

## Control Intermédiaire ARCHI 2 (Corrige)

## Exercice no. 2

On dispose de quatre types de véhicules:

- 0: véhicule léger (tourisme)
- 1: véhicule de petit tonnage (camionnette)
- 2: véhicule de moyen tonnage (camion)
- 3: véhicule de grand tonnage (semi-remorque)

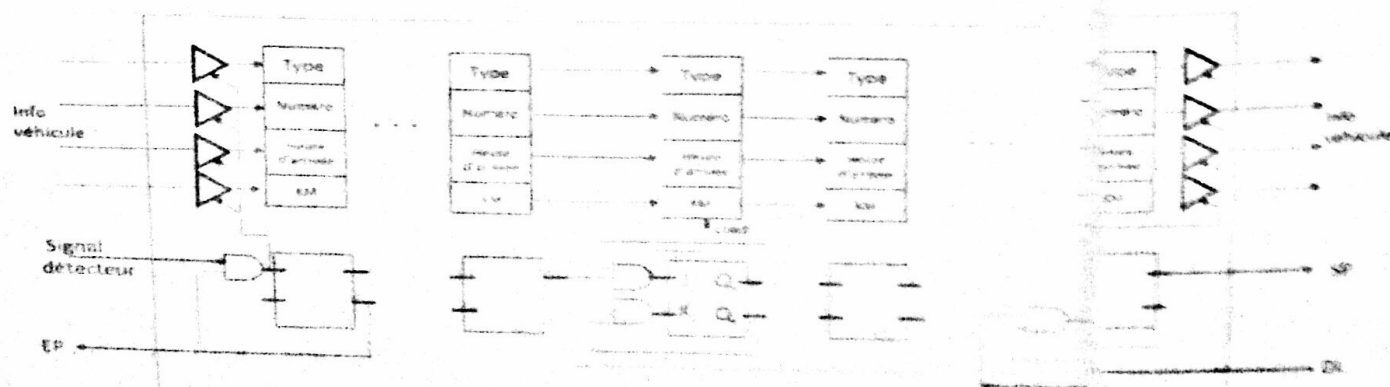
On dispose aussi de quatre parkings A, B, C, et D associés aux types de véhicules.

## PARTIE A

Lorsqu'un véhicule arrive, un détecteur génère automatiquement un signal pour ouvrir la barrière et les informations suivantes sont enregistrées dans une mémoire qui conserve l'ordre d'arrivée.

- Type du véhicule (T)
- Numéro du véhicule (No)
- Heure d'arrivée du véhicule (HA)
- Nombre de kilomètres parcourus par le véhicule (KM)

1. Le type de mémoire à utiliser pour enregistrer l'information des véhicules selon l'ordre d'arrivée est une file (FIFO) [0.5pt]
2. Schéma détaillé de la mémoire, du circuit de commande ainsi que les signaux qui indiquent si l'enregistrement est autorisé [2 pts]



