

2°) Contenu de la PROM

A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Sorties				D <sub>3</sub>	Etat suivant		
			D <sub>7</sub> (Q <sub>6</sub> )	D <sub>6</sub> (Q <sub>5</sub> )	D <sub>5</sub> (Q <sub>4</sub> )	D <sub>4</sub> (Q <sub>3</sub> )		D <sub>2</sub> (Q <sub>2</sub> )	D <sub>1</sub> (Q <sub>1</sub> )	D <sub>0</sub> (Q <sub>0</sub> )
0	0	0	0	0	0	1		0	0	1
0	0	1	0	1	0	1		0	1	0
0	1	0	0	1	1	0		0	1	1
0	1	1	0	1	0	0		1	0	0
1	0	0	1	0	0	1		1	0	1
1	0	1	0	0	1	0		1	1	0
1	1	0	0	0	0	0		0	0	0

**EXERCICE 15**

1°) Le nombre de séquences d'animation possibles est fonction du code d'animation stocké dans le registre d'entrée. Or, ce registre possède 3 entrées. Le nombre total de codes d'animation est donc égale à  $2^3 = 8$  codes d'animation.

La plage d'adresse de chaque séquence peut se déterminer de la façon suivante :

- Les bits A<sub>4</sub> à A<sub>6</sub> donnent le code de chacune des 8 séquences.
- Les bits A<sub>0</sub> à A<sub>3</sub> codent les 16 pas de chaque séquence.

On en tire donc les plages d'adresse des 8 séquences.

Code séquence			Code pas				Séquence et adresse en hex
A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	
0	0	0	0	0	0	0	Séquence 0 00 - 0F
0	0	1	0	0	0	0	Séquence 1 10 - 1F
0	1	0	0	0	0	0	Séquence 2 20 - 2F
0	1	1	0	0	0	0	Séquence 3 30 - 3F
1	0	0	0	0	0	0	Séquence 4 40 - 4F
1	0	1	0	0	0	0	Séquence 5 50 - 5F
1	1	0	0	0	0	0	Séquence 6 60 - 6F
1	1	1	0	0	0	0	Séquence 7 70 - 7F

2°) Contenance de la PROM pour la séquence 0.

La génération d'une séquence passe par les étapes suivantes :

- chargement du code de la séquence dans le registre d'entrée
- génération des sorties
- remise à zéro du compteur à la fin de la séquence.

Chaque cas mémoire de la PROM doit donc contenir un mot divisé en 3 parties comprenant :

- l'état des lampes
- la commande de l'horloge du registre d'entrée
- la commande de la remise à zéro du compteur

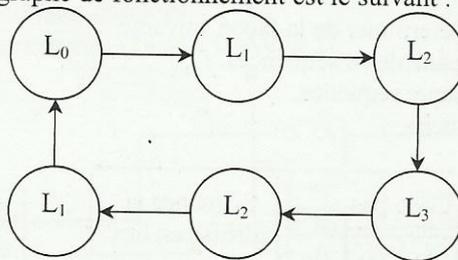
Le format du mot est le suivant :

Sorties				Bits non utilisés		Horloge registre entrée	Remise à zéro du compteur
D <sub>7</sub> (Q <sub>7</sub> )	D <sub>6</sub> (Q <sub>6</sub> )	D <sub>5</sub> (Q <sub>5</sub> )	D <sub>4</sub> (Q <sub>4</sub> )	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub> (Q <sub>1</sub> )	D <sub>0</sub> (Q <sub>0</sub> )
L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>0</sub>			CH_ENTREE	RAZ

Le chargement du code d'entrées s'effectue en générant un front montant sur CH\_ENTREE. Pour cela, dans **la première case** mémoire à l'adresse 00 de la séquence 0, on mettra :

- les sorties D4 à D7 à 0
- CH\_ENTREE à 0
- RAZ à 0

Dans la deuxième case, on mettra CH\_ENTREE à 1 pour générer un front montant. Les suivantes vont générer le fonctionnement du chenillard qui consiste à allumer successivement les lampes de gauche à droite et de droite à gauche. La graphe de fonctionnement est le suivant :



A la fin de séquence, il doit y avoir une remise à zéro du compteur (RAZ = 1)

**Contenu de la PROM pour la séquence 0** (fonctionnement du chenillard)

Adresse	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	
00 H	0	0	0	0	0	0	0	0	Chargement du code d'entrée
01 H	0	0	0	0	0	0	1	0	
02 H	0	0	0	1	0	0	0	0	Génération des sorties du chenillard
03 H	0	0	1	0	0	0	0	0	
04 H	0	1	0	0	0	0	0	0	
05 H	1	0	0	0	0	0	0	0	
06 H	0	1	0	0	0	0	0	0	
07 H	0	0	1	0	0	0	0	0	
08 H	0	0	0	1	0	0	0	0	
09 H	0	0	0	0	0	0	0	1	
0A H	0	0	0	0	0	0	0	0	Zone non utilisée
0B H	0	0	0	0	0	0	0	0	
0C H	0	0	0	0	0	0	0	0	
0D H	0	0	0	0	0	0	0	0	
0E H	0	0	0	0	0	0	0	0	
0F H	0	0	0	0	0	0	0	0	

On peut se limiter à un seul cycle de fonctionnement car la zone mémoire non utilisée ne permet pas d'avoir plus d'un cycle complet.

**Contenu de la PROM pour la séquence 1** (allumage alternative des lampes paires et impaires)

Sur 16 cases mémoires réservées à chaque séquence, après avoir utilisé 2 cases pour le chargement du code d'entrée et 1 case pour la remise à zéro du compteur, il reste 13 cases. Un cycle de clignotement prend 2 cases mémoires. On aura donc au maximum 6 cycles possibles.

Adresse	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	
10 H	0	0	0	0	0	0	0	0	Chargement du code d'entrée
11 H	0	0	0	0	0	0	1	0	
12 H	0	1	0	1	0	0	0	0	Génération des sorties du chenillard
13 H	1	0	1	0	0	0	0	0	
14 H	0	1	0	1	0	0	0	0	
15 H	1	0	1	0	0	0	0	0	
16 H	0	1	0	1	0	0	0	0	
17 H	1	0	1	0	0	0	0	0	
18 H	0	1	0	1	0	0	0	0	
19 H	1	0	1	0	0	0	0	0	
1A H	0	1	0	1	0	0	0	0	
1B H	1	0	1	0	0	0	0	0	
1C H	0	1	0	1	0	0	0	0	
1D H	1	0	1	0	0	0	0	0	
1E H	0	0	0	0	0	0	0	1	
1F H	0	0	0	0	0	0	0	0	

**EXERCICE 16**

1°) Capacité de l'UV PROM

La capacité d'une mémoire peut être calculée avec la formule suivante :  $C_{ap} = 2n \times D$  (avec n le nombre de bits d'adresse et D le nombre de bits de données).

$$\left. \begin{array}{l} n = 4 (A_{12} \text{ à } A_{15}) \\ D = 8 (D_0 \text{ à } D_7) \end{array} \right\} \Rightarrow C_{ap} = 2^4 \times 8 = 16 \text{ octets.}$$

**Capacité de l'UVPROM = 16 octets**

2°) Capacité des éléments mémoire.

**ROM\_1 et ROM\_2 :**

- bus d'adresse : 12 bits ( $A_0$  à  $A_{11}$ )
- bus de données : 8 bits ( $D_0$  à  $D_7$ )

Cap =  $2^{12} \times 8 \text{ bits} = 2^2 \times 2^{10} \times 8 \text{ bits} = 4 \text{ Ko} \Rightarrow$  **Capacité ROM\_1 = Capacité ROM\_2 = 4 Ko**

**Ram\_1, Ram\_2, Ram\_3, Ram\_4, Ram\_5 et Ram\_6**

- bus d'adresse: 12 bits ( $A_0$  à  $A_{11}$ )
- bus de données: 4 bits ( $D_0$  à  $D_3$ )

Cap =  $2^{12} \times 4 \text{ bits} = 2^2 \times 2^{10} \times 4 \text{ bits} = 4\text{K} \times 4 = 2 \text{ Ko}$

Ram\_1 = Ram\_2 = Ram\_3 = Ram\_4 = Ram\_5 = Ram\_6 = 2 Ko

3°) Adresses des éléments mémoires

Adresse UVPROM	Adresse venant sur les mémoires	Adresses éléments mémoires	Elément sélectionné
$A_{12}$ à $A_{15}$	$A_0$ à $A_{11}$	$A_0$ à $A_{15}$	
0 h à 1 h	000 h à FFF h	<b>0000 – 1FFF</b>	Rom_1
2 h à 5 h	000 h à FFF h	<b>2000 – 5FFF</b>	Ram_1 – Ram_2
6 h à 9 h	000 h à FFF h	<b>6000 – 9FFF</b>	Ram_3 – Ram_4
A h à D h	000 h à FFF h	<b>A000 – DFFF</b>	Ram_5 – Ram_6
E h à F h	000 h à FFF h	<b>E000 – FFFF</b>	Rom_2

