

Contrôle final
d'architecture des ordinateurs- CPI 2

Durée: 2 h.

Tous documents interdits.

Exercice 1 : (7 points)

Sous-compétences à valider :

- Concevoir des systèmes d'interruption de base.
- Concevoir des circuits à base de mémoires associatives.

Partie 1 :

On veut réaliser un système qui surveille les interruptions provoquées par des alertes dans une usine. Le système d'interruptions est hiérarchisé. Il est possible de valider ou invalider le système d'It, Masquer un niveau et inhiber une cause. On doit pouvoir connecter 12 causes d'interruption, réparties comme suit:

Niveau 0:	* Cause 0	: It alimentation;
Niveau 1:	* Cause 1	: It salle 1;
	* Cause 2	: It salle 2;
	* Cause 3	: It salle 3;
	* Cause 4	: It salle 4;
Niveau 2:	* Cause 5	: It contrôleur d'Entrées/Sorties;
	* Cause 6	: It contrôleur d'Entrées/Sorties;
	* Cause 7	: It contrôleur d'Entrées/Sorties;
	* Cause 8	: Cause inhibée;
Niveau 3:	* Cause 9	: It matériel;
	* Cause 10	: It matériel;
	* Cause 11	: It matériel;

Remarque importante :

Un Niveau 4 existe, mais il est réservé aux interruptions internes (ce niveau ne doit pas apparaître sur le schéma mais un bit lui sera réservé dans le registre masque).

Soit la séquence d'interruptions suivante :

Temps	Programme	Durée (unités temps)
T	Initialisation	
T+50	Lancement d'un programme de communication de niveau 4	30
T+55	Arrivée d'un signal d'It de niveau 2, cause 8 ;	40
T+70	Arrivée d'un signal d'It de niveau 3, cause 9 ;	50
T+90	Arrivée d'un signal d'alarme de la chambre 2;	20
	Fin de tous les programmes.	

Questions :

1. Faire un schéma détaillé du système d'It **en indiquant le contenu des registres importants au démarrage de la machine.** Donner le temps total que prendra l'exécution de cette séquence.
2. Représenter sur un schéma la séquence en précisant les instants des différents événements et en associant à chaque étape une étiquette (Instant A, Instant B...). Vous devrez identifier clairement chacune des étapes importantes de la séquence. **Indiquer le contenu de la pile et du registre masque à chaque instant en incluant le niveau 4 dans le registre masque.**

Partie 2 :

On veut sauvegarder l'état des interruptions qui arrivent, et connaître le nombre d'interruptions pour chaque type de causes. Pour cela, on utilise une mémoire associative qui contient les champs suivants :

- Numéro du niveau ;
- Numéro de la cause ;
- Etat de l'interruption à l'instant T ;
- Nombre de fois que l'interruption s'est déclenchée depuis que la machine a été allumée. **On supposera qu'elle ne peut pas se déclencher plus de 15 fois ;**
- Trois bits non utilisés.

