

CONTROLE INTERMEDIARE - ARCHI2

Durée: 2h00

Documents interdits

Exercice 1: (4 points) – conception de circuits mémoires

On veut réaliser une machine avec une mémoire centrale d'une capacité de 256 Giga mots de 64 bits, divisée en 2 modules entrelacés avec un degré d'entrelacement $D = 2$ (l'entrelacement se fait à l'intérieur de chaque module). Cette mémoire est réalisée avec des boîtiers de base de 32 Giga mots de 32 bits.

Questions :

1. Quelle est la taille du bus adresse de cette mémoire ? (0.5 pt)
2. Quelle est la taille d'un module et le nombre de boîtiers de base d'un module ? (0.5 pt)
3. Faire le schéma détaillé d'un seul module, en détaillant toutes les connexions avec les différents bus (bus d'adresses, de données et de commande), et en précisant clairement tous les poids des bits du bus d'adresse global. (3 pts)

Corrigé:

Q1 : Taille du bus adresse (0.5 point)

⇒ Mémoire vive de 256 Giga mots de 64 bits = $2^8 * 2^{30}$ mots = 2^{38} mots → la taille du bus d'adresse est de 38 bits (A0-A37).

Q2 : Taille du module et nombre de boîtiers (0.5 point)

⇒ Taille de chaque module = taille mémoire vive / 2

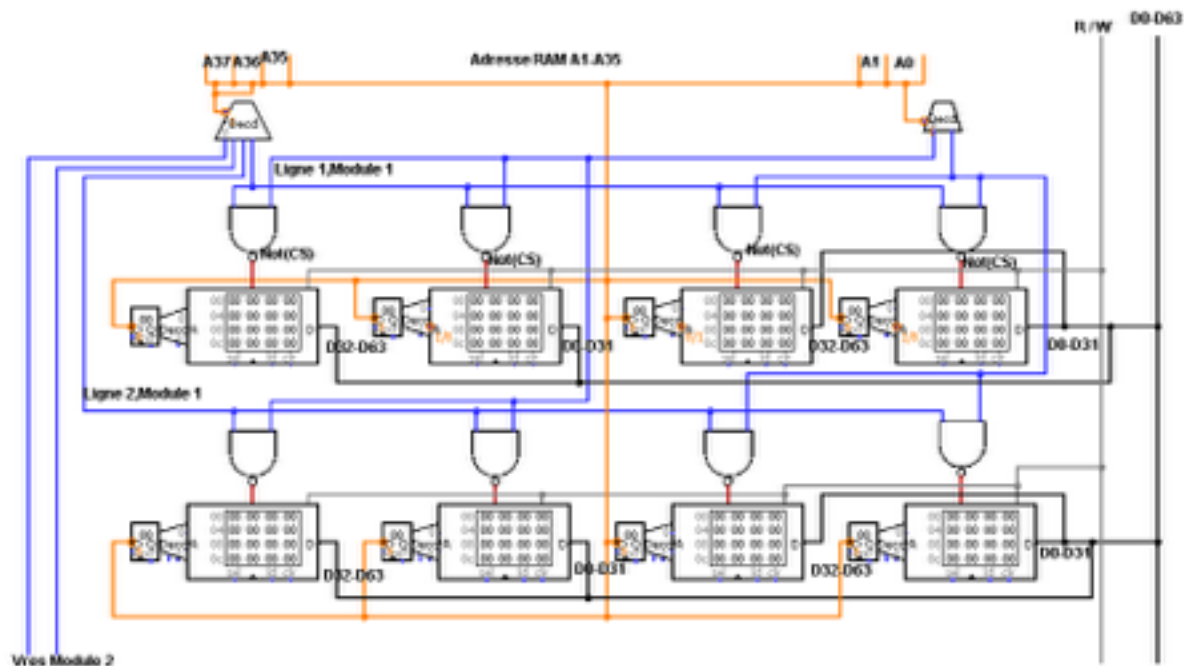
⇒ Nombre de boîtiers par module :

$P = \text{taille module} / \text{taille boitier} = 4$

$Q = 64 / 32 = 2$

$P * Q = 4 * 2 = 8$ boîtiers par module

Q3 : Schéma (3 points)



Exercice 2 : (3 points) – Les mémoires mortes ROM

On veut réaliser le circuit de commande d'une usine de fabrication de voitures, à l'aide d'une ROM.

La fabrication d'une voiture passe par trois phases, qui sont :

- Phase 1 : la fabrication du châssis, de l'ossature de la carrosserie et du moteur ;
- Phase 2 : le montage du moteur sur le châssis ;
- Phase 3 : le montage de la carrosserie sur le châssis et la peinture.