4°) Contenu de l'UVPROM

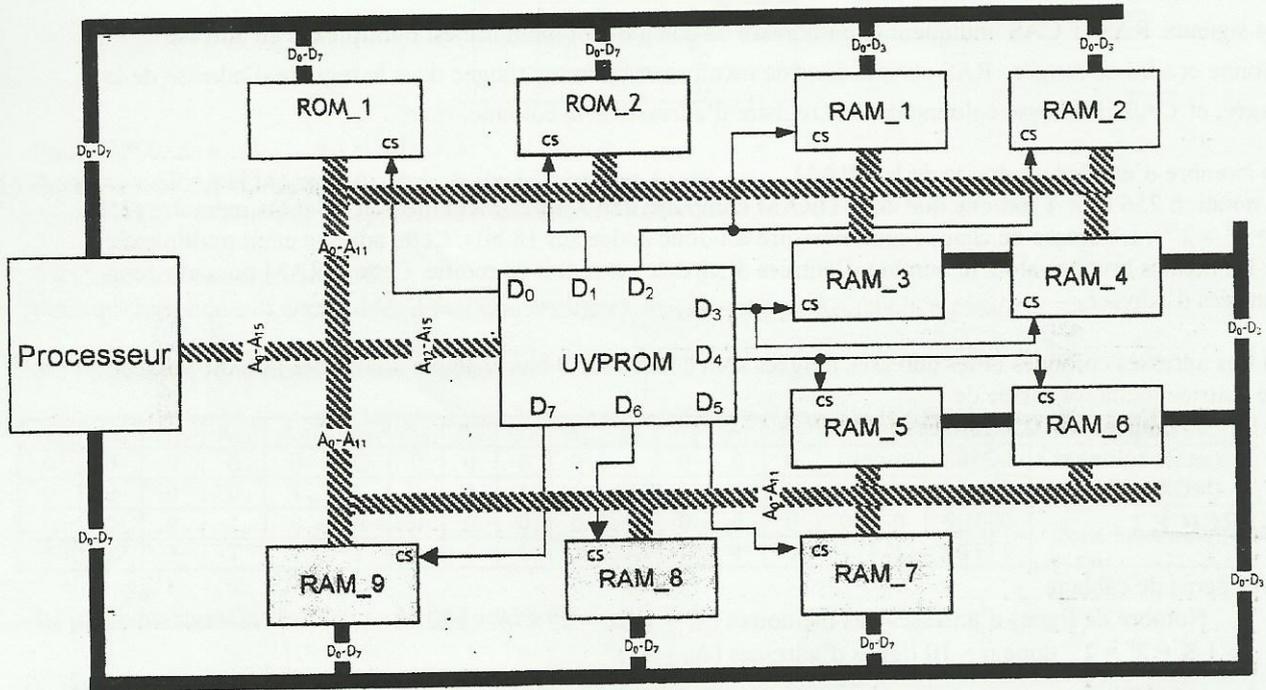
Adresse UVPROM	Adresses Processeur	Contenu de la PROM								Elément adressé
		D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	
0 h à 1 h	0000 – 1FFF	0	0	0	0	0	0	0	1	Rom_1
2 h à 5 h	2000 – 5FFF	0	0	0	0	1	1	0	0	Ram_1 – Ram_2
6 h à 9 h	6000 – 9FFF	0	0	1	1	0	0	0	0	Ram_3 – Ram_4
A h à D h	A000 – DFFF	1	1	0	0	0	0	0	0	Ram_5 – Ram_6
E h à F h	E000 – FFFF	0	0	0	0	0	0	1	0	Rom_2

5°) Modification du circuit

Pour connecter 3 autres circuits mémoires, il faut libérer 3 sorties de l'UVPROM. Pour ce faire, on va connecter :

- Ram\_1 et Ram\_2 sur la sortie D2
- Ram\_3 et Ram\_4 sur la sortie D3
- Ram\_5 et Ram\_6 sur la sortie D4

Les sorties D5, D6 et D7 sont ainsi libérées pour recevoir les 3 circuits Ram supplémentaires que nous appellerons Ram\_7, Ram\_8 et Ram\_9.



**ADRESSAGE MEMOIRES DANS LES SYSTEMES A MICROPROCESSEUR**

**EXERCICE 1**

1°) Fonctions des signaux  $\overline{\text{RAS}}$  et  $\overline{\text{CAS}}$

Les signaux  $\overline{\text{RAS}}$  et  $\overline{\text{CAS}}$  indiquent que l'adresse de chaque mot mémoire est multiplexée en adresse colonne et adresse rangée.  $\overline{\text{RAS}}$  permet donc de mémoriser l'adresse rangée dans le registre d'adresse de la rangée, et  $\overline{\text{CAS}}$ , l'adresse colonne dans le registre d'adresse de la colonne.

2°) Nombre d'entrées d'adresse de la DRAM

La notation  $256 \text{ K} \times 1$  indique que cette DRAM comporte 256 K mots mémoire soit  $2^{18}$  mots mémoire ( $256 \text{ K} = 2^8 \times 2^{10}$ ). L'adresse de chaque mot mémoire est donc codée sur 18 bits. Cette adresse étant multiplexée sur les mêmes broches alors le nombre d'entrées d'adresse est réduit de moitié. Cette DRAM possède donc 9 entrées d'adresse.

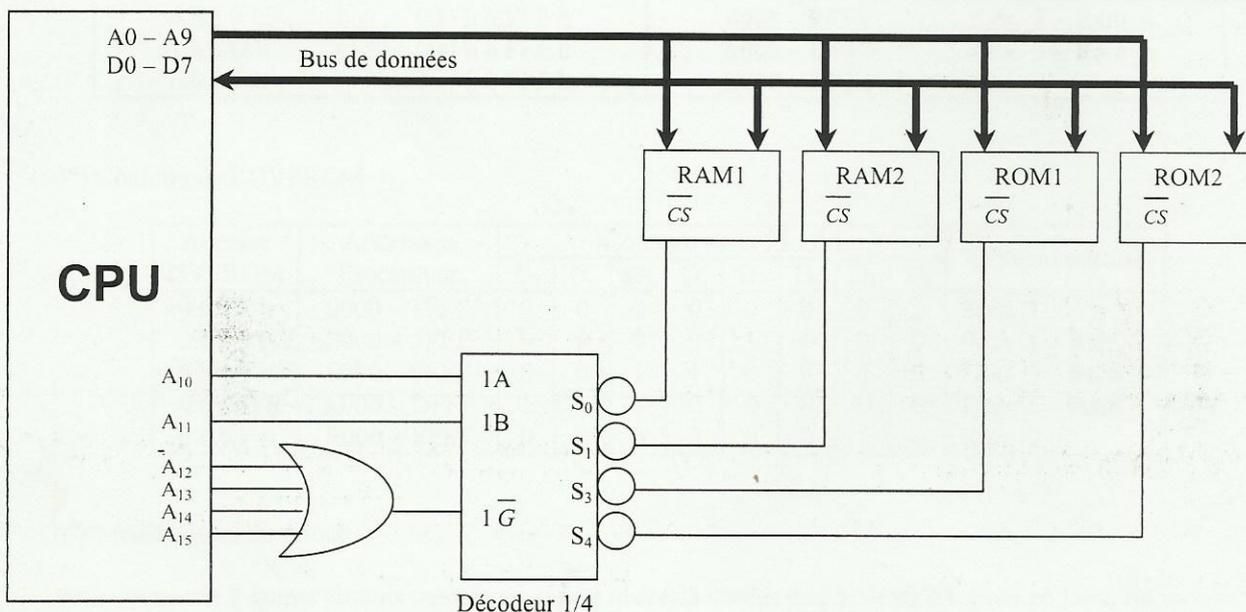
3°) Les adresses colonnes et les adresses rangées sont codées sur 9 bits chacune donc cette DRAM possède une matrice mémoire carrée de :

- $2^9$  rangées soit 256 rangées ;
- et  $2^9$  colonnes soit 256 colonnes.

**EXERCICE 2**

1°) Schéma de câblage

- Nombre de lignes d'adresses des mémoires :  $C = 1 \text{ Ko} = 2^n \times 8 = 1 \text{ K} \times 8$
- $\Rightarrow 1 \text{ K} = 2^n = 2^{10}$  donc  $n = 10$  lignes d'adresses ( $A_0 - A_9$ )



**Remarque :**

Pour que les lignes  $A_{12}$  à  $A_{15}$  soient forcées à zéro et permettent de sélectionner le décodeur, on doit utiliser une porte "OU". La porte "OU" force tous les bits à zéro (0) pour que la sortie soit à zéro (0) pour valider le décodeur par  $1\overline{G}$ .

Dès qu'une ligne et une seule est à un logique (1), alors la sortie de la porte "OU" est à un (1) et le décodeur est invalide ; donc aucune mémoire adressable.

2°) L'adresse \$0F00

Les lignes  $A_{10}$  et  $A_{11}$  sélectionnent la sortie S du décodeur tel que :

$A_{11}$	$A_{10}$	Sortie
0	0	$S_0$
0	1	$S_1$
1	0	$S_2$
1	1	$S_3$

Pour \$0F00, on a :  $A_{15} - A_{12} = S_0$

$A_{11} - A_8 = S_1 = (1111)_2$  soit  $A_{11} = 1, A_{10} = 1, A_9 = 1$  et  $A_8 = 1$ .

$A_{11} = 1, A_{10} = 1$  donc \$0F00 appartient à la mémoire ROM 2.

3°) Condition de zone accessible

Pour qu'une zone soit accessible, il faut que les lignes  $A_{12}, A_{13}, A_{14}$  et  $A_{15}$  soient à zéro.

4°) Plage de la zone mémoire inaccessible

$A_{15}$	$A_{14}$	$A_{13}$	$A_{12}$	$A_{11}$	$A_{10}$	$A_9$	$A_8$	$A_7$	$A_6$	$A_5$	$A_4$	$A_3$	$A_2$	$A_1$	$A_0$	Plage	Boîtier
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	\$ 0000	Mémoire centrale (RAM + ROM)
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	\$ 0FFF	
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	\$ 1000	Zone Inaccessible
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	\$ FFFF	

La plage inaccessible est donc : **\$ 1000 - \$ FFFF**

### EXERCICE 3

1°) Plage d'adresse des Ram\_1 et Ram\_2.

#### Sélection de la Ram\_1

Pour le décodeur 1, il faut :

$A_{15} = 1, A_{14} = 1, A_{13} = 0, A_{12} = 0, A_{11} = 0$

Pour le décodeur 2, il faut :

$A_{13} = 0, A_{12} = 0, A_{11} = 0$

La Ram\_1 de 2 Ko aura besoin de 11 fils d'adresse pour être adressée ( $A_0 - A_{10}$ ). Ces bits vont varier entre 0 et 1.

$A_{15}$	$A_{14}$	$A_{13}$	$A_{12}$	$A_{11}$	$A_{10}$	.....	$A_0$
					0	.....	0
					1	.....	1

L'adresse de la Ram\_1 est : **C000 – C7FF**

#### Sélection de la Ram\_2

Pour le décodeur 1, il faut :

$A_{15} = 1, A_{14} = 1, A_{13} = 1, A_{12} = 1, A_{11} = 1$

La Ram\_2 de 2 Ko aura besoin de 11 fils d'adresse pour être adressée ( $A_0 - A_{10}$ ). Ces bits vont varier entre 0 et 1.

$A_{15}$	$A_{14}$	$A_{13}$	$A_{12}$	$A_{11}$	$A_{10}$	.....	$A_0$
					0	.....	0
					1	.....	1

L'adresse de la Ram\_2 est : **F800 – FFFF**

2°) Taille et plage d'adresse du circuit X.

Pour valiser le boîtier X, il faut :

Pour le décodeur 1

$A_{15} = 1$   
 $A_{14} = 1$   
 $A_{13} = 1$   
 $A_{12} = 0$   
 $A_{11} = 1$

pour le décodeur 1

$A_{10} = 1$   
 $A_9 = 0$   
 $A_8 = 1$   
 $A_7 = 1$

Les bits d'adresse restants peuvent venir sur le circuit X dont il faut déterminer la taille. Ces bits sont A0 à A6, soit 7 bits. Le circuit X aura donc comme taille  $2^7 = 128$  octets.