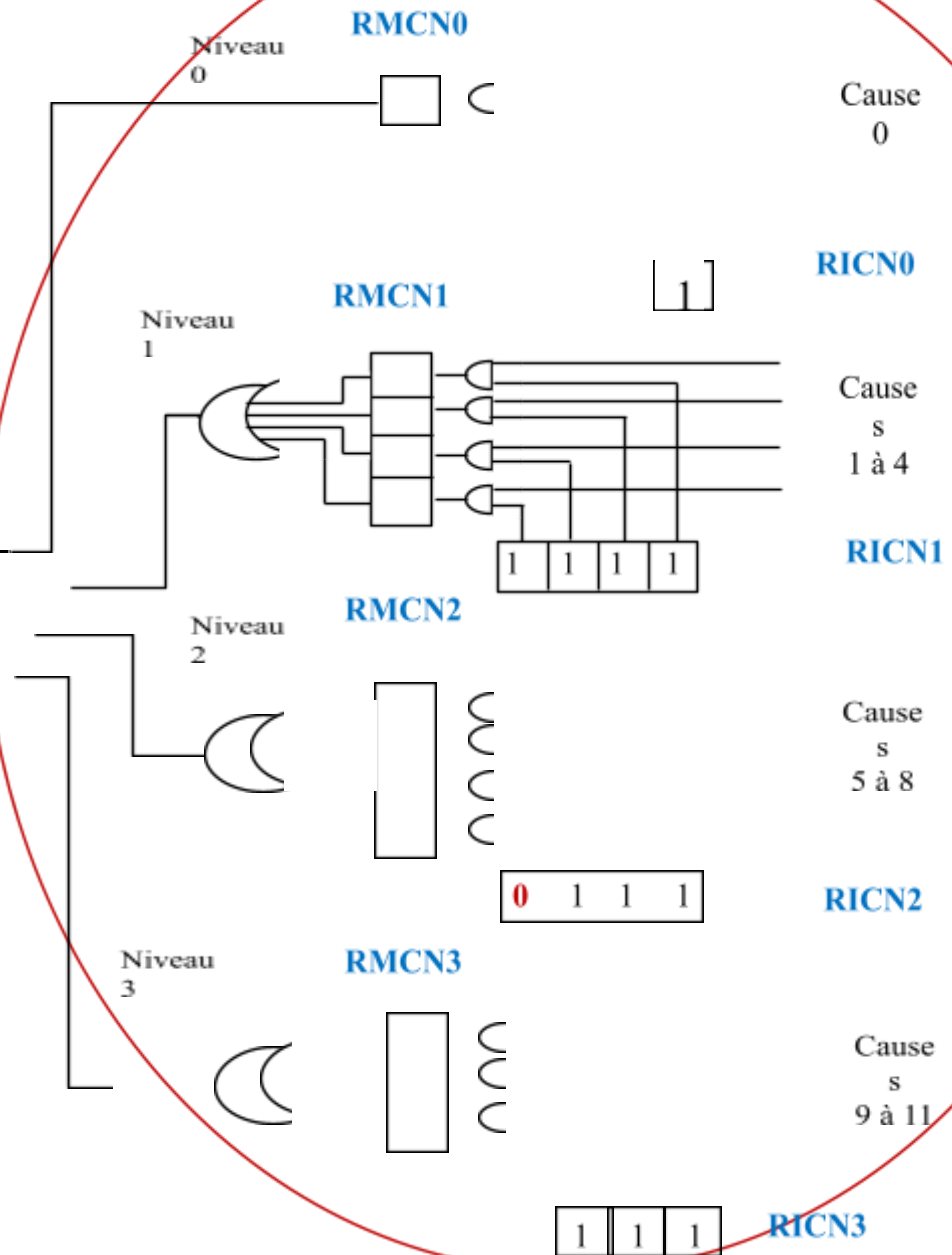
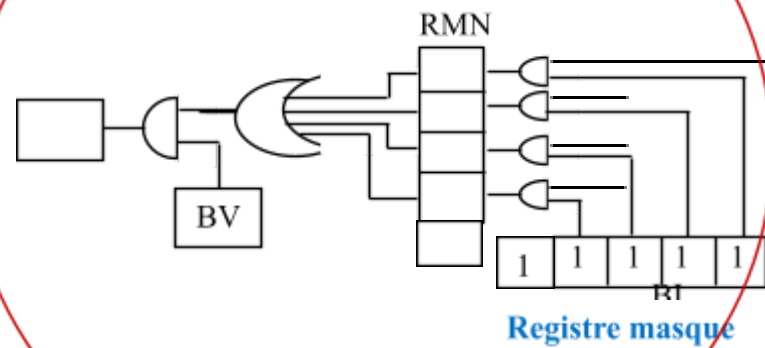
Lorsqu'une nouvelle interruption arrive et qu'elle est prise en compte par le système d'interruption, deux cas de figure peuvent se présenter :

- A- Le niveau et la cause d'interruption sont déjà enregistrés dans la mémoire associative, dans ce cas les opérations suivantes sont réalisées :
 - a. Mettre à '1' le champ correspondant à l'état de l'interruption pour signaler que l'interruption est active et prise en compte.
 - b. Incrémenter le champ correspondant au nombre de fois que l'interruption s'est déclenchée depuis que la machine a été allumée.
- B- Le niveau et la cause ne se trouvent pas dans la mémoire associative. Dans ce cas, il faut enregistrer cette interruption dans la mémoire et exécuter les étapes (a) et (b) de l'opération 'A' ci-dessus.

