

20-06-2022

## Examen de Rattrapage

Durée 1h30mn

### Exercice 1 : (7 Points)

On veut réaliser un compteur  $C$  à 8 états (0, 1, 2, 3, ..., 6, 7) tel que :

État Suivant = État Présent + 3 si État Présent < 5

État Suivant = État Présent - 5 si État Présent  $\geq$  5

- 1- Représenter la séquence correspondant à ce compteur.
- 2- Établir la table d'excitation de ce compteur en utilisant des bascules  $T$ .
- 3- Donner les équations des entrées  $T$  des bascules.
- 4- Réaliser le circuit.
- 5- On veut ajouter une fonction de décalage circulaire à gauche à ce circuit, donner les équations générales du circuit qui remplit les 2 fonctions précédentes.

### Exercice 2 : (7 Points)

On considère une machine avec la configuration suivante :

- Mémoire centrale de taille 1Moctets.
- Mot mémoire de taille 2 octets.
- Bus d'adresse (ou registre adresse) de taille 20 bits.

- 1- Calculer la taille minimale du bus d'adresse qui permet d'accéder à cette mémoire.
- 2- Déterminer la plage d'adressage de cette mémoire (adresse minimale et adresse maximale).
- 3- Quelle est la taille maximale que peut avoir la mémoire centrale sur cette machine
- 4- On veut reconfigurer la mémoire de cette machine de la manière suivante :
  - On dispose d'un bloc de mémoire vive RAM, de taille 512Koctets adressable à partir de l'adresse (00000)<sub>16</sub>.
  - On dispose aussi de deux blocs de ROM de taille 256Koctets, le premier est adressable à partir de l'adresse (60000)<sub>16</sub> et le deuxième à partir de l'adresse (C0000)<sub>16</sub>.
    - a. Déterminer les plages d'adressage des différents blocs de RAM et ROMs ? Y a-t-il de l'espace non utilisé. Si oui donner les plages d'adresses de l'espace non utilisé.
    - b. Donner le schéma global de cette mémoire.
    - c. Est-ce que la mémoire de cette machine est extensible ? Si oui, combien de blocs RAM et ROM peut-on rajouter ?

**Exercice 3: (6 Points)**

Soit le programme assembleur suivant commençant à l'adresse 128.

- 1- Dérouler ce programme en donnant le contenu de l'Acc et des registres CO, MAR, MBR et RI après l'exécution de chaque instruction.
- 2- Donner les valeurs des variables Y et P à la fin de l'exécution sachant que la valeur **p** lue au périphérique d'entrée est 10.

(@Mémoire)	Contenu
1	10
---	
10	2
---	
16	1
---	
128	READ
129	STORE P, D
130	MUL 1, D
131	ADD 16, D
132	STORE Y, D
133	MUL 1, IND
134	STORE P, D
135	LOAD Y, D
136	MUL 10, IMM
137	ADD 255, D
138	STORE Y, D
139	MUL 3, IMM
140	STORE P, D
141	WRITE
---	
255	16
Y	
P	

*Bonne Chance*

## Corrigé Proposé

### Exercice 1 : 7 Pts

#### 1-Séquence du compteur : 1 Pts

0 - 3 - 6 - 1 - 4 - 7 - 2 - 5 - 0...

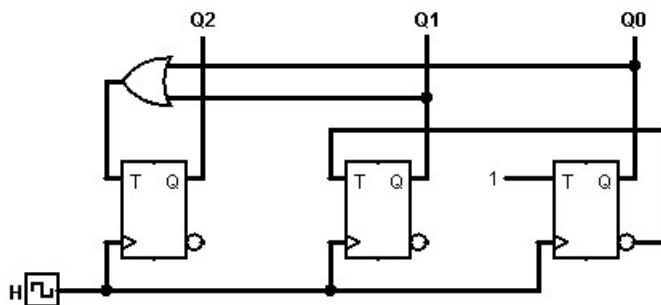
#### 2-Table d'excitation : 1.5 Pts

$Q_2$	$Q_1$	$Q_0$	$Q_2^+$	$Q_1^+$	$Q_0^+$	$T_2$	$T_1$	$T_0$
0	0	0	0	1	1	0	1	1
0	0	1	1	0	0	1	0	1
0	1	0	1	0	1	1	1	1
0	1	1	1	1	0	1	0	1
1	0	0	1	1	1	0	1	1
1	0	1	0	0	0	1	0	1
1	1	0	0	0	1	1	1	1
1	1	1	0	1	0	1	0	1

#### 3- Equations des entrées des bascules T : 1.5 Pts

$$T_0 = 1, \quad T_1 = \overline{Q_0}, \quad T_2 = Q_1 + Q_0$$

#### 4- Circuit : 1 Pts



#### 5- Equations générales pour assurer les deux fonctions : 2 Pts

Pour que le circuit assure deux fonctions, on doit ajouter une variable de contrôle. Soit X cette variable.

