

Durée: 2 heures

Tous documents interdits

Exercice 1 : (4 points) - Mémoires associatives.

On veut réaliser un contrôleur de mémoire cache avec une organisation purement associative.
Pour cela, on dispose d'un index.

La mémoire centrale a une capacité de 2^{30} mots. La mémoire cache et son index ont une même capacité de 2^{20} mots.

Lorsque l'UC fait une lecture, le mot est en même temps recherché dans la mémoire cache et dans la mémoire centrale.

Rappels :

- L'adresse utilisée par l'Unité Centrale (UC) pour la recherche, est l'adresse en mémoire centrale.
- Si le mot est trouvé en mémoire cache, l'UC peut le lire depuis cette mémoire et il n'est pas lu en MC.
- Sinon, le mot est lu en MC, il est transmis à l'UC, et il est copié en mémoire cache.

Dans ce dernier cas, les opérations sur l'index se déroulent de la manière suivante :

- A. L'adresse en mémoire centrale ADR du mot est copiée dans le premier mot libre de l'index, le contenu du mot est copié en mémoire cache dans l'emplacement correspondant, et le mot est marqué « occupé » dans l'index.
- B. Si aucun mot « libre » n'est trouvé dans l'index, tous les mots de l'index sont réinitialisés à '0' et tous les mots sont marqués « libre ». Le mot est alors copié dans le premier mot libre et l'opération A est alors effectuée.

Questions :

1. Quel est le type de mémoire utilisé pour l'index ?
2. Quelle est le nombre de bits de chaque mot de l'index ?
3. Quel mécanisme peut être utilisé pour marquer les mots « occupé » ou « libre » dans l'index ?
4. Ecrire le programme le plus court qui effectue les opérations A et B décrites ci-dessus.

Remarque : vous ne devez représenter que les opérations dans l'index et non pas dans la mémoire cache.

ATTENTION : seuls les programmes qui consomment un minimum de temps seront considérés.

Exercice 2 : (4 points) - Systèmes d'IT

Sur la station spatiale internationale, le calculateur principal permet d'effectuer les opérations suivantes :

- 0 - Surveiller les deux alimentations électriques A0 et A1, et déclencher un signal en cas de coupure de courant.
 - 1 - Surveiller le niveau d'oxygène dans les 4 salles principales SP0 à SP3 et déclencher une alarme si le niveau est inférieur à un certain seuil.
 - 2 - Surveiller la température dans les 2 centres de calcul CC0 et CC1 et déclencher une alarme si la température dépasse un seuil.
 - 3 - Surveiller les 4 portes extérieures P0 à P3 et déclencher une alarme si l'une des portes est ouverte.
 - 4 - Exécuter des programmes de communication (on suppose qu'on exécute un programme à la fois).
- Pour cela, la machine doit disposer d'un système d'interruption adapté.

Questions:

- 1- Définir les niveaux nécessaires pour le système d'interruption qu'on veut réaliser, et faire un schéma détaillé de ce système en indiquant le contenu des registres importants au démarrage (il est inutile de représenter sur le schéma le niveau réservé aux programmes).

Soit la séquence d'exécution suivante :

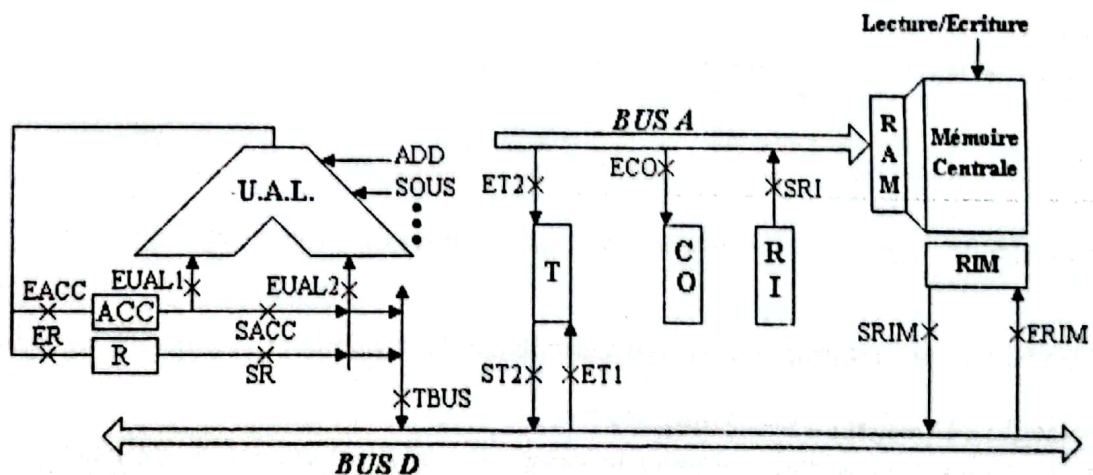
Temps	Programme	Durée (unités temps)
T	Initialisation	
T+50	Lancement d'un programme de communication	30
T+60	Arrivée d'un signal d'alarme de la porte P3 qui est ouverte	50
T+64	Arrivée d'un signal d'alarme pour une température anormale dans le centre de calcul CC1 pendant l'étape précédente ;	50
T+70	Arrivée d'un signal d'alarme sur le niveau d'oxygène dans SP0 pendant l'étape précédente ;	20
	Fin de tous les programmes.	