Questions :

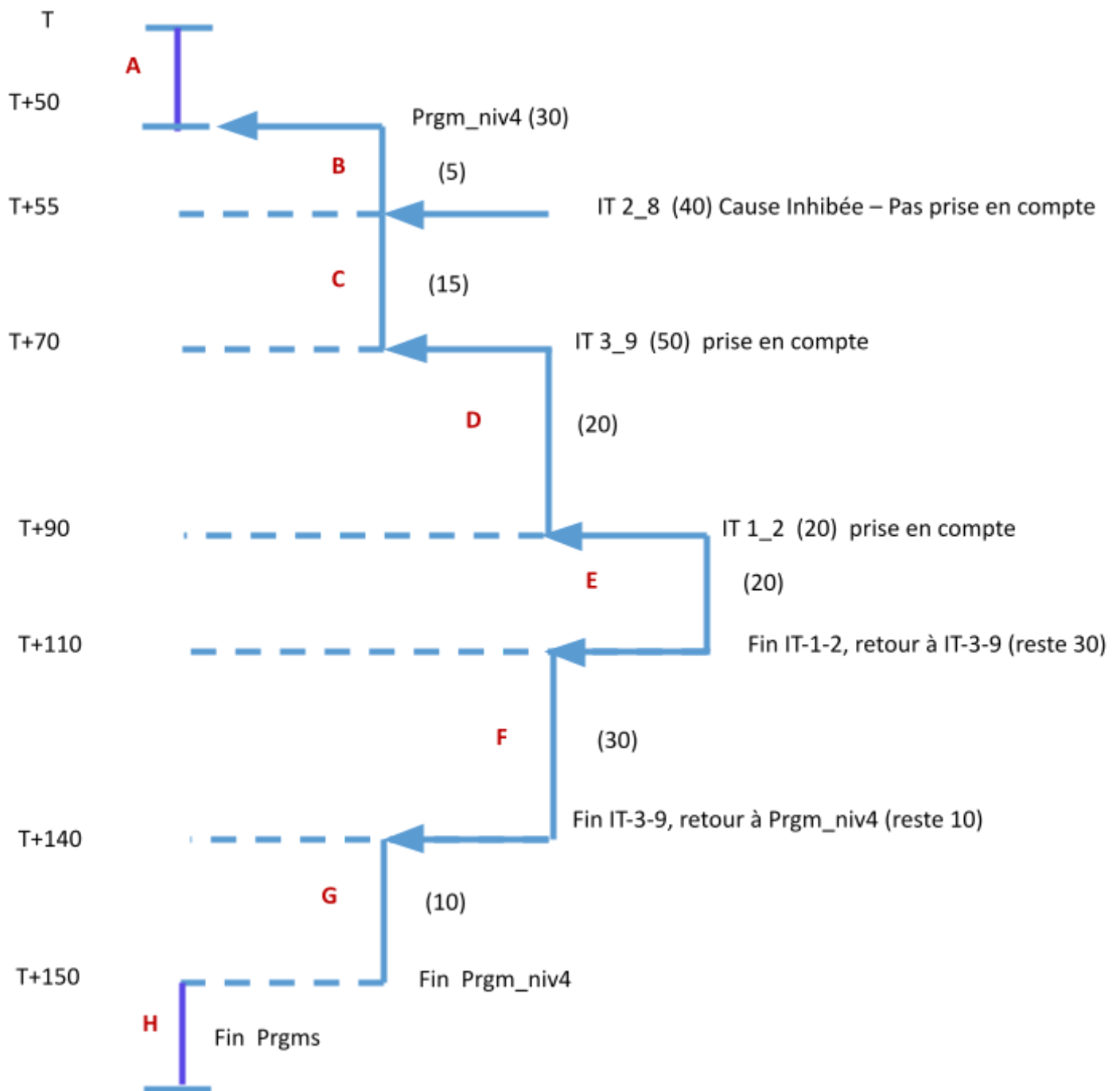
3. Représenter le registre clé, avec les différents champs et la taille de chaque champ.
4. Donner l'algorithme qui enregistre/met à jour le mot mémoire correspondant à l'interruption qui a été prise en compte à l'instant T.

(1.25Pts), La note complète est attribuée si le schéma est complet, incluant les contenus et les noms de tous les registres.



Question2 :

2-1) schéma de la séquence



Nom :	Prénom :
Groupe :	

1Pts

2-2 le contenu de la pile et du registre masque à chaque instant (0.75 Pt)

Registre masque :

N4				N0
----	--	--	--	----

Pile :

1 1 1 1 1

--

0 1 1 1 1

--

0 0 1 1 1

CTX Prg_niv4

0 0 0 0 1

CTX IT 3-9
CTX Prg_niv4

A,H B,C,G D,F E

- Le temps total que prendra l'exécution de cette séquence est 150(0.5 Pt)
- Question 3 :(0.5 Pt)

Nom :	Prénom :
Groupe :	

3 bits inutilisés	Nombre de fois que l'interruption s'est déclenchée depuis que la machine a été allumée	Etat de l'interruption à l'instant T	Numéro de la cause	Numéro du niveau
X X X	4 bits	1 bit	4 bits	3 bits

Bx

Question 4 : Pour un instant T (3 Pts)