A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>
1	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0
									1	1	1	1	1	1	1

La plage d'adresse du circuit X est la suivante : **ED80 – EDFF**

**EXERCICE 4**

1°) La structure d'E/S utilisé est la structure E/S par instruction mémoire. En effet, le processeur ne fait aucune distinction entre E/S et mémoire. Cela se voit par l'absence dans le circuit du signal  $\overline{\text{MEM}/\text{IO}}$ .

2°) On veut connaître les boîtiers et les octets auxquels les adresses suivantes font référence. Ce sont les adresses 0000 h, 1000 h, C000 h, DFFB h, EFF8 h, F003 h. On dresse d'abord la table de vérité du décodeur 1 parmi 16 utilisé pour à quelle boîtier correspond ces combinaisons d'entrée. On éclatera ensuite en binaire les adresses données pour déterminer les boîtiers et les rangs des octets désignés par ces adresses.

Table de vérité

A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	Sortie valide			A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Ac en	
0	0	0	S <sub>0</sub> (Rom)			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	S <sub>1</sub>			0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	1	0	S <sub>2</sub> (Ram)			1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C
0	1	1	S <sub>3</sub>			1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	D
1	0	0	S <sub>4</sub>			1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	D
1	0	1	S <sub>5</sub>			1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	E
1	1	0	S <sub>6</sub> (PIA 1)			1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	0	0	F
1	1	1	S <sub>7</sub> (PIA 2)			1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	F

Les bits barrés dans la table des adresses sont des bits non utilisés par les circuits correspondants. En analysant la table de vérité et la table des adresses, on trouve :

- Les adresses 0000 et 1000 correspondant au 1<sup>er</sup> octet de la Rom.
  - Les adresses C000 et DFFB appartiennent au PIA\_1 et correspondent respectivement au 1<sup>er</sup> et au 4<sup>ème</sup> octet.
  - Les adresses EFF8 et F003 appartiennent au PIA\_2 et correspondent respectivement au 1<sup>er</sup> et au 4<sup>ème</sup> octet.
- 3°) Adresses du 2048<sup>ème</sup> octet de la Rom et du 256<sup>ème</sup> octet de la Ram.
- Adresses possibles du 2048<sup>ème</sup> octet de la Rom : 07FF et 17FF
  - Adresses possibles du 256<sup>ème</sup> octet de la Ram : 40FF et 50FF.

4°) L'adresse générale en binaire du 1<sup>er</sup> octet de PIA\_1 peut se mettre sous la forme suivante : 110XXXXXXXXXX000. Les X peuvent prendre n'importe quelle valeur. On peut donc en tirer 4 adresses possibles en hexadécimal : C000, CFFC, D000, DFFC.

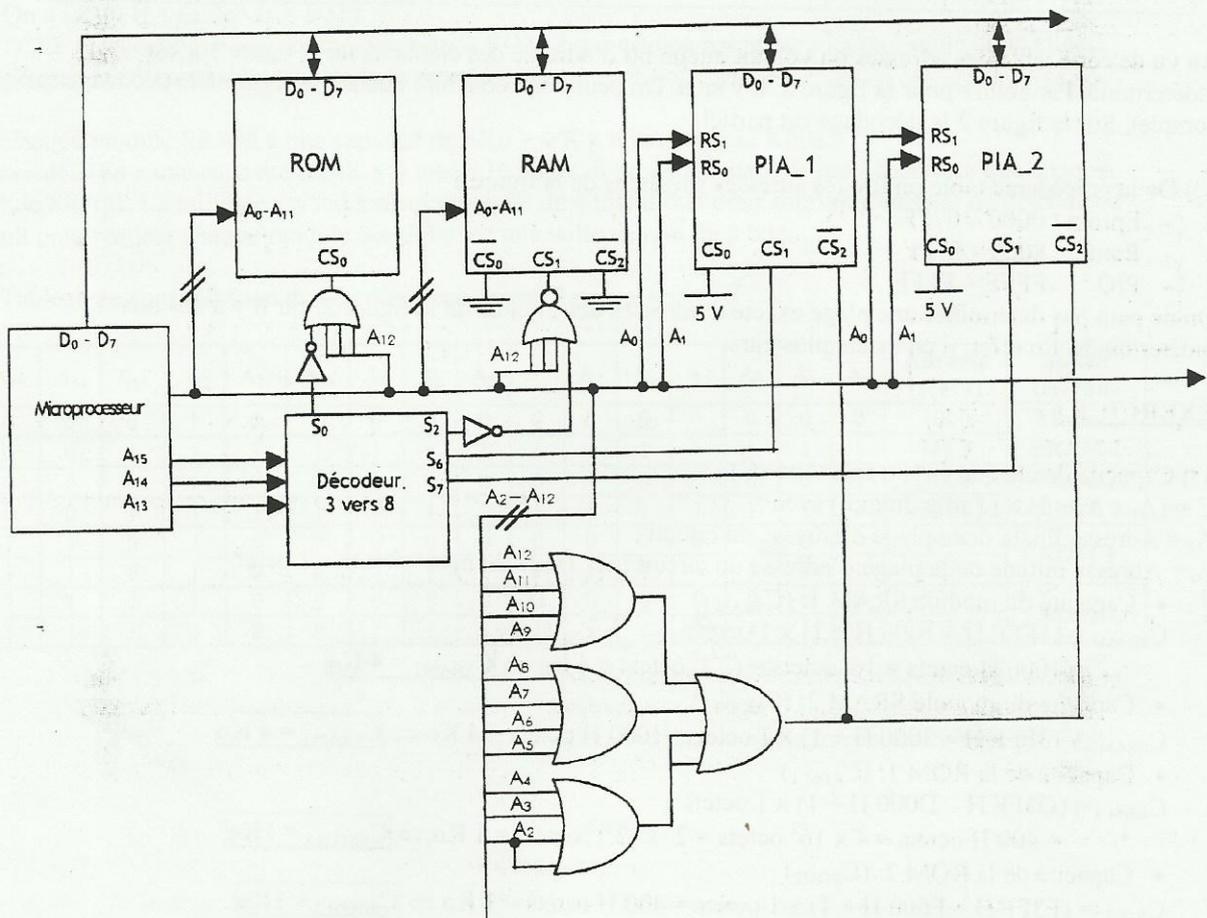
L'adresse générale en binaire du 3<sup>ème</sup> octet du PIA\_2 peut se mettre sous la forme suivante : 111XXXXXXXXXX10. Les X peuvent prendre n'importe quelle valeur. On peut donc en tirer 4 adresses possibles en hexadécimal : E002, EFFE, F002, FFFE.

Ces adresses ne sont pas exhaustives. Il en existe d'autres.

5°) Eliminer les adresses images consiste à obliger les bits indéterminés à prendre des valeurs fixes. On peut choisir par exemple de les mettre à 0. Le schéma obtenu est ci-dessous.

6°) Mapping du décodage complet.

A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Adresse	Circuit
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	Rom
				1	1	1	1	1	1	1	1	1	1	1	1	0FFF	
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000	Ram
				1	1	1	1	1	1	1	1	1	1	1	1	4FFF	
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C000	PIA_1
														1	1	C003	
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	E000	PIA_2
														1	1	E003	



**EXERCICE 5**

1°) Identification des types de décodage

	A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Circuit	
Figure 1	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	Eprom	
						1	1	1	1	1	1	1	1	1	1	1	1	2 Ko
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Ram		
						1	1	1	1	1	1	1	1	1	1	1	2 Ko	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	
																	1	PIO
Figure 2	0	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	Eprom	
						1	1	1	1	1	1	1	1	1	1	1	1	2 Ko
	1	0	X	X	X	0	0	0	0	0	0	0	0	0	0	Ram		
						1	1	1	1	1	1	1	1	1	1	1	2 Ko	
1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	
																	1	PIO

Au vu de cette table des adresses on voit qu'aucun bit d'adresse des éléments de la figure 1 n'est indéterminé. Par contre pour la figure 2, il y en a. On peut donc conclure que sur la figure 1 le décodage est complet. Sur la figure 2 le décodage est partiel.

2°) De la précédente table on tire les adresses suivantes de la figure 1 :

- Eprom : 0000 – 07FF
- Ram : 8000 – 87FF
- PIO : FFFE – FFFF

On ne peut pas déterminer une plage exacte d'adresses des circuits de la figure 2, car il y a des bits indéterminés. En effet, il en existe plusieurs.

**EXERCICE 6**

1°) Capacité de chaque circuit mémoire (C)

$C = (A_f - A_i + 1) \times (\text{Taille du mot})$  avec :

$A_f$  = Adresse finale de la plage d'adresse du circuit ;

$A_i$  = Adresse initiale de la plage d'adresse du circuit ; ici Taille du mot = 8 bits = 1 octet.

- Capacité du module SRAM 1: ( $C_{SRAM1}$ )

$$C_{SRAM1} = (1FFF H - 1000 H + 1) \times 1 \text{ octets} = 1000 H \text{ octets} = 16^3 \text{ octets} = (2^4)^3 \text{ octets} = 4 \text{ Ko} \Rightarrow \boxed{C_{SRAM1} = 4 \text{ Ko}}$$

- Capacité du module SRAM 2: ( $C_{SRAM2}$ )

$$C_{SRAM2} = (3FFF H - 3000 H + 1) \times 1 \text{ octets} = 1000 H \text{ octets} = 4 \text{ Ko} \Rightarrow \boxed{C_{SRAM2} = 4 \text{ Ko}}$$

- Capacité de la ROM 1: ( $C_{ROM1}$ )

$$C_{ROM1} = (D3FF H - D000 H + 1) \times 1 \text{ octets} = 400 H \text{ octets} = 4 \times 16^2 \text{ octets} = 2^2 \times (2^4)^2 \text{ octets} = 1 \text{ Ko} \Rightarrow \boxed{C_{ROM1} = 1 \text{ Ko}}$$

- Capacité de la ROM 2: ( $C_{ROM2}$ )

$$C_{ROM2} = (F3FF H - F000 H + 1) \times 1 \text{ octets} = 400 H \text{ octets} = 1 \text{ Ko} \Rightarrow \boxed{C_{ROM2} = 1 \text{ Ko}}$$

2°) \* L'adresse du 1000ème mot mémoire en ROM

Chaque mémoire ROM possède une capacité de 1024 octets soit 1024 mots mémoire (1 mot mémoire = 1 octet).

Le premier mot mémoire en ROM étant à l'adresse D000 H (ROM 1), le 1000<sup>ème</sup> est (également) en ROM 1 à l'adresse D000 H + 3E7 H (3E7 H = 999<sub>10</sub>) soit à l'adresse D3E7 H de l'espace adressable du microprocesseur.