Question :

- 1- Faire un schéma complet de la séquence en affectant à chaque étape une étiquette (exemple : **Instant A**, ...). Identifiez clairement chacune des étapes importantes de la séquence. Indiquer le contenu de la pile et du registre masque à chaque instant.
- 2- Donner le temps total nécessaire à l'exécution de toutes les tâches de l'initialisation à la fin de tous les programmes.

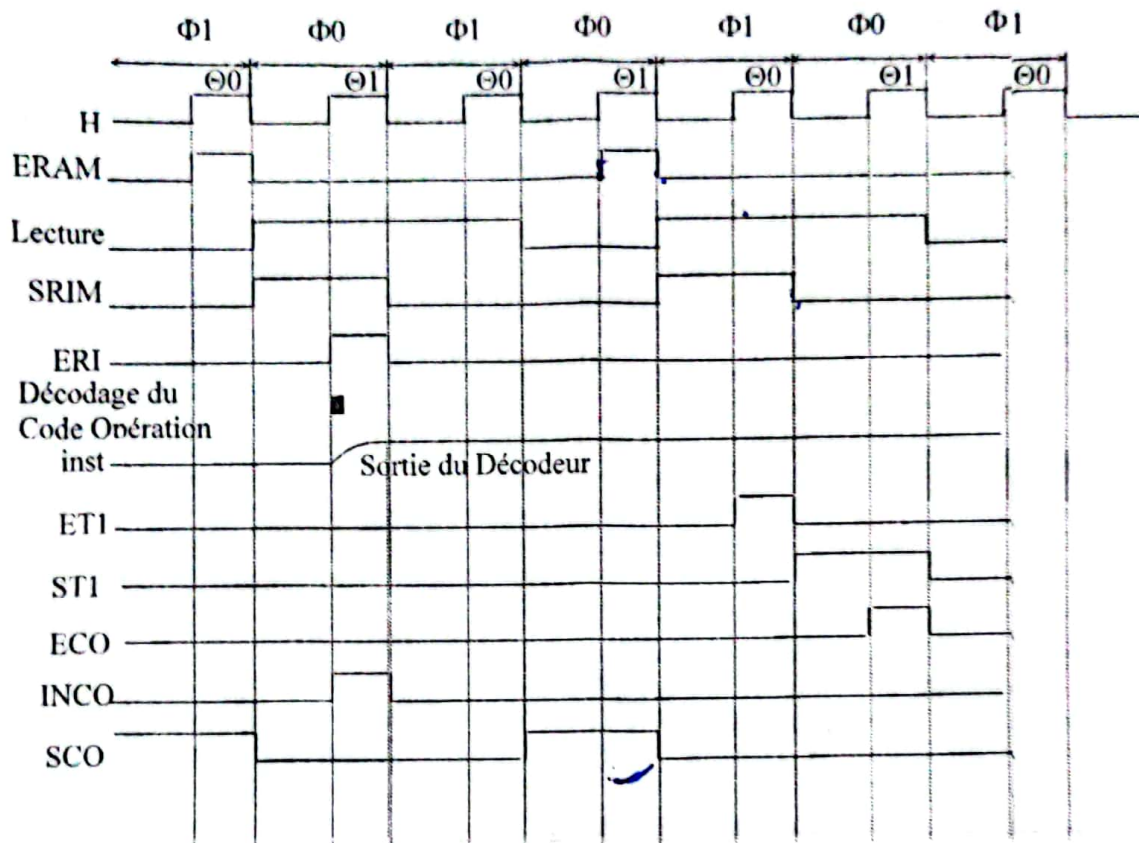
Exercice 3 : (4 points) Le séquenceur

Le chemin de données de la machine que l'on veut concevoir est le suivant.



- Le cycle mémoire nécessite deux phases. Lors d'une opération de lecture, l'information est disponible en première phase. Lors d'une opération d'écriture, l'information est écrite dans le mot mémoire durant la deuxième phase.
- Une opération dans l'UAL nécessite une phase.
- Pour les instructions de format court, le RI contient le Code opération, le mode d'adressage et l'opérande (ou l'adresse) ;
- Pour les instructions de format long, le RI contient le Code opération et le mode d'adressage seulement.

Chronogramme question 1 :



Questions:

1. Considérer le chronogramme qui vous est fourni. Donnez, **sur la feuille qui vous est remise**, le micro-programme détaillé de l'instruction représentée sur le chronogramme (**donner pour chaque étape les microinstructions et l'ensemble des micro-commandes associées**).
2. Indiquez brièvement ce que fait cette instruction (indiquez le code opération, le format de l'instruction et le mode d'adressage).
3. Sur la feuille qui vous est remise, complétez le schéma de la machine avec les microcommandes manquantes.
4. Donner le micro-programme détaillé de l'instruction suivante :
 - CHG *ADR (instruction de format long qui effectue le chargement en **mode indirect**, dans l'accumulateur, du contenu du mot considéré).