Etiquett e	Instructions	Commentaires	Barème
	Reset, S/I <input type="checkbox"/> '0'	Initialisations	
	C[Bx] <input type="checkbox"/> '0' M[Bx] <input type="checkbox"/> '1', 0 ailleurs SET Ecriture	Utiliser un bit parmi ceux inutilisés afin de distinguer les mots de mémoire libres et occupés, ce qui est nécessaire dans le cas de l'insertion d'une nouvelle interruption.	
	Clé.Numniv <input type="checkbox"/> Numniv Clé.Numcause <input type="checkbox"/> Numcause	Recherche du niveau et de la cause dans la MA	
	Masque.Numniv <input type="checkbox"/> '1', Masque.Numcause <input type="checkbox"/> '1', 0 ailleurs		
	Set		
	Recherche		
	Si S/N=0 aller à Insertion	Si le niveau et la cause n'existent pas dans la MA aller à l'insertion	
	Lecture	Mettre à jour le mot mémoire correspondant à l'interruption qui a été prise en compte <u>à l'instant T.</u>	
	Clé.nbrelT <input type="checkbox"/> R.sortie.nbrelT+ 1 Clé.EtatIT <input type="checkbox"/> '1'		
	Masque.nbrelT <input type="checkbox"/> '1' Masque. EtatIT <input type="checkbox"/> '1'		
	Ecriture		
Fin	Stop	Fin de programme	
Insertion	C[Bx] <input type="checkbox"/> '0' M[Bx] <input type="checkbox"/> '1', 0 ailleurs		
	Set		
	Recherche		
	S/I <input type="checkbox"/> '1'		
	Clé.nbrelT <input type="checkbox"/> '1' Clé.EtatIT <input type="checkbox"/> '1' C[Bx] <input type="checkbox"/> '1'	Les clés de Numniv et Numcause sont déjà chargés	
	Masque <input type="checkbox"/> '1...1'	Masque. Numniv <input type="checkbox"/> '1' ; Masque.Numcause <input type="checkbox"/> '1' Masque. nbrelT <input type="checkbox"/> '1', Masque. EtatIT <input type="checkbox"/> '1'	
	Ecriture, BT <input type="checkbox"/> '1'	Insertion du mot mémoire correspondant à l'interruption qui a été prise en compte <u>à l'instant T.</u>	
	S/I <input type="checkbox"/> '0'		

Nom :	Prénom :
Groupe :	

	Aller à fin		
--	-------------	--	--

Exercice 2 : (2 points)

Sous-compétences à valider :

- Lister les composants de base d'un système d'Entrées/Sorties.
- Expliquer le fonctionnement d'un système d'Entrées/Sorties évolué.

Soit un contrôleur DMA évolué (processeur d'Entrées/Sorties) qui permet de gérer quatre périphériques. Pour cela, il utilise des canaux physiques (il existe un jeu de registres internes pour chaque périphérique géré).

Questions :

- Donner le schéma interne du contrôleur (montrer l'ensemble des registres), et montrer sur le schéma tous les signaux entre les différents circuits (UC, périphériques, arbitre, MC...).
- Donner les différentes étapes de déroulement d'une Entrée/Sortie.

Réponse

Question 1

Architecture d'un DMAC (0.5)

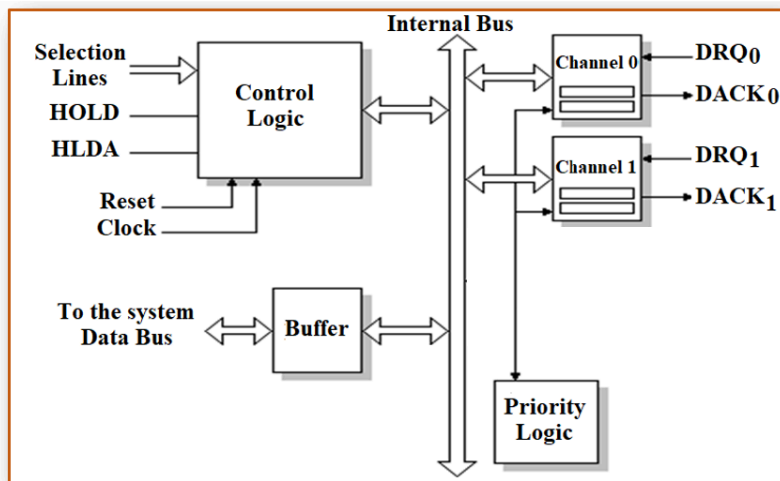
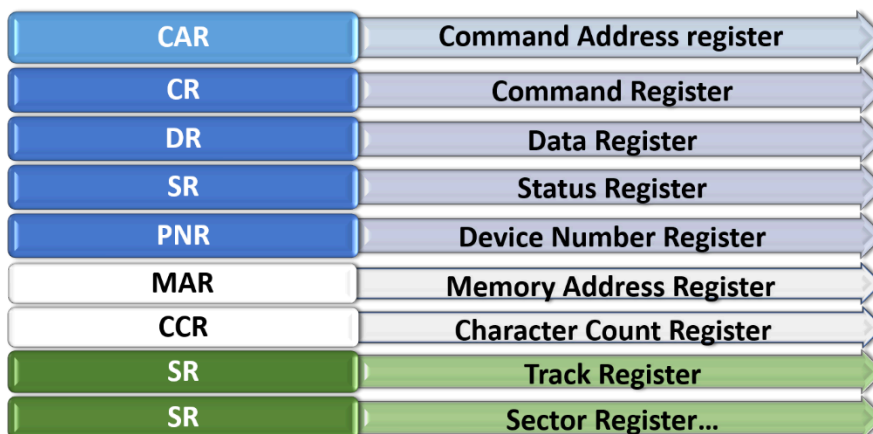