$$J_1 = P_1 \cdot \bar{P}_1$$

$$K_1 = P_1 \cdot \bar{P}_1$$

$$J_0 = \text{Détecteur} \cdot \bar{V}_1$$

$$K_0 = Q_1 \cdot P_1 \cdot \bar{V}_1 \cdot \bar{d} = 1$$

$$EP = E \quad (\text{Entrée prête})$$

$$SP = P \quad (\text{Sortie prête})$$

## PARTIE B

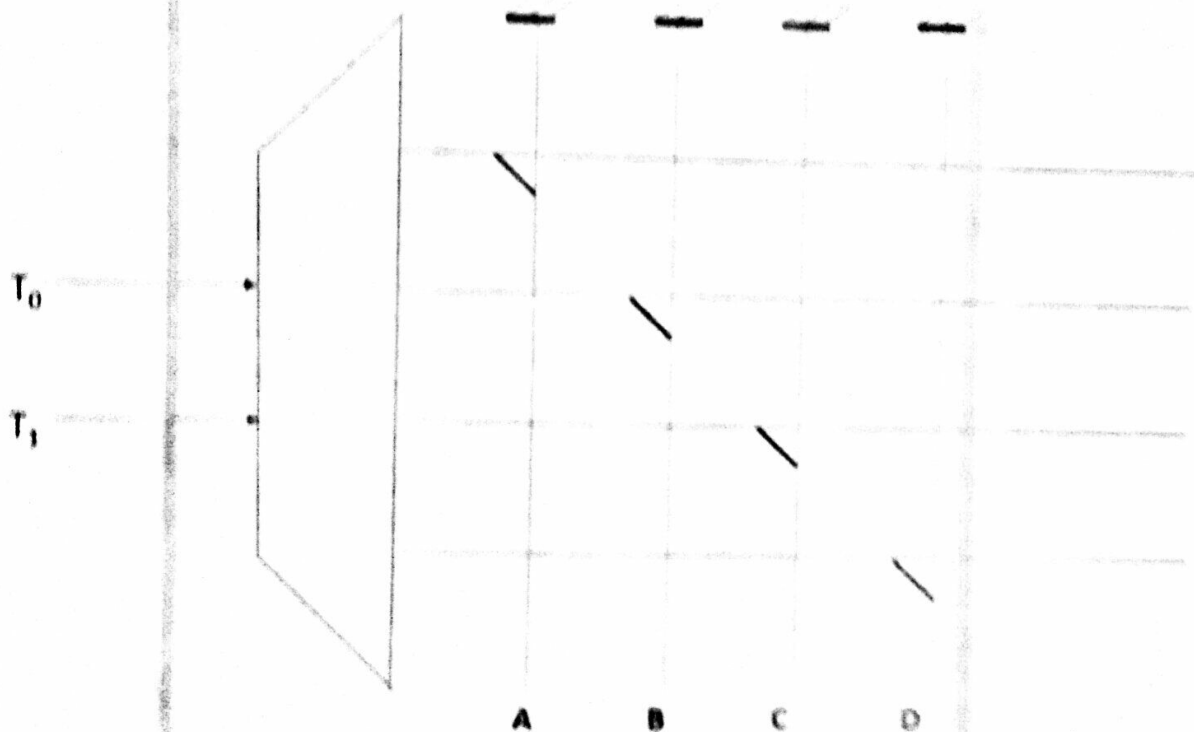
L'entrée à chaque parking est contrôlée par une barrière qui s'ouvre de manière automatique selon le type de véhicule.

On veut réaliser un circuit qui ouvre automatiquement la barrière du parking selon le type de véhicule. On utilisera une mémoire.

1. Le type de mémoire à utiliser pour réaliser ce circuit est une ROM dont la taille est comme suit :
  - A. 2 bits d'adresse en entrée pour le type de véhicule ce qui donne 4 mots.
  - B. Chaque mot est constitué de 4 bits (les 4 parkings).

Donc la ROM a 4 mots de 4 bits. **[0.5 pt pour le type et 0.5 pt pour la taille]**

2. Schéma de la ROM 4 x 4 **[1pt]**



## PARTIE C

Lorsqu'un véhicule rentre dans le parking qui lui est associé, les informations suivantes sont enregistrées dans une mémoire associative.

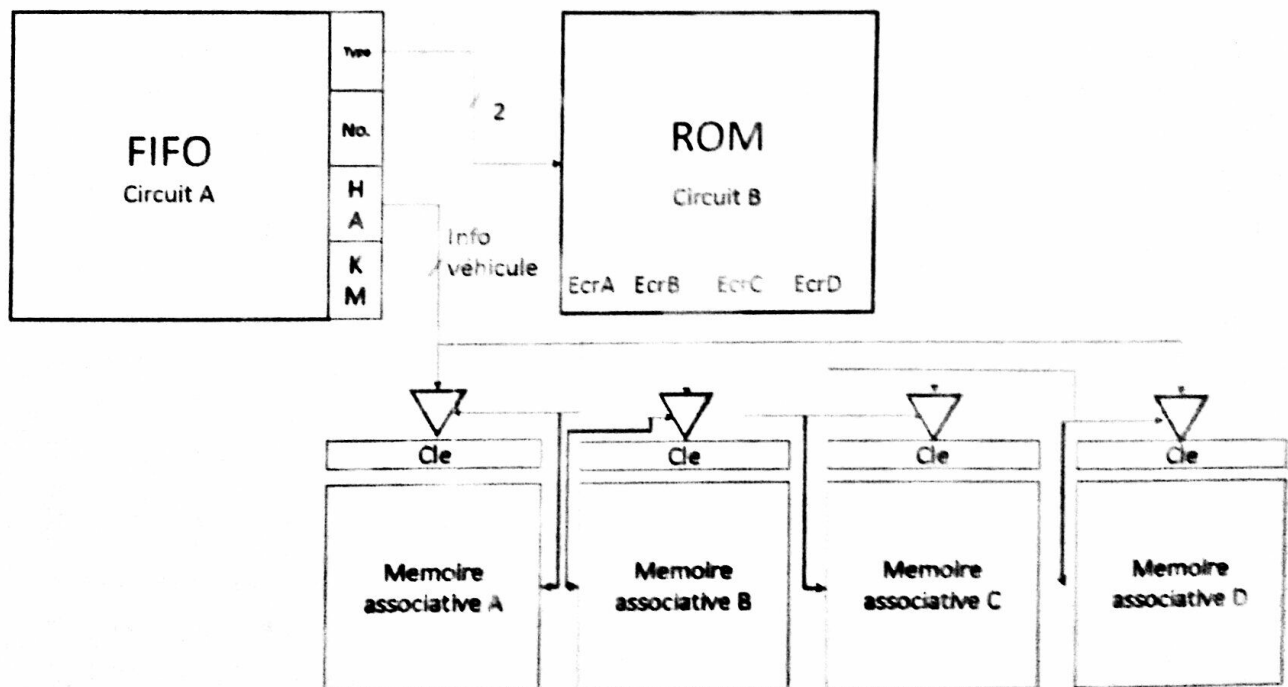
Numéro du véhicule (No.)