\* L'adresse du 512<sup>ème</sup> position mémoire en RAM

Chaque module SRAM a une capacité de 4096 octets (plus de 512 octets) donc la 512<sup>ème</sup> position mémoire en RAM est situé dans le module SRAM 1 et son adresse est celle du premier octet du module SRAM 1 (1000 H) plus 511 octets soit :  $1000\text{ H} + 1\text{FF H} = 11\text{FF H}$  ( $1\text{FF H} = 511_{10}$ )  
 La 512<sup>ème</sup> position mémoire en RAM est donc à l'adresse 11FF H de l'espace adressage du microprocesseur.

3°) Les adresses qui permettent d'enregistrer une donnée en mémoire parmi la 16384<sup>ème</sup>, la 16400<sup>ème</sup> et la 54000<sup>ème</sup> adresse du microprocesseur.

On a :  $16384_{10} = 4000\text{ H}$ ,  $16400_{10} = 4010\text{ H}$  et  $54000_{10} = \text{D2F0 H}$ .

La 1<sup>ère</sup> (première) adresse du microprocesseur étant 0000 H alors :

- La 16384<sup>ème</sup> adresse du microprocesseur est 3FFF H ( $3\text{FFF H} = 4000\text{ H} - 1$ )

Cette adresse est celle du dernier octet du module SRAM 2 qui est une mémoire à lecture et écriture donc elle permet d'enregistrer une donnée en mémoire.

- La 16400<sup>ème</sup> adresse du microprocesseur est 400F H ( $400\text{F H} = 4010 - 1$ )

Cette adresse ne sélectionne aucun mot mémoire donc elle ne permet pas d'enregistrer une donnée en mémoire.

- La 54000<sup>ème</sup> adresse du microprocesseur est D2EF H ( $\text{D2EF H} = \text{D2F0 H} - 1$ )

On a  $\text{D000 H} < \text{D2EF H} < \text{D3FF H}$ .

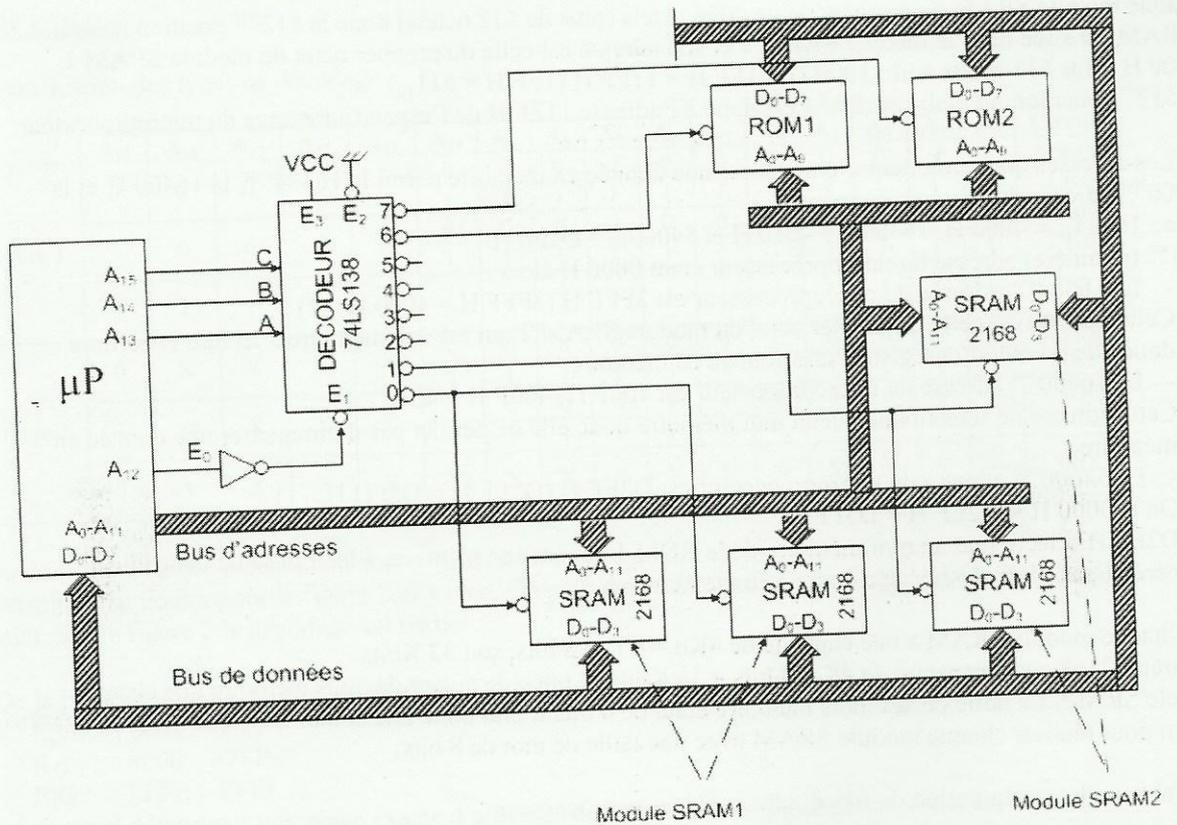
D2EF H sélectionne un mot mémoire de la ROM 1 qui est une mémoire à lecture seule donc elle ne permet pas d'enregistrer une donnée en mémoire.

4°) Chaque module SRAM a une capacité de  $4\text{Ko} = 4\text{ K} \times 8\text{ bits}$ , soit 32 Kbits.

Le circuit 2168 a une capacité de  $4\text{K} \times 4\text{ bits} = 16\text{ Kbits}$ . Il possède autant de mots mémoire que chaque module SRAM. La taille de ses mots mémoire étant de 4 bits il faut donc une combinaison de deux de ce circuit pour réaliser chaque module SRAM avec une taille de mot de 8 bits.

5°) Tableau de configuration du bus d'adresse de microprocesseur

A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Adresse (Hexa)	Boîtier (circuit)
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1000	Module SRAM 1
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1FFF	
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	3000	Module SRAM 2
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFF	
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	D000	ROM 1
1	1	0	1	0	0	1	1	1	1	1	1	1	1	1	1	D3FF	
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	F000	ROM 2
1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	F3FF	



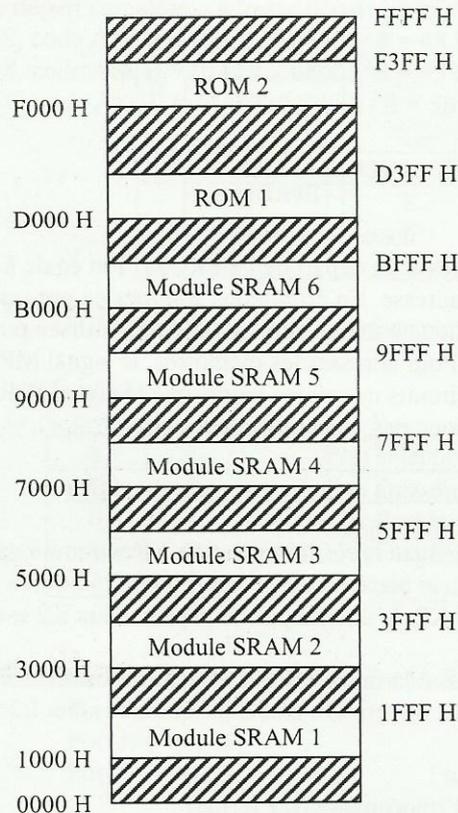
### 6°) Nouvelle cartographie

Les sorties libres du décodeur sont les sorties 2, 3, 4 et 5 auxquelles nous connectons respectivement le module SRAM 3, le module SRAM 4, le module SRAM 5 et le module SRAM 6. Ces modules sont adressés suivant la configuration ci-dessous :

Circuit	Sortie décodeur	Entrées décodeur					Plage d'adresse µP
		A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	Valeur Hexa.	
Module SRAM 3	2	0	1	0	1	5 H	5000 H à 5FFF H
Module SRAM 4	3	0	1	1	1	7 H	7000 H à 7FFF H
Module SRAM 5	4	1	0	0	1	9 H	9000 H à 9FFF H
Module SRAM 6	5	1	0	1	1	B H	B000 H à BFFF H

**Remarque :** Les bits A<sub>11</sub> à A<sub>0</sub> varient pour chaque module de 000 H à FFF H.

On obtient la cartographie suivante :



**EXERCICE 7**

A - Cartographie

1°) En analysant le schéma, on constate qu'il s'agit d'un adressage par décodage en structure E/S par instruction E/S (présence de 2 décodeurs et de 2 signaux MREQ et IC/RQ permettant de distinguer les mémoires des circuits d'E/S).

2°) Les mémoires EPROM\_1 et EPROM\_2 reçoivent toutes les deux un bus d'adresse de 11 bits ( $A_0$  à  $A_{10}$ ) et un bus de données de 8 bits ( $D_0$  à  $D_7$ ). Elles ont donc chacune une capacité de  $2^{11}$  octets, soit 2 Ko.

3°) Adresse de départ du PIO.  
 Pour valider le PIO, il faut :  $A_3 = 0, A_2 = 0$ .  
 Son premier octet sera sélectionné si :  $A_1 = 0, A_0 = 0$ .

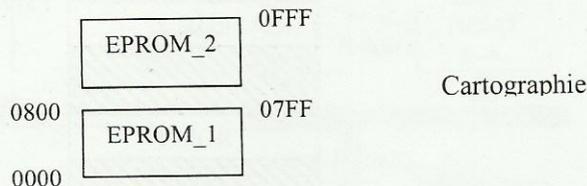
Si on met les  $A_4$  à  $A_{15}$  non utilisés du processeur à 0, on obtient pour adresse de départ du PIO 8255 l'adresse : 0000 H.

4°) Cartographie des circuits EPROM

$A_{15}$	$A_{14}$	$A_{13}$	$A_{12}$	$A_{11}$	$A_{10}$	$A_9$	$A_8$	$A_7$	$A_6$	$A_5$	$A_4$	$A_3$	$A_2$	$A_1$	$A_0$	Circuit
x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	Eprom_1
x	x	x	0	1	0	0	0	0	0	0	0	0	0	0	0	Eprom_2

On peut choisir de mettre les bits non utilisés du processeur à 0. Ce qui permet de tirer les adresses suivantes :

EPROM\_1 : 0000 – 07FF  
 EPROM\_2 : 0800 – 0FFF



5°) L'adresse de départ du PIO est égale à l'adresse de départ de l'EPROM\_1 et égale à 0000 H. On ne peut pas pour autant conclure qu'il y a un conflit d'adresse. En effet, nous sommes en présence d'une structure d'E/S par instruction E/S, structure dans laquelle une même adresse peut être utilisée pour adresser les mémoires et les E/S et ce de façon alternative. Pour adresser les mémoires, le signal MREQ (Memory REQuest) est mis à l'état bas pour valider les circuits mémoires tandis que le signal IORQ est mis à l'état haut pour invalider les circuits d'E/S. il n'y a donc pas de conflit d'adresse.