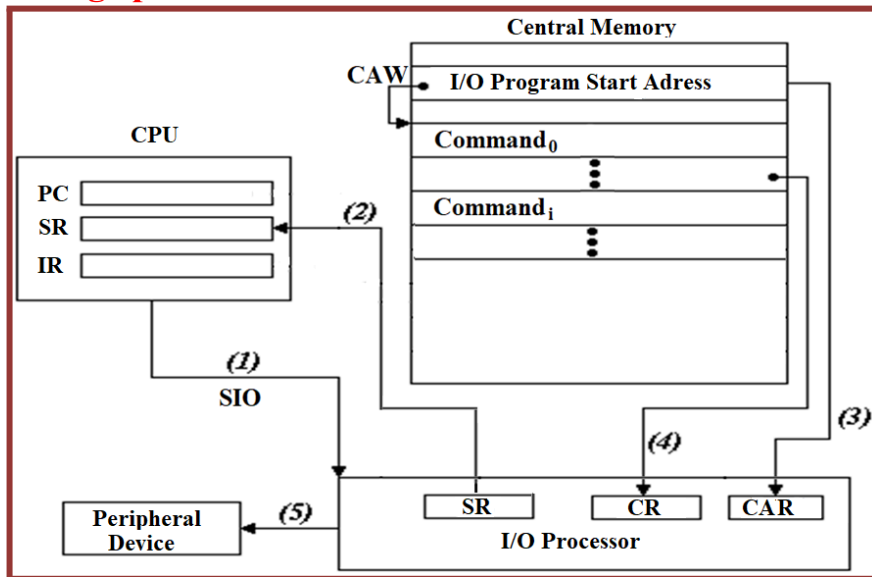
Schéma interne d'un canal (0.5)



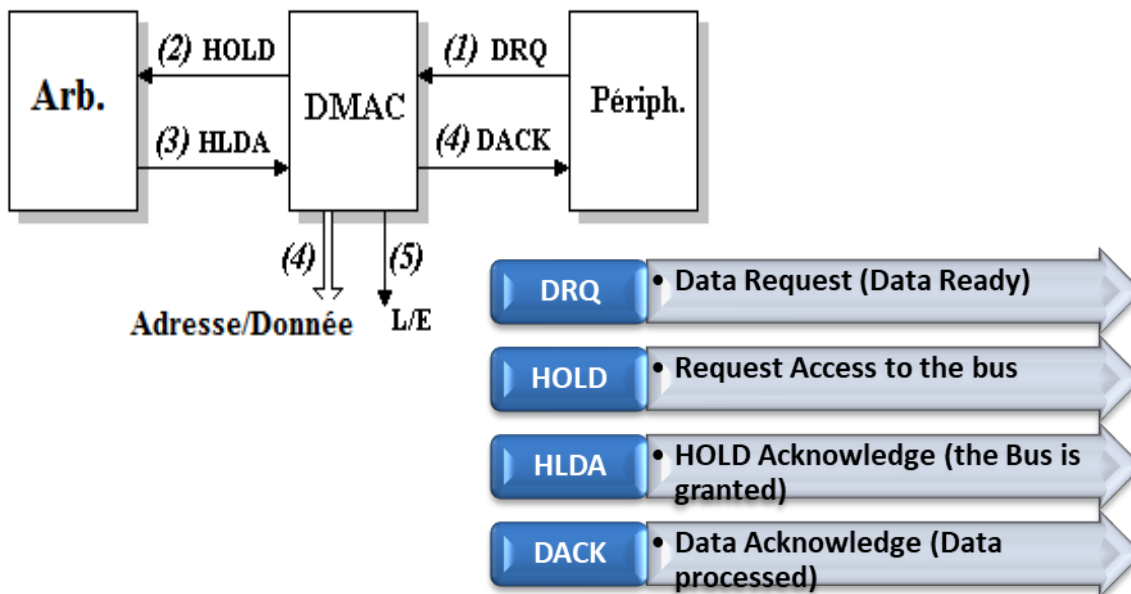
Nom :	Prénom :
Groupe :	