Les différentes phases sont pilotées par une horloge H de période P.

La phase 1 dure un temps égal à $4 \cdot P$; la phase 2 prend un temps de $2 \cdot P$ et la phase 3 un temps de $2 \cdot P$.

Questions :

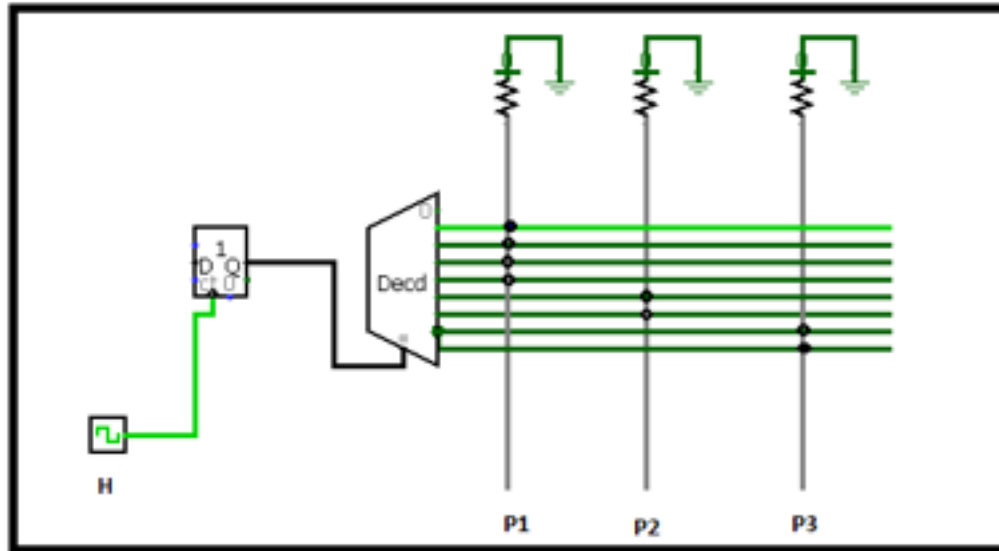
1. Faire la table de vérité du circuit. (1 pt)
2. Réaliser le schéma du circuit avec une ROM de 8 mots et un minimum de circuits. (2 pts)

Corrigé

Q1 : Table de vérité (1 pt)

H	N2	N1	N0	P1	P2	P3
	0	0	0	1	0	0
	0	0	1	1	0	0
	0	1	0	1	0	0
	0	1	1	1	0	0
	1	0	0	0	1	0
	1	0	1	0	1	0
	1	1	0	0	0	1
	1	1	1	0	0	1

Q2 : Schéma (2 pts)



Exercice 3 : (4 points) - Les mémoires cache

Partie A - La mémoire cache est une mémoire statique de taille réduite.

Questions :

1. Lorsque la mémoire cache est pleine, que doit-on faire pour enregistrer de nouvelles données ? (0.5 pt)
2. Citez et expliquez les différentes techniques de remplacement de lignes en mémoire cache. (1.5 pts)

Partie B – Lorsqu'une donnée est modifiée en mémoire cache (suite à une opération d'écriture), la mise à jour de cette dernière en mémoire centrale devient obligatoire.

Questions :

1. Pourquoi doit-on mettre à jour la donnée modifiée en mémoire centrale ? (0.5pt)
2. Citez et explicitez les différentes techniques d'écriture en mémoire centrale pour les données modifiées en mémoire cache. (1.5 pts)

Corrigé

Partie A

Q1 (0.5 pt) : Pour enregistrer de nouvelles données, on doit procéder au **remplacement** d'une ligne en mémoire cache selon une technique de remplacement donnée.

Q2 : Les techniques de remplacement de lignes en mémoire cache (1.5 pts)

On distingue 3 techniques de remplacement de lignes en mémoire cache :

- **Remplacement Aléatoire:** le numéro de la ligne à remplacer est généré aléatoirement.
- **Remplacement direct :** réservé exclusivement à la Mémoire Cache à placement direct; le block i est chargé dans la ligne i correspondante en mémoire cache.

- **Remplacement LRU (Least Recently Used):** la ligne la plus ancienne qui n'a pas été utilisée est choisie pour être remplacée.

Partie B

Q1 (0.5 pt) : la mise à jour de la donnée modifiée en mémoire centrale est impérative pour garder la **cohérence** de la donnée (même donnée implique même valeur aussi bien en mémoire cache qu'en mémoire centrale).