### B - Gestion de l'impression et du mouvement du papier

1°) Si l'imprimante est allumée, la tête sera forcement après initialisation à l'extrémité gauche, K1 sera fermé pour indiquer cela. S'il n'y a pas de papier dans le bac à papier, l'imprimante indiquera « PAPER OUT ». En effet c'est le papier qui ferme l'interrupteur K2. S'il n'y a pas de papier, alors K2 est ouvert.

2°) Si K1 et K2 sont fermés, E1 et E2 sont reliés à la masse. Si K1 et K2 sont ouverts, E1 et E2 sont reliés à +5V. Or dans le cas présent, K1 est fermé et K2 ouvert. On en déduit que E1 = 0 et E2 = 1

3°) Pour qu'une impression soit possible, il faut :

- que la tête soit à l'extrémité gauche de l'imprimante (K1 fermé)
- qu'il y est du papier dans le bac (K2 fermé)

Si K1 et K2 sont fermés, alors on aura E1 = 0 et E2 = 0.

### C - Génération des caractères du code ASCII

1°) La matrice caractère est faite de 8 octets. Pour 8 bits par octet, la matrice caractères en bits est de 64 bits.

2°) L'EPROM\_2 a une taille de 2 Ko, soit 2048 octets. Si un caractère occupe 8 octets dans cette mémoire, on aura  $2048 / 8 = 256$  caractères.

3°) La formule de calcul de l'adresse de départ est donnée.  
 Adresse départ de P =  $0800\text{ H} + 50\text{ H} \times 8 = 0A80\text{ H}$ .

**Adresse départ de P = 0A80 H**

**Adresse fin de P = 0A87 H**

La lettre P est le 81<sup>ème</sup> caractères de la table ASCII ( $50_{16} = 80_{10}$ ). Dans l'EPROM\_2 la case mémoire où cette lettre se trouve est la suivante :

**Rang = (adresse départ caractère – adresse départ EPROM) + 1**, soit  $(0A80 - 0800) + 1 = 281_{16} = 641_{10}$   
**Rang = 641<sup>ème</sup>**

4°) Le code ASCII d'un caractère est égale au rang du caractère dans la table diminué d'un rang.

- Code ASCII du 93<sup>ème</sup> caractère =  $(93 - 1)_{10} = 5C_{16}$ .
- Code ASCII du 129<sup>ème</sup> caractère =  $(129 - 1)_{10} = 80_{16}$ .
- Code ASCII du 256<sup>ème</sup> caractère =  $(256 - 1)_{10} = FF_{16}$ .

	93 <sup>ème</sup> caractère	129 <sup>ème</sup> caractère	256 <sup>ème</sup> caractère
Adresse	0AE0 H	0C00 H	0FF8 H
Rang dans l'EPROM	737 <sup>ème</sup>	1025 <sup>ème</sup>	2041 <sup>ème</sup>

5°) De la formule du 3°), on peut tirer la formule du code ASCII :

Code ASCII = (adresse de départ caractère – adresse départ EPROM) / 8.

- pour l'adresse 0A58, code ASCII = (0A58 – 0800) / 8 = 48 H
- pour l'adresse 0A18, code ASCII = (0A18 – 0800) / 8 = 43 H
- pour l'adresse 0AC0, code ASCII = (0AC0 – 0800) / 8 = 58 H

Code ASCII	Caractère
48H	K
43H	C
58H	X

D- Programmation de l'interface parallèle 8255

1°) L'énoncé donne le tableau suivant :

A <sub>1</sub>	A <sub>0</sub>	Port sélectionné
0	0	Port A
0	1	Port B
1	0	Port C
1	1	Registre de commande

De ce tableau, on tire que les adresses des ports A, B et C sont les suivantes :

Port A = 0000H

Port B = 0001H

Port C = 0002H

2°)

- Port A programmé en entrée
- Port B programmé en sortie
- Port C programmé en sortie

3°) Les groupes A et B sont programmés en mode 0 tous les deux. En effet, c'est dans ce mode que chaque port A, B, C peut être programmé en entrée ou en sortie, indépendamment les uns des autres.

4°) Mot de commande du 8255.

- Les modes 0 des groupes A et B imposent : D<sub>6</sub> = 0, D<sub>5</sub> = 0, D<sub>2</sub> = 0
- Le port A en entrée impose : D<sub>4</sub> = 1
- Le port B en sortie impose : D<sub>1</sub> = 0
- Le port C en sortie impose : D<sub>3</sub> = 0, D<sub>0</sub> = 1.

Mot de commande du 8255							
D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1	0	0	1	0	0	0	1
<b>Mot de commande = 91H</b>							

5°) Si les échanges entre imprimante et PC ont lieu sur le principe du « Handshacking », les groupes A et B passent en mode 1. Dans ce mode, D<sub>6</sub> = 0, D<sub>5</sub> = 1, D<sub>2</sub> = 1.

Mot de commande du 8255							
D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1	0	1	1	0	1	0	1
<b>Mot de commande = B5H</b>							

6°) Si le mot de commande est 86H, on a :

Mot de commande du 8255							
D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1	0	0	0	0	1	1	0
Mot de commande = 86H							

- D<sub>6</sub> = 0 et D<sub>5</sub> = 0 donnent le mode 0 pour le groupe A
- D<sub>2</sub> = 1, donne mode 1 pour le groupe B
- D<sub>4</sub> = 0 donne le port A en sortie
- D<sub>1</sub> = 0 donne le port B en entrée
- D<sub>3</sub> = 0 donne le quartet supérieur du port C en sortie
- D<sub>2</sub> = 1 donne le quartet inférieur du port C en entrée.

### **EXERCICE 8**

1°)

- La mémoire 1 est une EEPROM en technologie CMOS de capacité 64 Kbits d'après l'inscription 28C64.
- La mémoire 2 est une UVPROM. En effet, les signaux V<sub>pp</sub> (alimentation de programmation) et PGM (signal de positionnement de la mémoire en mode programmation) sont propres aux UVPROM.
- La mémoire 3 est aussi une UVPROM pour les mêmes raisons que précédemment.

2°)

Mémoire 1 :

- bus d'adresse = 13 bits (A<sub>0</sub> à A<sub>12</sub>)
- bus de donnée = 8 bits (I/O<sub>1</sub> à I/O<sub>8</sub>)

Mémoire 2 :

- bus d'adresse = 14 bits (A<sub>0</sub> à A<sub>13</sub>)
- bus de donnée = 8 bits (O<sub>0</sub> à O<sub>7</sub>)

Mémoire 3 :

- bus d'adresse = 11 bits (A<sub>0</sub> à A<sub>10</sub>)
- bus de donnée = 8 bits (O<sub>0</sub> à O<sub>7</sub>)

3°)

Capacité	Mémoire 1	Mémoire 2	Mémoire 3
En mots mémoires	8192	16384	2048
En Ko	8	16	2
En Kbits	64	128	16

4°) D'après les capacités en Kbits, on déduit comme référence :

**Mémoire 2 : 27C128**

**Mémoire 3 : 27C16**

**EXERCICE 9**

**A- Lecture de schéma**

- 1°)
- Nombre de lignes d'adresses du processeur = 16 bits ( $A_0 - A_{16}$ )
  - Nombre de lignes de données du processeur = 8 bits ( $D_0 - D_7$ )

- 2.a) La Ram et l'Eprom ont la même capacité.
- Capacité de chaque boîtier mémoire : 2 Ko
  - Capacité sous la forme  $N \times T$  :  $2 K \times 8 \text{ bits} = 2048 \times 8 \text{ bits}$

2.b) Capacité de la mémoire centrale = 4 Ko

3.a) La broche de chacune des 3 boîtiers est active à l'état bas.

3.b) Equations de sélection

$$\overline{CS}_{Eprom} = \overline{A_{15} + A_{14} + A_{13} + A_{12} + A_{11}} = \overline{A_{15}} + \overline{A_{14}} + \overline{A_{13}} + \overline{A_{12}} + \overline{A_{11}}$$

$$\overline{CS}_{Ram} = \overline{A_{15} + A_{14} + A_{13} + A_{12} + A_{11}} = \overline{A_{15}} + \overline{A_{14}} + \overline{A_{13}} + \overline{A_{12}} + \overline{A_{11}}$$

$$\overline{CS}_{PIO} = \overline{A_{15} \cdot A_{14} \cdot A_{13} \cdot A_{12} \cdot A_{11} \cdot A_{10} \cdot A_9 \cdot A_8 \cdot A_7 + A_6 \cdot A_5 \cdot A_4 \cdot A_3 \cdot A_2}$$

- 3.c)
- Pour valider les mémoires, les bits de sélection doivent être tous à 0.
  - Pour valiser le PIO, tous les bits de sélection doivent être à 1.

4°) Le mode d'adressage utilisé est l'adressage par sélection linéaire.

**B- Cartographie**

1°) Il n'y a pas d'adresses images sur ce schéma. Aucun des 3 circuits ne dispose de bit indéterminé. En effet, pour la sélection de chaque boîtier, une seule combinaison des bits de sélection est possible. Les autres bits utilisés pour l'adressage internes des boîtiers sont tous utilisés.

2°) Table des adresses

A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Circuits	adresse
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Eprom	0000
					1	1	1	1	1	1	1	1	1	1	1		
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Ram	8000
					1	1	1	1	1	1	1	1	1	1	1		
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	PIO	FFFC
														1	1		

3°) La mémoire centrale fait 4 Ko soit 4096 cases mémoires. L'Eprom possédant 2048 octets, occupe la partie basse de l'espace mémoire centrale. La 3072<sup>ème</sup> case mémoire centrale appartient donc à la Ram et se trouve à la  $(3072 - 2048) = 1024$ <sup>ème</sup> position mémoire de la Ram. Cette 1024<sup>ème</sup> case mémoire de la Ram se trouve à l'adresse  $A = (1024 - 1)_{10} + 8000_{16} = 03FF + 8000 = 83FF$ .

**La 3072<sup>ème</sup> case mémoire centrale se trouve à l'adresse 83FF**

4°) L'adresse 07FF correspond à l'adresse de la dernière case de l'Eprom, soit la 2048<sup>ème</sup> position mémoire de l'Eprom et de la mémoire centrale.