Question 2 (1pt)

Dialogue pour l'initialisation de l'E/S



Dialogue durant la phase de transfert



Nom :	Prénom :
Groupe :	

Exercice 3:(5 points)

Sous-compétences à valider : programmer un contrôleur de communication élémentaire.

Un programme lance une instruction d'entrée sortie dont le rôle est d'écrire une chaîne de caractères qui se trouve en mémoire centrale, sur un disque externe.

Le numéro de piste où doit être écrite la chaîne sur le disque se trouve à l'adresse 1500 en MC.

Le numéro de secteur se trouve en mémoire centrale à l'adresse 1501.

Le nombre d'octets de la chaîne de caractères se trouve en mémoire centrale à l'adresse 1502.

La chaîne de caractères commence à l'adresse 1505 en mémoire centrale.

Pour exécuter cette instruction, l'UC exécute un programme pour commander le contrôleur de disque.

Les registres du contrôleur sont les suivants :

- ⇒ **RE**:Registre d'Etat.
- ⇒ **RC**:Registre de Commandes,
- ⇒ **RD**:Registre de Données,
- ⇒ **RS** :Registre de Secteur,
- ⇒ **RP**:Registre de Piste,

L'unité centrale accède aux registres du contrôleur et aux mots mémoire en utilisant les instructions suivantes:

- ⇒ **LDI val** : chargement immédiat de l'accumulateur avec la valeur "val".
- ⇒ **LDA RD** : chargement du contenu du RD dans l'accumulateur.
- ⇒ **LDA RE** : chargement du contenu du RE 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** : rangement de l'accumulateur dans le RD.
- ⇒ **STA RC** : rangement de l'accumulateur dans le RC.
- ⇒ **STA RS** : rangement de l'accumulateur dans le RS.
- ⇒ **STA RP** : rangement de l'accumulateur dans le RP.
- ⇒ **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**.

L'UC exécute également les instructions suivantes :

- ⇒ **BZ étiquette**: branchement conditionnel à l'étiquette étiqu si le contenu de l'accumulateur est nul.
- ⇒ **BNZ étiquette**: branchement conditionnel à l'étiquette étiqu si le contenu de l'accumulateur n'est pas nul.
- ⇒ **BI étiquette**: branchement inconditionnel à l'étiquette étiqu.
- ⇒ **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.

Nom :	Prénom :
Groupe :	

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.

Chacun des registres du contrôleur ainsi que l'accumulateur ont une taille de 8 bits.

Nom :	Prénom :
Groupe :	

Le contrôleur exécute les commandes suivantes:

- ⇒ **SEEK (Recherche piste) →code opération: "000"**
- ⇒ **Read (Lecture secteur) →code opération: "001"**
- ⇒ **WRITE (Ecriture secteur) →code opération: "010"**
- ⇒ **STOP (Fin du programme d'entrée/sortie) →code opération: "111"**

Les bits du registre d'état **RE** sont positionnés par le contrôleur pour indiquer:

- ⇒ **B0=1** Le Registre de Données est plein en lecture.
Le Registre de Données est vide en écriture.
- ⇒ **B1=1** La commande exécutée sur le périphérique relié est terminée.
- ⇒ **B2=1** Le périphérique et le contrôleur sont prêts.
- ⇒ **B3=1** Indicateur général d'erreur (erreur sur l'entrée/sortie).
- ⇒ **B4=1** Erreur de formatage sur le disque (code erreur: 00).
- ⇒ **B5=1** Erreur d'écriture sur le disque (code erreur: 01).
- ⇒ **B6=1** Erreur de lecture sur le disque (code erreur: 10).
- ⇒ **B7=1** Erreur Numéro de piste inexistant (code erreur: 11).

Question:

Ecrire le programme **exécuté par l'unité centrale** pour écrire la chaîne de caractères sur le disque.