Heure d'arrivée (HA)

Nombre de kilomètres parcourus (KM)

Il existe une mémoire associative pour chaque parking.

1. Schéma global du circuit montrant les connexions avec le circuit de la partie A et de la partie B. [1.5 pt]



2. Le programme qui permet de lire les informations concernant le véhicule qui a parcouru le plus grand nombre de kilomètres pour un type de voiture donnée [2.5 pts]

ALGORITHME

RESET

SI = 0

M = 0 partout // registre masque

cpt = n // nombre de bits pour représenter le nombre de kilomètres

BCL Oie KM[cpt] = 1

M KM[cpt] = 1

Set

Recherche

Si (SI = 0) Aller à MAJ

DEC cpt = cpt - 1

Si (cpt = -1) Aller à LECT

Aller à BCL

MAJ Oie KM[cpt] = 0

Aller à DEC

LECT Set

Recherche

Lecture

Affecter RS N, RS HA, RS KM

STOP

END

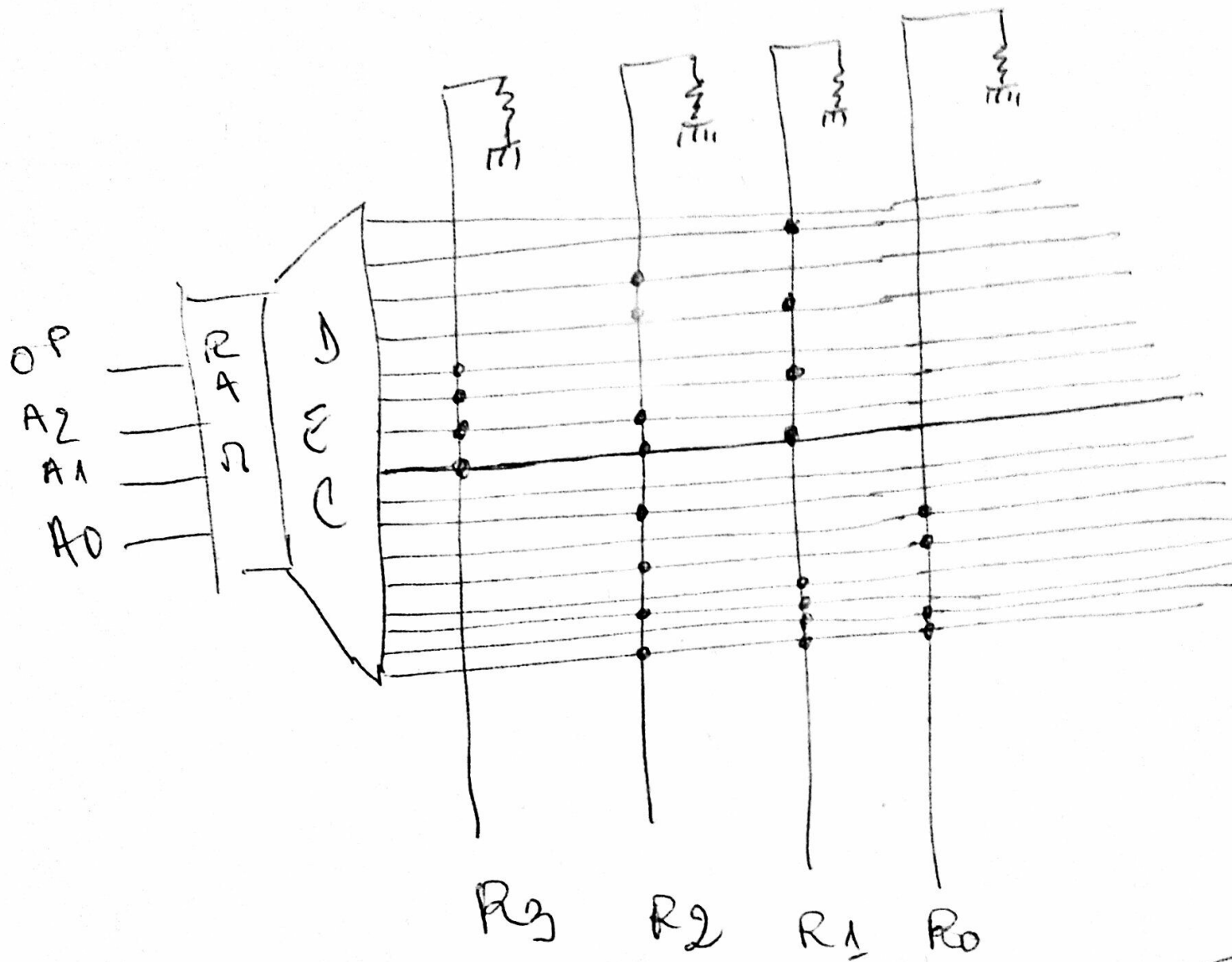
9.1 la Table de vérité

$A_3$	$A_2$	$A_1$	$A_0$	$R_3$	$R_2$	$R_1$	$R_0$
0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0
0	0	1	0	0	1	0	0
0	0	1	1	0	1	1	0
0	1	0	0	1	0	0	0
0	1	0	1	1	0	1	0
0	1	1	0	1	1	0	0
0	1	1	1	1	1	1	0
1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0
1	0	1	0	0	0	0	1
1	0	1	1	0	1	0	1
1	1	0	0	0	0	1	0
1	1	0	1	0	1	1	0
1	1	1	0	0	0	1	1
1	1	1	1	0	1	1	1

Pour la multiplication, le résultat est sur 4 bits ( $R_3, R_2, R_1, R_0$ ).

Pour la division, le quotient est sur 2 bits ( $R_1, R_0$ ) et la reste sur 2 bits.

## Q.2. Schema





### Exercice 4: (5 points)

Répondre directement sur la feuille jointe. Pour chaque question, cocher la ou les réponses correctes.

1.	<b>Technologie des mémoires</b> <b>Cochez la ou les affirmations correctes :</b> <input type="checkbox"/> l'écriture dans une mémoire dynamique prend deux fois plus de temps que la lecture; <input checked="" type="checkbox"/> une cellule de mémoire statique occupe plus d'espace qu'une cellule dynamique; <input type="checkbox"/> le rafraîchissement se fait uniquement sur les mots non modifiés; <input type="checkbox"/> les cellules des mémoires dynamiques sont plus rapides que les mémoires statiques.