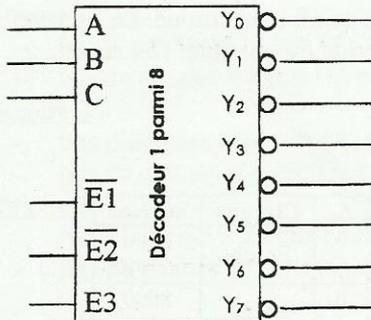
**EXERCICE 10**

1°) La table des adresses est la suivante :

A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Circuit	Adresse
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	Moniteur	F800
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFFF		
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	Circuit interface	F000
1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	F7FF		
1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	Eprom	E800
1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	FFFF		
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Bloc_1	E000
1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	E7FF		
1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	Bloc_2	D800
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	DFFF		
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	Bloc_3	D000
1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	D7FF		
1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	Bloc_4	C800
1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	CFFF		
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bloc_5	C000
1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	C7FF		

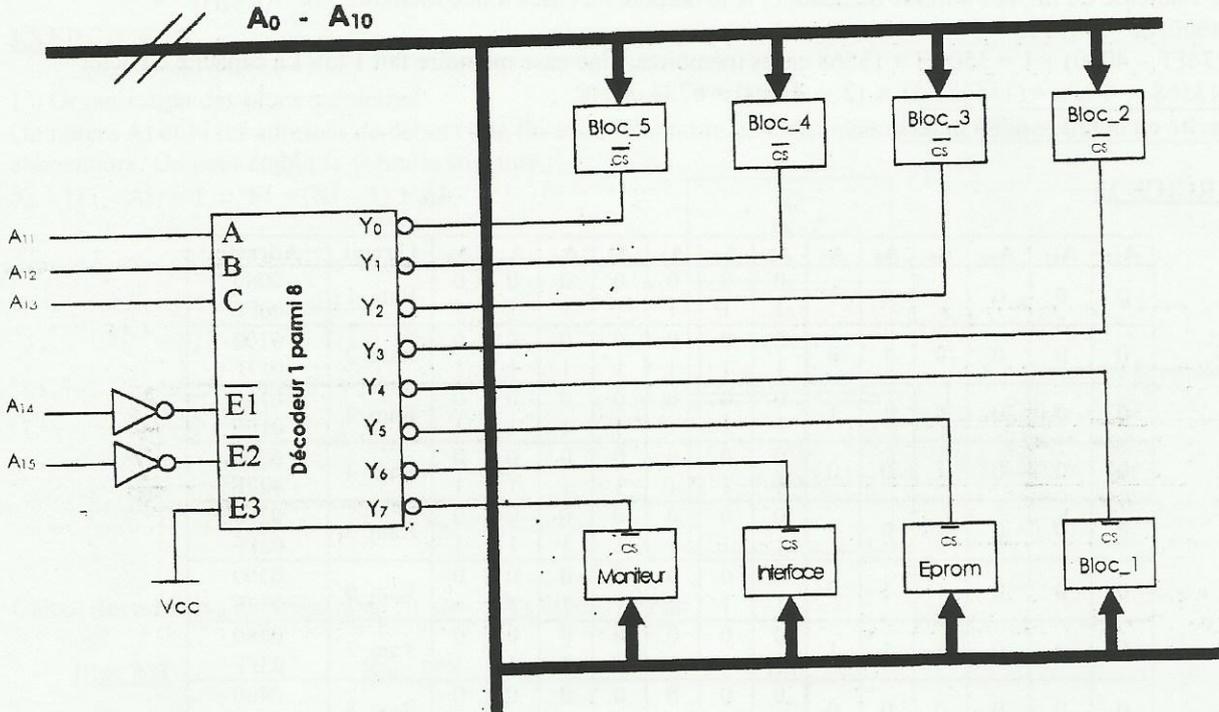
On constate dans cette table que les A<sub>15</sub> et A<sub>14</sub> sont fixés, que les bits A<sub>13</sub>, A<sub>12</sub>, A<sub>11</sub> constituent les variables d'entrée du décodeur. Les bits A<sub>0</sub> à A<sub>10</sub> serviront à l'adressage interne des circuits.

**Le décodeur et sa table de vérité**



A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	Sortie	Circuit connecté
0	0	0	/Y <sub>0</sub>	Bloc_5
0	0	1	/Y <sub>1</sub>	Bloc_5
0	1	0	/Y <sub>2</sub>	Bloc_5
0	1	1	/Y <sub>3</sub>	Bloc_5
1	0	0	/Y <sub>4</sub>	Bloc_5
1	0	1	/Y <sub>5</sub>	Eprom
1	1	0	/Y <sub>6</sub>	Interface
1	1	1	/Y <sub>7</sub>	Moniteur

**Schéma d'implantation**



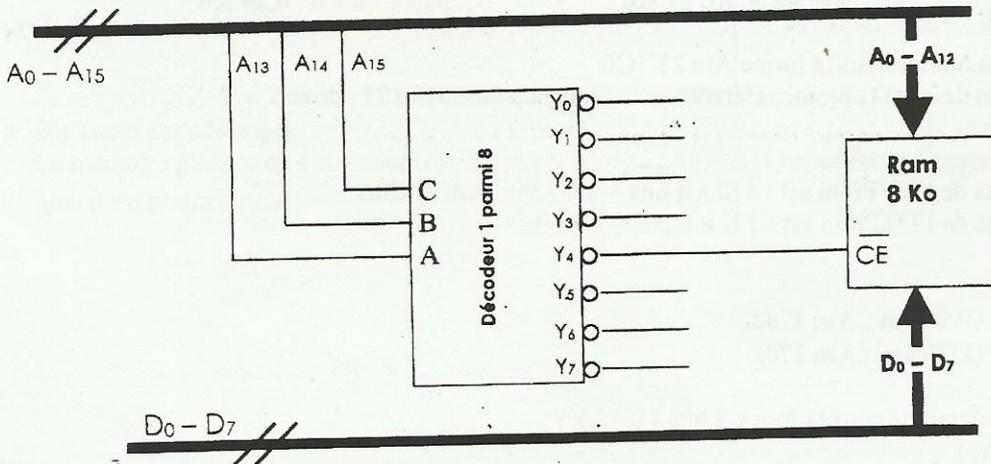
Un bus de donnée n'a pas été représenté sur cette figure. Le décodage qui est fait sur ce schéma est un décodage complet. En effet, aucun bit d'adresse n'a de valeur indéterminée.

**EXERCICE 11**

En éclatant l'adresse 8000H en binaire, on a la table des adresses suivantes :

A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Adresse en hex.
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8000

Pour une Ram de 8 Ko, 13 fils d'adresse (A<sub>0</sub> - A<sub>12</sub>) seront nécessaires pour faire l'adressage interne de la mémoire. Les 3 fils restant (A<sub>13</sub>, A<sub>14</sub>, A<sub>15</sub>) serviront d'entrées de variables pour le décodeur utilisé (décodeur 1 parmi 8). La combinaison A<sub>15</sub> = 1, A<sub>14</sub> = 0, A<sub>13</sub> = 0 doit valider la sortie Y<sub>4</sub> du décodeur. On aboutit au schéma de décodage suivant



**EXERCICE 12**

Soit F l'adresse de fin, A l'adresse de début et X le nombre de cases d'une mémoire :  $X = (F - A) + 1$   
 $A = 4000H, F = 74FFH$

$X = (74FF - 4000) + 1 = 3500H = 13568$  cases mémoires. Une case mémoire fait 1 bit. La capacité en octet est :  $13568 \times 4 \text{ bits} = (13568 / 2) \times (2 \times 4 \text{ bits}) = 6784$  octets.

**Capacité en octets = 6784 octets**

**EXERCICE 13**

A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Circuit	Adresse
0	0	0	0	0	1	0	0	0	0	0	0	0	Ram_1	0080 00FF
0	0	0	0	1	0	0	0	0	0	0	0	0	Ram_2	0100 017F
0	0	0	0	1	1	0	0	0	0	0	0	0	Ram_3	0180 01FF
0	0	0	1	0	0	0	0	0	0	0	0	0	Ram_4	0200 027F
0	0	0	1	0	1	0	0	0	0	0	0	0	Ram_5	0280 02FF
0	0	0	1	1	0	0	0	0	0	0	0	0	Ram_6	0300 037F
0	0	0	1	1	1	0	0	0	0	0	0	0	Ram_7	0380 03FF
0	0	0	0	0	0	0	0	0	0	0	0	0	Ram_8	0400 047F

**EXERCICE 14**

1°)

	Ram dynamique 16K × 1 bit	OT Prom 1K × 8 bits	UVProm 512 × 4 bits
Nb. de mots	16384	1024	512
Nb. de lignes d'adresse	14	10	9
Nb. de lignes de données	1	8	4

2°)

Ram dynamique

OT Prom

UVProm

Capacité en octets : **2048 octets**  
 Capacité en Ko : **2 Ko**

Capacité en octets : **1024 octets**  
 Capacité en Ko : **1 Ko**

Capacité en octets : **256 octets**  
 Capacité en Ko : **0,25 Ko**

3°) Référence des mémoires mortes sous la forme Am 2Y XX'

Les UVProm et les OTProm de AMD ont une référence qui commence le chiffre 27 : d'où Y = 7. XX désigne la capacité en Kbits.

- La capacité en Kbits de l'UVProm est :  $512 \times 4 \text{ bits} = 2048 \text{ bits}$ , soit 2 Kbits.
- La capacité en Kbits de l'OTProm est :  $1 \text{ K} \times 8 \text{ bits} = 8 \text{ Kbits}$ .

D'où les références suivantes :

- **Référence UVProm : Am 2702**
- **Référence OTProm : Am 2708**

4°) Référence de la Ram dynamique sous la forme TMS 4 X C YY

X = 1 (taille du mot mémoire)

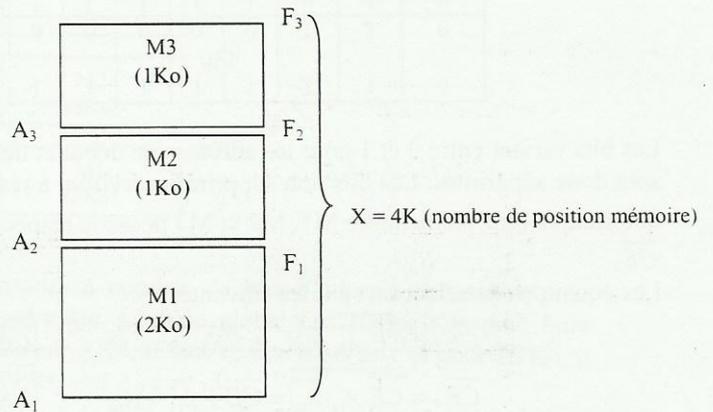
YY = 16 (nombre de mots en kilo)  
 Référence : 41C16

**EXERCICE 15**

1°) Organisation des blocs mémoires

On notera  $A_i$  et  $F_i$  les adresses de début et de fin et  $X_i$  le nombre de cases mémoires de chaque bloc élémentaire. On peut établir la formule suivante :

$$X_i = (F_i - A_i) + 1 \Rightarrow F_i = (X_i - 1) + A_i$$



Calcul des adresses de début et de fin des blocs élémentaires.