Q2 : Les techniques d'écriture en mémoire centrale (1.5 pts)

On distingue 3 techniques d'écriture en mémoire centrale :

- **L'écriture immédiate (WRITE THROUGH) :** MAJ simultanée en mémoire cache et mémoire centrale.
- **L'écriture retardée (POSTED WRITE THROUGH) :** l'écriture s'effectue dès que le BUS est disponible. Une file FIFO contiendra les mots modifiés.
- **L'écriture différée (WRITE BACK) :** l'écriture s'effectue dès que la ligne en mémoire cache doit être libérée. Un indicateur supplémentaire est utilisé en mémoire cache pour repérer les mots modifiés.

Exercice 4 : (9 points) – FIFO et mémoire associative

On veut automatiser le vote pour les élections des présidents d'APC et des membres de l'assemblée de wilaya qui se font pendant le même vote.

- Les élections sont identifiées par leur code (A ou W).
- Il existe deux listes de candidats :
 - Pour la présidence de l'APC (élection A);
 - Pour l'assemblée de wilaya (élection W);
- Les candidats à chacune des deux élections sont identifiés par un numéro.
- Les votants sont identifiés par leur numéro d'électeur.

Le vote se déroule selon la procédure suivante :

- Chaque votant peut voter pour une ou les deux élections (A), (W), ou (A et W) ;
- Chaque votant doit choisir un candidat (numéro de candidat) pour chaque élection à laquelle il décide de participer.
- Pour chaque élection à laquelle il participe, un votant est enregistré, dans l'ordre d'arrivée, dans la même structure unique, en stockant son numéro d'électeur, ainsi que son choix (code du candidat choisi pour chaque élection) et le code de l'élection (A ou W).
- Après le vote, les informations des votants sont réparties vers deux mémoires associatives nommées A et W (selon le code de l'élection) pour être traitées plus tard.

L'objectif de l'élection est de connaître les élus et le nombre de votants par élection (A ou W).

Questions :

1. Quel type de structure est utilisé pour enregistrer l'arrivée des votants ? (0,5 pt)
2. Définir les champs de cette structure. (0,5 pt)

3. Faire le schéma qui permet de répartir les données de la structure unique vers les deux mémoires associatives en fonction du code de l'élection (A ou W), en précisant les différents liens qui les relient ainsi que les équations logiques de tous les signaux. (2,5 pt)
4. Ecrire le programme qui initialise à 0 les mémoires associatives, puis écrit les mots qui arrivent de la structure unique pour être stockées en fonction du code de l'élection (1,5 pt)
5. Ecrire le programme le plus rapide qui permet de donner le score (nombre de votes) pour chaque candidat (identifié par son numéro) dans l'une des deux élections (il est inutile de répéter le programme pour les deux élections). (4 pts)