2.	<b>Mémoires cache</b> <b>Dans les mémoires cache, la localité spatiale signifie :</b> <input type="checkbox"/> que la donnée et l'instruction utilisent la même adresse; <input checked="" type="checkbox"/> que si une information est utilisée à l'adresse i, alors la probabilité est élevée pour qu'elle le soit aussi à une adresse très proche ; <input type="checkbox"/> que les données utilisées souvent ne sont jamais transférées en mémoire cache; <input type="checkbox"/> que les données et les instructions sont stockées dans des mémoires différentes.
3.	<b>Mémoires cache</b> <b>Cochez la ou les affirmations correctes :</b> <input type="checkbox"/> les mémoires cache actuelles sont aussi grandes que les mémoires centrales ; <input type="checkbox"/> les caches internes sont toujours plus grandes que les caches externes ; <input type="checkbox"/> les caches de grandes tailles sont plus rapides que les petites caches ; <input checked="" type="checkbox"/> le remplacement aléatoire donne d'aussi bonnes performances que le LRU pour des mémoires de grandes tailles.
4.	<b>Mémoires mortes</b> <b>Cochez la ou les affirmations correctes :</b> <input type="checkbox"/> les PROM sont réalisées à base de diodes uniquement; <input type="checkbox"/> les EEPROM peuvent remplacer les mémoires centrales ; <input checked="" type="checkbox"/> les EPROM sont effaçables électriquement ; <input type="checkbox"/> l'écriture dans les EPROM se fait en exposant la mémoire à des rayons ultraviolets.
5.	<b>Organisation des Mémoires</b> <b>Cochez la ou les affirmations correctes :</b> <input type="checkbox"/> il ne peut jamais y avoir de conflit d'accès dans des mémoires modulaires ; <input type="checkbox"/> les mémoires entrelacées permettent de réaliser des mémoires plus grandes que les mémoires modulaires; <input checked="" type="checkbox"/> les mémoires entrelacées sont utiles pour des architectures avec plusieurs processeurs ; <input type="checkbox"/> les mémoires modulaires sont plus rapides que les mémoires en un seul bloc.