**Bloc M1**

$A_1 = 0000H$   
 $X_1 = 2K = 0800H$   
 $F_1 = (0800 - 1) + 0000$   
 **$F_1 = 07FF$**

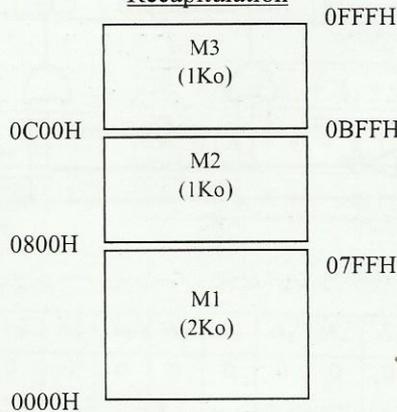
**Bloc M2**

$A_2 = F_1 + 1 = 07FF + 1 = 0800H$   
 $X_2 = 1K = 0400H$   
 $F_2 = (0400 - 1) + 0800$   
 **$F_2 = 0BFF$**

**Bloc M3**

$A_3 = F_2 + 1 = 0BFF + 1 = 0C00H$   
 $X_3 = 1K = 0400H$   
 $F_3 = (0400 - 1) + 0C00$   
 **$F_3 = 0FFF$**

Recapitulation



2°) Table des adresses

La mémoire globale de 4 Ko nécessite 12 bits d'adresse ( $A_0 - A_{11}$ ) pour être adressée. La sélection du bloc global est nommée  $\overline{CE}$ .

$\overline{CE}$	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Circuit
0	0	0	0	0	0	0	0	0	0	0	0	0	<b>0000</b>
0	0	1	1	1	1	1	1	1	1	1	1	1	<b>M1</b> <b>07FF</b>
0	1	0	0	0	0	0	0	0	0	0	0	0	<b>0800</b>
0	1	0	1	1	1	1	1	1	1	1	1	1	<b>M2</b> <b>0BFF</b>
0	1	1	0	0	0	0	0	0	0	0	0	0	<b>0C00</b>
0	1	1	1	1	1	1	1	1	1	1	1	1	<b>M3</b> <b>0FFF</b>

Les bits variant entre 0 et 1 pour les adresses de début et de fin ne participent aux équations de sélection. Ils sont donc supprimés. Les bits non supprimés serviront à réaliser les fonctions de sélection. On considèrera que chaque bloc élémentaire M1, M2 et M3 possède respectivement une broche de sélection  $\overline{CE}_1$ ,  $\overline{CE}_2$ ,  $\overline{CE}_3$ .

Les équations de sélection sont les suivantes :

$$\overline{CE}_1 = \overline{CE} \cdot \overline{A_{11}} = A_{11} + \overline{CE}$$

$$\overline{CE}_1 = A_{11} + \overline{CE}$$

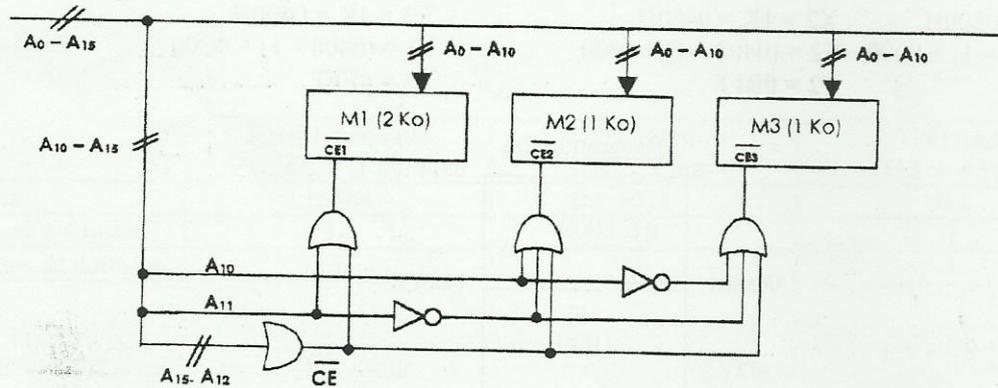
$$\overline{CE}_2 = \overline{CE} \cdot \overline{A_{11}} \cdot \overline{A_{10}} = \overline{CE} + \overline{A_{11}} + A_{10}$$

$$\overline{CE}_2 = A_{11} + A_{10} + \overline{CE}$$

$$\overline{CE}_3 = \overline{CE} \cdot \overline{A_{11}} \cdot A_{10} = \overline{CE} + \overline{A_{11}} + A_{10}$$

$$\overline{CE}_3 = A_{11} + A_{10} + \overline{CE}$$

### Schéma de câblage

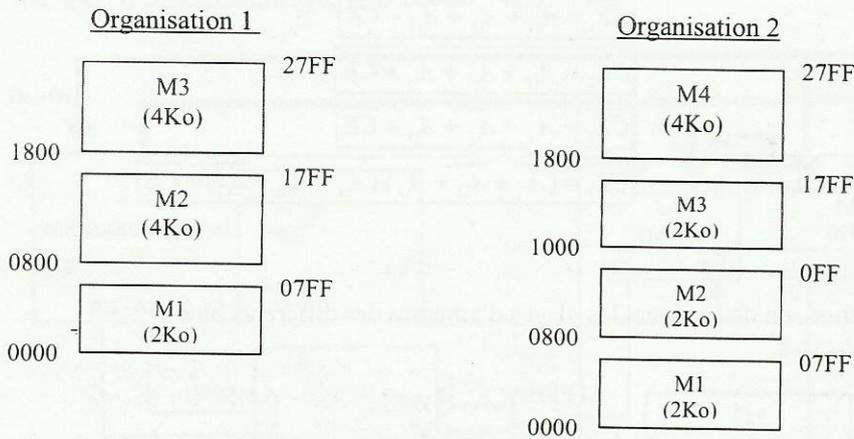


### EXERCICE 16

1°) Pour constituer un bloc de 10 Ko, on peut utiliser :

- 2 blocs élémentaires de 4 Ko et un bloc élémentaire de 2 Ko.
- 3 blocs élémentaires de 2 Ko et un bloc élémentaire de 4 Ko.

2°)



3°) Pour écrire les équations de sélections, on détermine le nombre de bits d'adresse nécessaire pour adresser le bloc global, puis on dressera la table des adresses. Le bloc global fait 10 Ko. On peut faire l'encadrement suivant : 8 Ko < 10 Ko < 16 Ko. Pour adresser 8 Ko, il faut 13 bits d'adresse et pour 16 Ko, il faut 14 bits. Pour 10 Ko, on ne peut alors prendre que 14 bits, soit A<sub>0</sub> à A<sub>13</sub>.

**Organisation 1**

$\overline{CE}$	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Circuit
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000 <u>M1</u> 07FF
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0800 <u>M2</u> 17FF
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1800 <u>M3</u> 27FF
0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	

$$\overline{CE}_1 = \overline{CE} \cdot \overline{A_{11}} \cdot \overline{A_{12}} \cdot \overline{A_{13}}$$

$$\overline{CE}_1 = A_{11} + A_{12} + A_{13} + \overline{CE}$$

$$\overline{CE}_2 = \overline{CE} \cdot (\overline{A_{11}} \cdot \overline{A_{12}} \cdot \overline{A_{13}} + A_{11} \cdot \overline{A_{12}} \cdot \overline{A_{13}})$$

$$\overline{CE}_2 = (\overline{A_{11}} + \overline{A_{12}} + \overline{A_{13}}) \cdot (A_{11} + A_{12} + A_{13}) + \overline{CE}$$

$$\overline{CE}_3 = \overline{CE} \cdot (\overline{A_{11}} \cdot \overline{A_{12}} \cdot \overline{A_{13}} + A_{11} \cdot \overline{A_{12}} \cdot \overline{A_{13}})$$

$$\overline{CE}_3 = (\overline{A_{11}} + \overline{A_{12}} + \overline{A_{13}}) \cdot (A_{11} + A_{12} + A_{13}) + \overline{CE}$$

**Organisation 2**

$\overline{CE}$	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Circuit
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000 <u>M1</u> 07FF
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0800 <u>M2</u> 0FFF
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1000 <u>M3</u> 17FF
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1800 <u>M4</u> 27FF
0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	

$$\overline{CE_1} = \overline{CE \cdot A_{11} \cdot A_{12} \cdot A_{13}}$$

$$\overline{CE_2} = \overline{CE \cdot A_{11} \cdot A_{12} \cdot A_{13}}$$

$$\overline{CE_3} = \overline{CE \cdot A_{11} \cdot A_{12} \cdot A_{13}}$$

$$\overline{CE_4} = \overline{CE \cdot (A_{11} \cdot A_{12} \cdot \overline{A_{13}} + \overline{A_{11}} \cdot \overline{A_{12}} \cdot A_{13})}$$

$$\overline{CE_1} = A_{11} + A_{12} + A_{13} + \overline{CE}$$

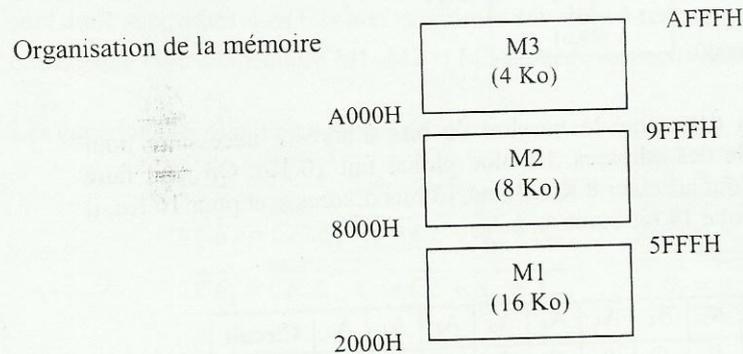
$$\overline{CE_2} = \overline{A_{11}} + A_{12} + A_{13} + \overline{CE}$$

$$\overline{CE_3} = A_{11} + \overline{A_{12}} + A_{13} + \overline{CE}$$

$$\overline{CE_4} = (\overline{A_{11}} + \overline{A_{12}} + A_{13}) \cdot (A_{11} + A_{12} + \overline{A_{13}}) + \overline{CE}$$

### EXERCICE 17

1°) Pour écrire les équations de sélection, on déterminera les plages d'adresses des différents blocs mémoires, on dressera la table des adresses.