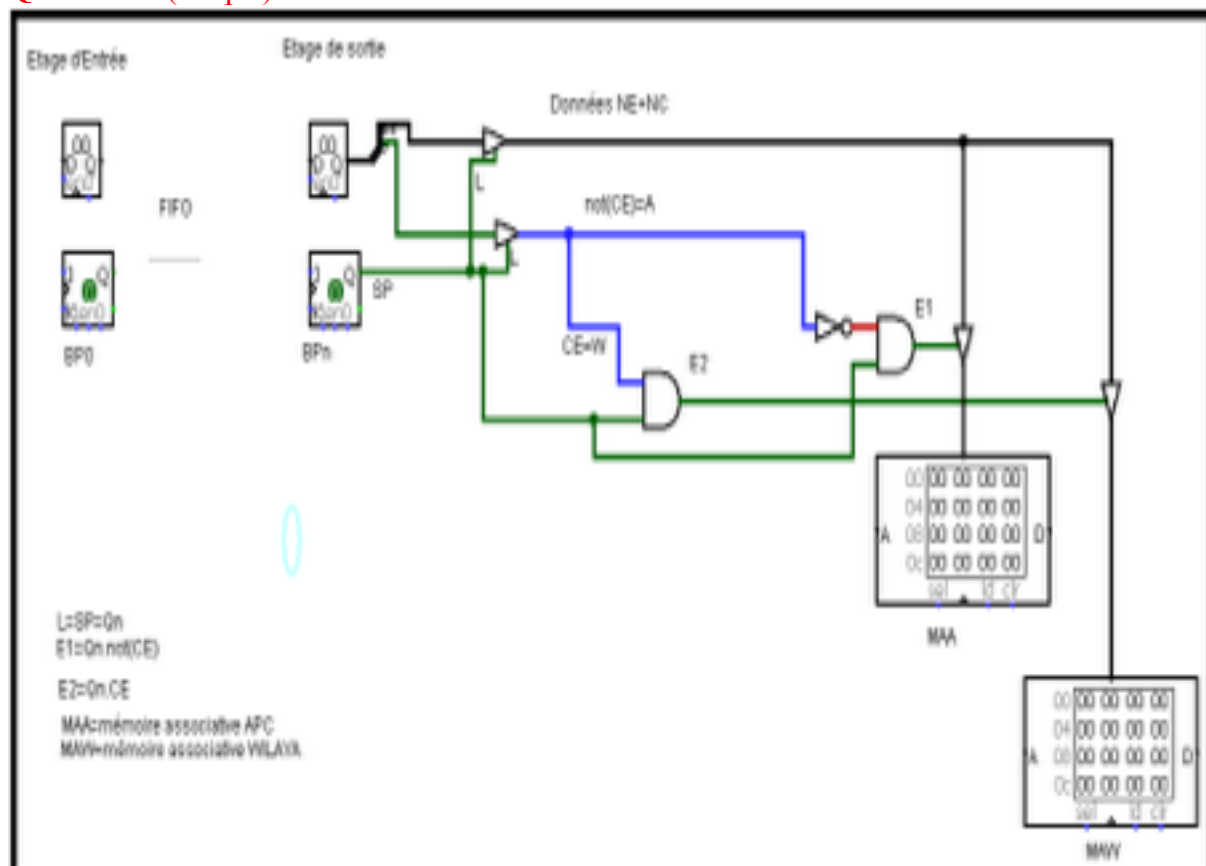
Corrigé

Q1 (0.5 pt) : La structure utilisée pour enregistrer l'arrivée des votants est une mémoire séquentielle de type **FIFO**.

Q2 Les champs de la structure FIFO (0.5 pt) :

Numéro électeur : NE	Numéro Candidat : NC	Code élection : CE
-----------------------------	-----------------------------	---------------------------

Q3 Schéma (2.5 pts)



Q4 Programme de remise à zéro et insertion d'un nouvel votant en mémoire associative (1.5 pts).

Etiquette	Instruction	Commentaire	Barème
	Clé ← '0'	RAZ de la mémoire associative	0.5
	Masque ← '1'		
	Set		
	Ecriture		
	S/I ← '0'	Initialisation	0.5
	Reset		
	Clé ← '0'	Recherche du prochain emplacement libre	
	Masque <input type="checkbox"/> '1'		
	Set		
	Recherche		
	Si S/N=0 aller à FIN	Test si emplacement libre disponible	
	S/I ← '1'	Insertion d'un nouvel votant en mémoire associative	0.5
	Clé.NE ← 'Numéro électeur'		
	Clé.NC ← 'Numéro Candidat'		
	Clé.CE ← 'Code élection'		
	Masque ← '1' partout		
	Ecriture		
	BT ← '1'		
	S/I ← '0'		
Fin	Stop	Fin du programme	

Q5 Programme qui permet de donner le score de chaque candidat (4 pts).

1^{ère} approche

Etiquette	Instruction	Commentaire	Barème
	S/I ← '0'	Cpt _i : compteur pour chaque candidat, i varie de 1 à N	0.5
	Reset		
	Cpt _i		
Boucle	Set		2
	Si S/N= 0 aller à affichage	Test de fin de traitement des votants	
	S/I ← 1		
	Lecture		
	Si sortie.NC=i alors Cpt _i ←Cpt _i +1		
	BT ← '1'		
	S/I ← 0		
	Aller à Boucle		
Affichage	Pour i de 1 à N	N=nombre total de candidats	1.5
	<i>Faire</i>		
	Afficher (Numéro candidat_i, Cpt _i)		
	<i>Finfaire</i>		
	Stop		

2^{ème} approche

Etiquette	Instruction	Commentaire	Barème
	S/I ← '0'	Initialisation	0.5
	Reset		
	Nbc ← '0'	Nbc=Nombre de candidats traités	
Début	Si Nbc > N aller à Fin	N : nombre total de candidats de chaque élection	2
	Nombre ← 0	Nombre=Nombre de votants	
	Clé.NC ← 'Numéro candidat'	Traitement d'un candidat	
	Masque.NC ← '1', 0 ailleurs		
Boucle	Set		
	Recherche		
	Si S/N=0 aller à suivant		
	S/I ← '1'	Incrémentatation du nombre de votants pour chaque candidat	
	Lecture		
	Nombre ← Nombre+1		
	BT ← '1'		
	S/I ← '0'		
	Aller à boucle		
Suivant	Afficher ('Numéro Candidat', Nombre)	Affichage du candidat et de son score	1.5
	Nbc ← Nbc+1	Passage au candidat suivant	
	Aller à début		
Fin	Stop	Fin de programme	