Donner pour chaque étape l'ensemble des micro-commandes associées et préciser clairement toutes les étapes pouvant se dérouler en parallèle.

Exercice 4 : (4 points) Entrées-Sorties

Un contrôleur gère deux périphériques dont les numéros sont les suivants :

- ⇒ 0 : Lecteur de disque dur.
- ⇒ 1 : Lecteur le lecteur de DVD.

Pour gérer ces deux périphériques, le contrôleur utilise deux canaux :

- ⇒ Canal 0 pour le lecteur de disque dur.
- ⇒ Canal 1 pour le lecteur de DVD.

Chaque canal contient les registres suivants :

- ⇒ RE_i: Registre d'Etat,
- ⇒ RC_i: Registre de Commandes,
- ⇒ RD_i: Registre de Données,
- ⇒ RP_i: Registre de Piste,
- ⇒ RS_i: Registre de Secteur.

Le contrôleur exécute les commandes suivantes :

- ⇒ **SEEK (Recherche piste)** → code opération : "001"
- ⇒ **READ (Lecture secteur)** → code opération : "010"
- ⇒ **WRITE (Ecriture secteur)** → code opération : "011"
- ⇒ **STOP (Fin du programme d'entrée/sortie)** → code opération : "000"

L'unité centrale accède aux registres des canaux du contrôleur en utilisant les instructions suivantes :

- ⇒ **LDI val** : chargement immédiat de l'accumulateur avec la valeur "val".
- ⇒ **LDA RD_i** : chargement du contenu du RD du canal i dans l'accumulateur.
- ⇒ **LDA RE_i** : chargement du contenu du RE du canal i dans l'accumulateur.
- ⇒ **LDA adr** : chargement direct de l'accumulateur avec la valeur contenue dans le mot **adr**.
- ⇒ **LDA*, adr** : chargement de l'accumulateur avec la valeur contenue dans le mot pointé par l'adresse qui se trouve dans le mot mémoire **adr**.
- ⇒ **STA RD_i** : rangement de l'accumulateur dans le RD du canal i.
- ⇒ **STA RC_i** : rangement de l'accumulateur dans le RC du canal i.
- ⇒ **STA RS_i** : rangement de l'accumulateur dans le RS du canal i.
- ⇒ **STA RP_i** : rangement de l'accumulateur dans le RP du canal i.
- ⇒ **STA adr** : rangement direct du contenu de l'accumulateur dans le mot **adr**.
- ⇒ **STA *, adr** : rangement du contenu de l'accumulateur dans le mot pointé par l'adresse qui se trouve dans le mot mémoire **adr**.
- ⇒ **BZ étiquette** : branchement conditionnel à l'étiquette étiquette si le contenu de l'accumulateur est nul.
- ⇒ **BNZ étiquette** : branchement conditionnel à l'étiquette étiquette si le contenu de l'accumulateur n'est pas nul.
- ⇒ **BI étiquette** : branchement inconditionnel à l'étiquette étiquette.
- ⇒ **ADDI val** : addition en mode immédiat de la valeur "val" au contenu de l'accumulateur; résultat dans l'accumulateur. Cette instruction positionne les indicateurs.
- ⇒ **SUBI val** : soustraction en mode immédiat de la valeur "val" au contenu de l'accumulateur; résultat dans l'accumulateur. Cette instruction positionne les indicateurs.
- ⇒ **ANDI 'valeur binaire'** : et logique en mode immédiat entre le contenu de l'accumulateur et la valeur binaire précisée dans le champ opérande; résultat dans l'accumulateur. Cette instruction positionne les indicateurs.

Toutes les opérations doivent se faire dans l'accumulateur. L'unité centrale ne peut faire aucun test directement sur les registres internes du contrôleur. Le contenu du registre à tester doit d'abord être chargé dans l'accumulateur.

Les bits des registres d'état RE_i des deux canaux sont positionnés par le contrôleur pour indiquer :

- ⇒ **B0_i=1** Le périphérique et le canal i sont prêts.
- ⇒ **B1_i=1** La commande exécutée sur le périphérique relié au canal i est terminée.
- ⇒ **B2_i=1** { Le Registre de Données du canal i est plein en lecture.
 { Le Registre de Données du canal i est vide en écriture.

Question :

Ecrire le programme qui permet de transférer directement, sans passer par la mémoire centrale, le dernier octet du secteur 18 de la piste 12 du disque dur vers le premier octet du secteur 16 de la piste 12 du disque dur, et ensuite de transférer ce même octet vers le premier octet du secteur 13 de la piste 12 du disque dur.

Remarques :

- Vous devez utiliser uniquement les instructions qui sont décrites dans le sujet.
- La taille des secteurs est de **1024** octets.
- Chaque programme doit se terminer par une commande **STOP**.