Pour  $X=0$ , le circuit assure la première fonction (séquence précédente). Donc les équations des  $T_i$  seront multiplier par  $\overline{X}$ .

Pour  $X=1$ , le circuit assure la fonction de décalage circulaire à gauche. Dans ce cas, on doit avoir :

$$Q_2 = Q_1, \quad Q_1 = Q_0, \quad Q_0 = Q_2$$

Mise à Q de T

$Q_i$	$Q_j$	$Q_j^+ = Q_i$	$T_j$
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

D'après cette table, pour mettre  $Q_j$  à  $Q_i$  on doit avoir :

$$T_j = Q_j \oplus Q_i$$

Donc  $T_2 = Q_2 \oplus Q_1$  avec  $X=1$ , on trouve :  $T_2 = X Q_2 \oplus Q_1$

De la même manière, on trouve :  $T_1 = X Q_1 \oplus Q_0$  ,  $T_0 = X Q_2 \oplus Q_0$

D'où les équations générales sont :

$$T_2 = X Q_2 \oplus Q_1 + \bar{X}(Q_1 + Q_0) \quad , \quad T_1 = X Q_1 \oplus Q_0 + \bar{X} \bar{Q}_0 \quad , \quad T_0 = X Q_2 \oplus Q_0 + \bar{X}$$

### **Exercice 2 : 7 Pts**

#### **1- Taille minimale du bus d'adresses : 0.5 Pts**

Soit N la taille minimale du bus d'adresses.

On a  $C = m \times n = 2^N \times n$

Pour cette machine,  $C = 1\text{Mo} = 2^{20} \times 8$  et  $n = 16$ .

Donc  $C = 2^{19} \times (16) \Rightarrow N = 19 \text{ bits}$ .

#### **2- Plages d'adresses (@min, @max) : 0.5 Pts**

Le bus d'adresses de la machine étant de 20 bits, donc :

$$@\min = (00000)\text{H} \quad \text{et} \quad @\max = (\text{FFFFF})\text{H}$$

#### **3- Taille maximale de la mémoire : 0.5 Pts**

$$\text{On a : } C = m \times n = 2^N \times n = 2^{20} \times 16 = 2 \times 2^{20} \times 8 = 2 \text{ Mo}$$

#### **4- a- Plages d'adresses des différents blocs : 1.5 Pts**

##### **Bloc RAM :**

$$C_{\text{RAM}} = 512 \text{ Ko} = 2^9 \times 2^{10} \times 8 = 2^{18} \times 16, \text{ donc elle est contenue } 2^{18} \text{ mots.}$$

$$\text{On a : } \text{NbMots} = @Fin - @Debut + 1 \Rightarrow @Fin = \text{NbMots} + @Debut - 1 = 2^{18} + 0 - 1.$$

$$\text{Donc } @Fin = (3FFFF)\text{H}$$

D'où la plage de la RAM est :  $(00000)\text{H} - (3FFFF)\text{H}$

##### **Bloc ROM1:**

$$C_{\text{ROM}} = 256 \text{ Ko} = 2^8 \times 2^{10} \times 8 = 2^{17} \times 16, \text{ donc elle est contenue } 2^{17} \text{ mots.}$$

$$\text{On a : } \text{NbMots} = @Fin - @Debut + 1 \Rightarrow @Fin = \text{NbMots} + @Debut - 1 = 2^{17} - 1 + (60000)\text{H}.$$

Donc @Fin = (1FFFF)H + (60000)H = (7FFFF)H

D'où la plage de la ROM1 est : (60000)H - (7FFFF)H

### Bloc ROM2:

$C_{ROM} = 256 \text{ Ko} = 2^8 \times 2^{10} \times 8 = 2^{17} \times 16$ , donc elle est contient  $2^{17}$  mots.

On a : NbMots = @Fin - @Debut + 1  $\Rightarrow$  @Fin = NbMots + @Debut - 1 =  $2^{17} - 1 + (C0000)H$ .

Donc @Fin = (1FFFF)H + (C0000)H = (DFFFF)H

D'où la plage de la ROM2 est : (C0000)H - (DFFFF)H

D'après les plages d'adresses des différents blocs utilisés, on remarque que les plages d'adresses ne sont pas consécutives, donc il y a des espaces libres non utilisés. **0.5 Pts**

Pour trouver les plages d'adresses des espaces non utilisés on utilise la table des adresses suivante :

A19	A18	A17	A16-----A0	Circuits	Plages des @
0	0	0		RAM : 512 Ko	(00000)H – (1FFFF)H
0	0	1			(20000)H – (3FFFF)H
0	1	0	Libre		(40000)H – (5FFFF)H
0	1	1		ROM1 : 256 Ko	(60000)H – (7FFFF)H
1	0	0	Libre		(80000)H – (9FFFF)H
1	0	1			(A0000)H – (BFFFF)H
1	1	0		ROM2 : 256 Ko	(C0000)H – (DFFFF)H
1	1	1	Libre		(E0000)H – (FFFFFF)H

D'après cette table on trouve trois espaces libres non utilisés dont les plages d'adresses sont :

Espace 1 (256 Ko): (40000)H – (5FFFF)H

Espace 2 (512 Ko): (80000)H – (BFFFF)H

Espace 3 (256 Ko): (E0000)H – (FFFFFF)H **1 Pts**

**Remarque :** afin d'éviter les opérations en hexa pour le calcul des adresses des RAM et ROM on peut utiliser directement ce tableau

### 4- b- Schéma de la mémoire : 1.5 Pts

Pour tracer le schéma de la mémoire on doit trouver les CS de chaque bloc.