Remarques :

- L'UC doit vérifier si aucune erreur ne s'est produite. Si une erreur est signalée, son code doit être écrit dans un mot en mémoire centrale, dont le nom est « ERREUR » et qui se trouve à l'adresse 2000.
- Vous devez utiliser uniquement les instructions qui sont décrites dans le sujet.
- Le programme doit se terminer par une commande STOP.

Exercice N°3

Etiquettes	Instructions	barèmes
	LDI 1500 STA VRP LDI 1501 STA VRS LDI 1502 STA NBO LDI 1505 STA ADR LDI 2000 STA ERREUR	1
Test_B2 :	LDA RE ANDI '00000100' BZ Test_B2	0.25
	LDA VRP STA RP	0.5

Nom :	Prénom :
Groupe :	

Test_B1 :	LDA VRS STA RS LDI '000' STA RC LDA RE ANDI '00000010' BZ Test_B1	
Test_B3 :	LDA RE ANDI '00001000' BZ Suite1	1
Test_B4 :	LDA RE ANDI '00010000' BZ Test_B5 LDI '00' STA *,ERREUR BI STOP	
Test_B5 :	LDA RE ANDI '00100000' BZ Test_B6 LDI '01' STA *,ERREUR BI STOP	
Test_B6 :	LDA RE ANDI '01000000' BZ Test_B7 LDI '10' STA *,ERREUR BI STOP	
Test_B7 :	LDI '11' STA *, ERREUR BI STOP	
Suite1:	LDA *, ADR STA RD LDI '010' STA RC LDA RE ANDI '00000001' BZ Test_B0	0.5
Test_B0 :		
Test_B3 :	LDA RE ANDI '00001000' BZ Suite2	1
Test_B4 :	LDA RE ANDI '00010000'	

Nom :	Prénom :
Groupe :	

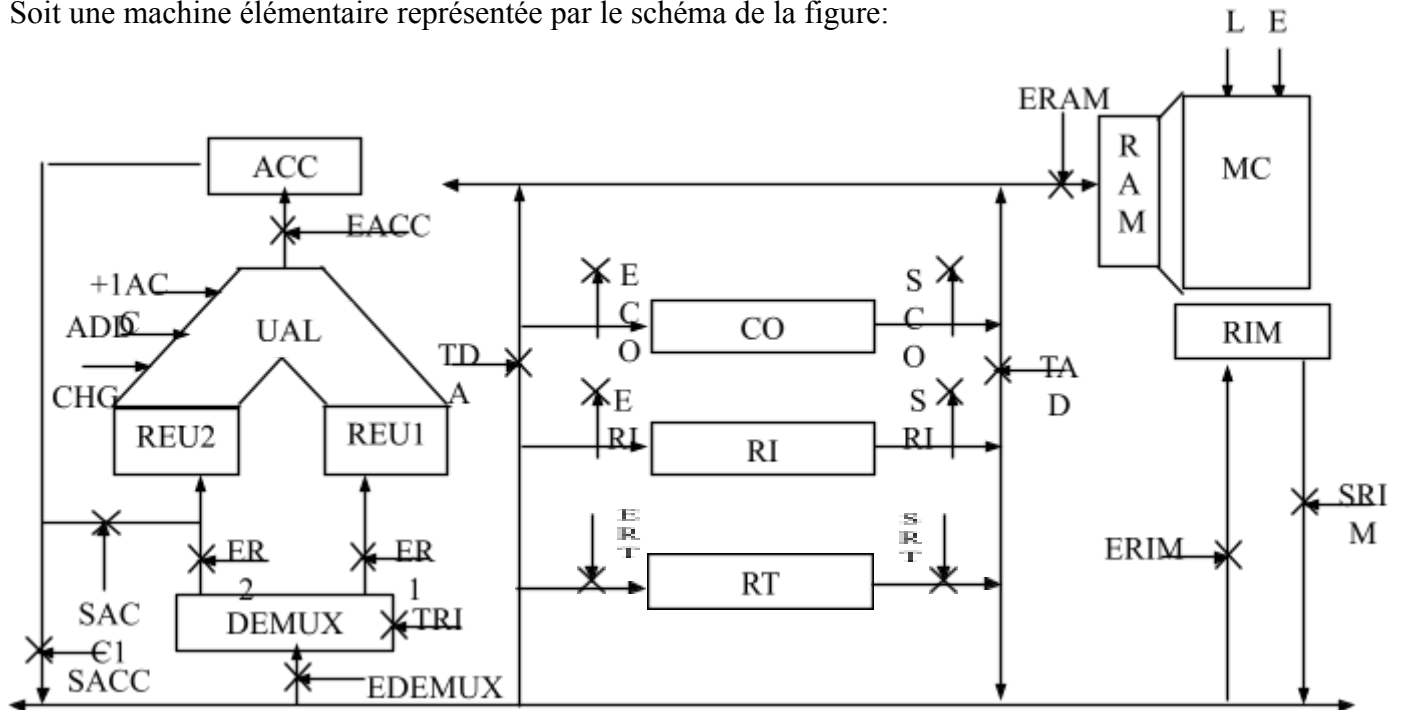
Test_B5 :	BZ Test_B5 LDI '00' STA *,ERREUR BI STOP LDA RE ANDI '00100000' BZ Test_B6 LDI '01' STA *,ERREUR BI STOP	
Test_B6 :	LDA RE ANDI '01000000' BZ Test_B7 LDI '10' STA *,ERREUR BI STOP	
Test_B7 :	LDI '11' STA *, ERREUR BI STOP	
Suite2:	LAD *,NBO SUBI 1 STA *,NBO BZ STOP LDA ADR ADDI 1 STA ADR LDA *, ADR STA RD BI Test_B0	0,5
STOP :	LDI '111' STA RC	0.25

