée.

### A niveau

-> Tous les autres signaux



### Représentation des différentes phases des signaux

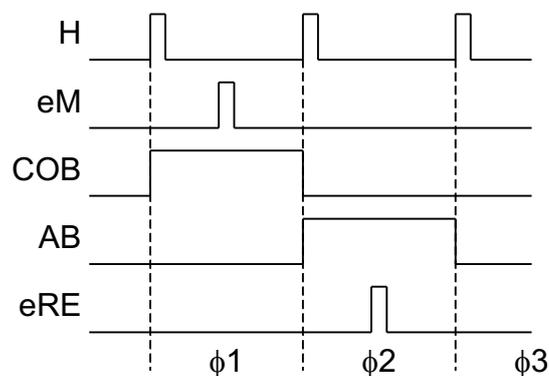
a.

$\phi 1$  : eM, COB, ...

$\phi 2$  : AB, eRE, ...

$\phi 3$  : ...

b. Chronogramme



## **B. QUESTIONS**

### **Exercice 1 : Cycle Fetch**

Détailler le cycle Fetch.

#### Structure S3

Ph1 : COB1, XS, eRAM

Ph2 : sM, COB1, XP1, eCO

Ph2 : REB1, XS, eRI

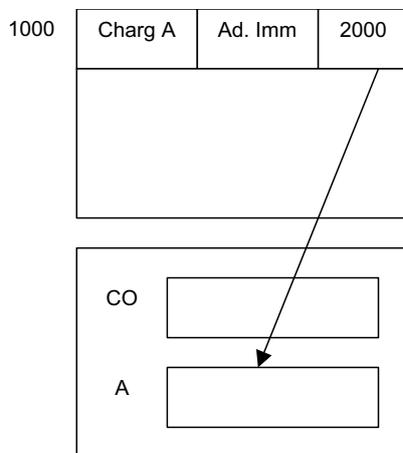
Libération de l'adresse sur le bus et écriture dans RAM.

Lecture de l'instruction dans la mémoire et écriture dans RE.

Libération de l'adresse sur le bus et incrémentation. Ecriture de la nouvelle adresse dans CO.

Libération de l'instruction sur le bus et écriture dans RI.

### **Exercice 2 : Mode d'Adressage Immédiat**



**LOAD A, IMM, valeur**

Ph1 à 3 : Fetch TD

Ph4 : RIB1, XS, eA, FIN

### **Exercice 3 : Mode d'Adressage Direct**

**STORE A, DIRECT, RA**

Ph1 à 3 : Fetch TD

Ph4 : RIB1, XS, eRAM

Ph5 : AB1, XS, eRE

Ph6 : eM, FIN

**LOAD A, DIRECT, RA**

Ph1 à 3 : Fetch TD

Ph4 : RIB1, XS, eRAM

Ph5 : sM

Ph6 : REB1, XS, eA, FIN

**ADD B, DIRECT, RA**

Ph1 à 3 : Fetch TD

Ph4 : RIB1, XS, eRAM

Ph5 : sM

Ph6 : REB1, BB2, ADD, eB, FIN

## **Exercice 4 : Mode d'Adressage Indirect**

### **LOAD A, DIRECT, RA**

Ph1 à 3 : Fetch TD

Ph4 : RIB1, XS, eRAM

Ph5 : sM

Ph6 : REB1, XS, eRAM

Ph7 : sM

Ph8 : REB1, XS, eA, FIN

## **Exercice 5 : Mode d'Adressage Relatif**

### **LOAD A, Relatif, RA**

Ph1 à 3 : Fetch Cours (par d'incrémentation de CO)

Ph4 : RIB1, COB2, ADD, eRAM

Ph5 : sM, COB1, XP1, eCO

Ph6 : REB1, XS, eA, FIN

## **Exercice 6 : Mode d'Adressage Indéxé**

### **STORE A, INDEXE, RA**

Ph1 à 3 : Fetch TD

Ph4 : RIB1, XB2, ADD, eRAM

Ph5 : AB1, XS, eRE

Ph6 : eM, FIN

## **Exercice 7 : Instruction JUMP**

### **JUMP, Direct, RA**

Ph1 à 3 : Fetch TD

Ph4 : RIB1, XS, eCO, FIN

### **JUMP Conditionnel, Relatif, RA**

Ph1 à 3 : Fetch Cours

Condition vraie

Ph4 : RIB1, COB2, ADDX, eCO, FIN

Condition fausse

Ph4 : COB1, XP1, eCO, FIN

## **Exercice 8 : Adressage étendue**

### **LOAD A, IMM ET, valeur**

Ph1 à 3 : Fetch Cours

Ph4 : COB1, XP1, eRAM, eCO

Ph5 : sM, COB1, XP1, eCO

Ph6 : REB1, XS eA, Fin

**LOAD A, DIRECT ET, valeur**

Ph1 à 3 : Fetch Cours

Ph4 : COB1, XP1, eRAM, eCO

Ph5: sM

Ph6 : REB1, XS, eRAM

Ph7 : sM, COB1, XP1, eCO

Ph8 : REB1, XS eA, Fin

...