D'après l'étude précédente, on a vu que le bloc RAM est adressable par 18 bits (A17..A0) et les blocs ROM sont adressable par 17 bits (A16..A0), donc deux bits (A19, A18) sont utilisés pour le CS de la ROM et trois bits (A19,A18,A17) sont utilisés pour les CS des ROM.

D'après la table des adresses précédente, on trouve :

$$CS_{RAM} = \overline{A19} \overline{A18}$$

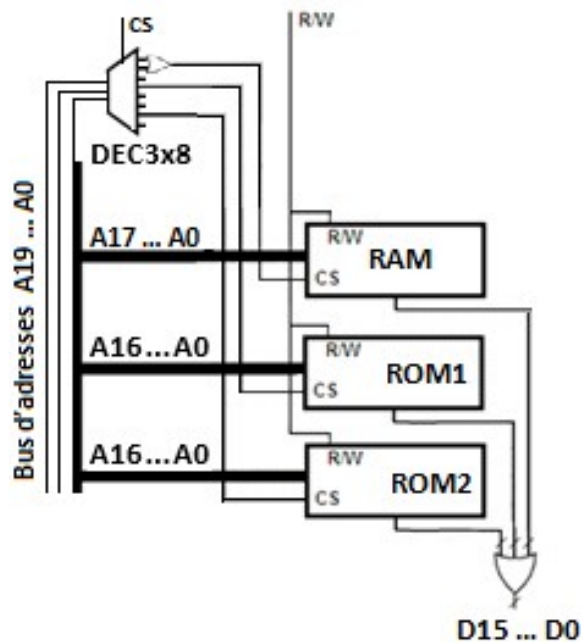
$$CS_{ROM1} = \overline{A19} A18 A17$$

$$CS_{ROM2} = A19 A18 \overline{A17}$$

D'où le schéma global suivant :

Pour les CSi, on peut utiliser un décodeur 3x8 avec :

$$CS_{RAM} = D0+D1, CS_{ROM1} = D3 \text{ et } CS_{ROM2} = D6.$$



#### 4-c- Nombre de blocs RAM et ROM qu'on peut rajouter : 1 Pts

Comme il y a des espaces non utilisés, la mémoire est extensible. D'après les plages d'adresses des espaces non utilisés, on a deux espaces de taille 256 Ko et un espace de 512 Ko, donc on peut rajouter **un bloc RAM et deux blocs ROM**.

#### Exercice 3 : 6 Pts

##### 1- Déroulement du programme : 5 Pts

CO	MAR	MBR	RI	ACC	P	Y
129	128	READ	READ	p		
130	P	p	STORE P, D	p	p	
131	1	10	MUL 1, D	10p	p	
132	16	1	ADD 16, D	10p+1	p	
133	Y	10p+1	STORE Y, D	10p+1	p	10p+1
134	10	2	MUL 1, IND	2(10p+1)	p	10p+1
135	P	2(10p+1)	STORE P, D	2(10p+1)	2(10p+1)	10p+1
136	Y	10p+1	LOAD Y, D	10p+1	2(10p+1)	10p+1
137	136	MUL 10, IMM	MUL 10, IMM	10(10p+1)	2(10p+1)	10p+1
138	255	16	ADD 255, D	10(10p+1)+16	2(10p+1)	10p+1
139	Y	10(10p+1)+16	STORE Y, D	10(10p+1)+16	2(10p+1)	10(10p+1)+16
140	139	MUL 3, IMM	MUL 3, IMM	3(10(10p+1)+16)	2(10p+1)	10(10p+1)+16
141	P	3(10(10p+1)+16)	STORE P, D	3(10(10p+1)+16)	3(10(10p+1)+16)	10(10p+1)+16
142	141	WRITE	WRITE	3(10(10p+1)+16)	3(10(10p+1)+16)	10(10p+1)+16

##### 2- Valeur de Y et P pour p=10 : 1 Pts

On a :

$$(P) = 3(10(10p+1)+16) = 3(10(10*10+1)+16) = 3(10(101)+16) = 3(1010+16) = 3*1026 = \mathbf{3078}$$

$$(Y) = 3(10(10p+1)+16) = 10(10*10+1)+16 = 10(101)+16 = 1010+16 = \mathbf{1026}$$