Nom :	Prénom :
Groupe :	

Exercice 4:(4 points).

Compétence à valider : Concevoir un séquenceur élémentaire répondant à des caractéristiques données.

Soit une machine élémentaire représentée par le schéma de la figure:



Nom :	Prénom :
Groupe :	

RT, REU1 et REU2 sont des registres.

DEMUX: Demultiplexeur dont l'entrée adresse est la microcommande TRI :

- Si TRI = 0, alors l'entrée du demultiplexeur est envoyée vers l'entrée du registre REU1;
- Sinon, l'entrée du demultiplexeur est envoyée vers l'entrée du registre REU2;

CHG: Chargement de l'accumulateur avec le contenu du registre REU1;

+1ACC: incrémentation de l'accumulateur

Question :

Donner les micro-instructions et microcommandes de l'instruction de **format court** permettant **l'addition en mode indirect** du contenu de l'accumulateur avec l'opérande dont **l'adresse effective** est contenue dans le mot qui est pointé par l'adresse contenue dans le champ adresse de l'instruction.

Réponse à l'Exercice 4 : (4 points)

Etape	Micro-instructions de l'étape	Micro-commandes de l'étape
Recherche de l'instruction (0.75)	(CO) □ RAM Lecture (RIM) □ RI	SCO, ERAM L SRIM, ERI
Décodage	DEC	DEC

Nom :	Prénom :
Groupe :	

<div>Traitement de l'opération (1.75)</div>	<div>RI.ADR □ RAM Lecture (RIM) □ RAM Lecture (RIM) □ REU1 (ACC) □ REU1 ADD</div>	<div>SRI, TAD, ERAM L SRIM, TDA,ERAM L SRIM,EDEMUX,TRI=0,ER1,SACC1,ADD,EAC C</div>
--	---	---

Nom :	Prénom :
Groupe :	

<div>Préparation de la prochaine instruction (0.5)</div>	<div>(ACC) □ RT (0.5) (CO) □ REU1 (ACC) □ CO (RT) □ ACC (0.5)</div>	<div>SACC, ERT SCO,TAD,EDEMUX,TRI=0,ER1,+1ACC,EACC SACC,TDA,ECO SRT,EDEMUX,TRI=0,ER1,CHG,EACC</div>
---	---	---

Nom :	Prénom :
Groupe :	

Répondre sur la feuille

Question 1 :

Quel sont les avantages et les inconvénients d'un séquenceur microprogrammé ?

Réponse :

Avantage : (0.5)

- Si on veut ajouter une instruction, il suffit d'ajouter le micro-code dans la mémoire de micro-instructions, ainsi, le temps de conception est court et le temps de mise sur le marché est court également.

Inconvénient : (0.5)

- Ce type de séquenceur est moins rapide que le séquenceur câblé.

Question 2 :

Quel est le mode de codage adopté actuellement pour les microcommandes, dans les séquenceurs microprogrammés et pourquoi ?

Réponse :

Type de codage actuel :Le codage par champs séparés (0.5)

Raison : (0.5)

- Il permet le parallélisme entre les micro-commandes, et donc la taille des micro-instructions sont réduites, ainsi que le nombre de mots nécessaires dans la mémoire de micro-programmes.
- Les décodeurs sont de plus petite taille et sont donc plus rapides, permettant ainsi des temps d'exécution plus faibles pour les instructions.