La mémoire globale a une capacité de 28 Ko. Cette capacité est comprise entre 16 Ko et 32 Ko dont les bus d'adresse respectifs sont 14 bits et 16 bits. Il faudra donc 15 bits pour adresser la mémoire de 28 Ko, soit  $A_0$  à  $A_{14}$ . Avec 15 bits d'adresse, la plage d'adresse maximale que l'on peut avoir est : **0000 - 7FFF** si l'adresse de départ commence à 0000H. Pour gérer les espaces au-delà de 7FFF, il faut un bus d'adresse de 16 bits ( $A_0$  à  $A_{15}$ ).

La table des adresses est la suivante :

$A_{15}$	$A_{14}$	$A_{13}$	$A_{12}$	$A_{11}$	$A_{10}$	$A_9$	$A_8$	$A_7$	$A_6$	$A_5$	$A_4$	$A_3$	$A_2$	$A_1$	$A_0$	Circuit
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	<b>2000</b> <b>M1</b> <b>5FFF</b>
0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	<b>8000</b> <b>M2</b> <b>9FFF</b>
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<b>A000</b> <b>M3</b> <b>AFFF</b>
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	

$$\overline{CE_1} = \overline{(A_{13} \cdot A_{14} \cdot A_{15} + A_{13} \cdot \overline{A_{14}} \cdot \overline{A_{15}})}$$

$$\overline{CE_2} = \overline{A_{13} \cdot A_{14} \cdot A_{15}}$$

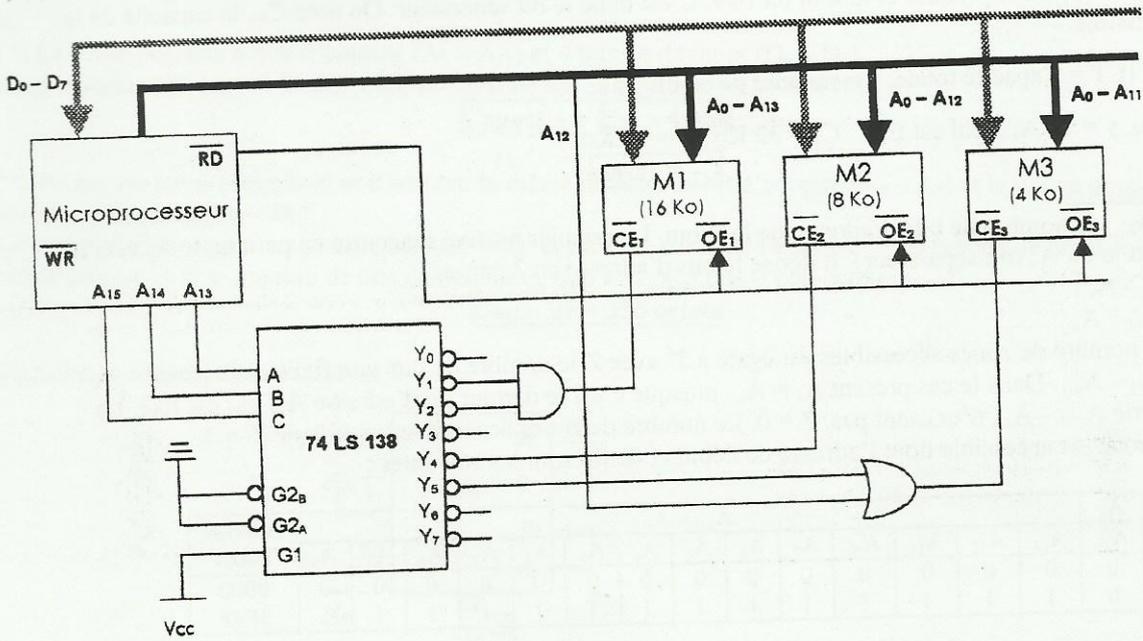
$$\overline{CE_3} = \overline{A_{12} \cdot A_{13} \cdot A_{14} \cdot A_{15}}$$

$$\overline{CE_1} = (\overline{A_{13}} + A_{14} + A_{15}) \cdot (A_{13} + \overline{A_{14}} + \overline{A_{15}})$$

$$\overline{CE_2} = A_{13} + A_{14} + A_{15}$$

$$\overline{CE_3} = A_{12} + \overline{A_{13}} + A_{14} + \overline{A_{15}}$$

2°) Schéma d'implantation avec le décodeur 74LS138.



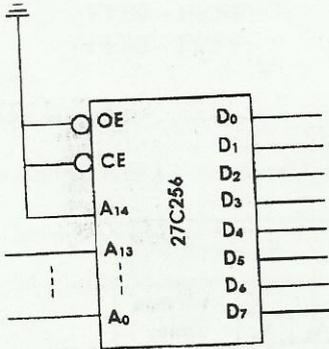
**EXERCICE 18**

La Prom 27C256 est une PROM de 32 Ko. Le nombre de bit d'adresse nécessaire pour l'adresser est de 15 bits ( $A_0 - A_{14}$ ).

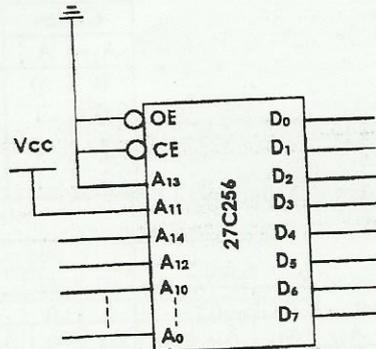
- 1-a) Le bit de poids fort est le bit  $A_{14}$  et il est fixé à 0.
- 1-b) On fixe les  $A_{13} =$  et  $A_{11} = 1$

La Prom 27C256 possède 2 broches de sélection CE et OE qu'il faudra mettre à la masse pour placer la Prom constamment en mode lecture. On aboutit aux schémas fonctionnels suivants :

On fixe  $A_{14} = 0$



On fixe  $A_{13} = 0$  et  $A_{11} = 1$



2-a)

On fixe  $A_{14}$  = qui est le premier et le seul bit fixé. C'est donc le bit séparateur. On note  $C_{PA}$  la capacité de la partie accessible.

$$C_{PA} = \frac{C_T}{2^y} \quad (C_T = \text{Capacité totale, } y = \text{nombre de bits fixés})$$

$$C_T = 32 \text{ Ko, } y = 1, A_{14} \text{ seul est fixé : } C_T = 32 \text{ Ko} / 2 = 16 \text{ Ko}$$

$$\boxed{C_T = 16 \text{ Ko}}$$

Zones accessibles

Soit  $A_0$  à  $A_{n-1}$  le nombre de bits d'adresse de la Prom. Le premier bit fixé rencontré en partant de  $A_0$  vers le bit  $A_{n-1}$  est le bit  $A_x$  (bit séparateur). Il divise le bus d'adresse en 2 parties :

- $A_0 - A_{x-1}$
- $A_{x+1} - A_{n-1}$
- Le nombre de zones accessibles est égale à  $2^Z$  avec  $Z$  le nombre de bits non fixés de la partie  $A_{x+1} - A_{n-1}$ . Dans le cas présent  $A_x = A_{n-1}$  puisque c'est le dernier bit d'adresse  $A_{14}$  qui est fixé. La partie  $A_{x+1} - A_{n-1}$  n'existant pas,  $Z = 0$ . Le nombre de zones accessibles vaut donc  $2^0 = 1$ .

Une seule zone est accessible dont l'adresse de début et de fin sont les suivantes :

$A_x$	$A_{x-1} - A_0$														Adresse zones	
$A_{14}$	$A_{13}$	$A_{12}$	$A_{11}$	$A_{10}$	$A_9$	$A_8$	$A_7$	$A_6$	$A_5$	$A_4$	$A_3$	$A_2$	$A_1$	$A_0$		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFF

2-b) On fixe  $A_{13} = 0$  et  $A_{11} = 1$

**Capacité de la partie accessible**

$$Y = 2 \Rightarrow C_{pa} = 32 \text{ Ko} / 4 \Rightarrow \boxed{C_{pa} = 8 \text{ Ko}}$$

**Zones accessibles**

$A_x = A_{11}$  (bit séparateur). Les bits non fixés de la partie:  $A_{x+1} - A_{n-1}$  sont  $A_{12}$  et  $A_{14}$ , donc  $Z = 2$ . Le nombre de zones accessibles vaut  $2^2 = 4$  zones.  $A_{12}$  et  $A_{14}$  permettent également de trouver le code de chaque zone, un code étant une combinaison des bits non fixés de la partie  $A_{x+1} - A_{n-1}$ .

Codes		
$A_{12}$	$A_{14}$	
0	0	Zone 1
0	1	Zone 2
1	0	Zone 3
1	1	Zone 4

Table des adresses

$A_{n-1} - A_{x+1}$			$A_x$	$A_{x-1} - A_0$											Adresse zones	
$A_{14}$	$A_{13}$	$A_{12}$	$A_{11}$	$A_{10}$	$A_9$	$A_8$	$A_7$	$A_6$	$A_5$	$A_4$	$A_3$	$A_2$	$A_1$	$A_0$		
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0800 Zone 1 0FFF
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1800 Zone 2 1FFF
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	4800 Zone 3 4FFF
1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	5800 Zone 4 5FFF

**EXERCICE 19**

A/

1°) La Prom possède 8 bits d'adresse ( $A_8 - A_{15}$ ) et 4 bits de données ( $D_0 - D_3$ ).  
 Sous la forme  $Nb\_mot \times Taille\_mot$ , sa capacité est  $28 \times 4 \text{ bits} = 256 \times 4 \text{ bits}$ .

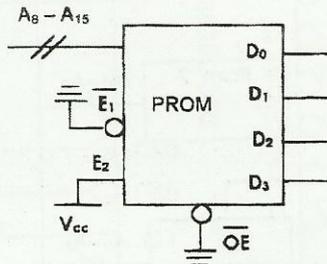
**Capacité = 256 × 4 bits**

2°) Toutes les mémoires (Rom et Ram) ont le même nombre de bit d'adresse ( $A_0 - A_7$ ) et le même nombre de bits de données ( $D_0 - D_7$ ).

La capacité d'une mémoire peut se calculer avec la formule suivante :  $Cap = 2^n \times D$  (avec n le nombre de bits d'adresse et D le nombre de bits de données).  $Cap = 2^8 \times 8 \text{ bits} = 256 \text{ octets}$ .

**Capacité = 256 octets**

3°) Schéma fonctionnel de la Prom



4°) On dispose d'un processeur de 16 bits d'adresse ( $A_0 - A_{15}$ ). L'espace d'adresse que le processeur pourra balayer est donc le suivant : 0000 - FFFF. L'espace d'adresse occupé par l'ensemble des blocs mémoires ne couvre pas la totalité de cet espace mémoire du processeur. Il y a donc des espaces libres à déterminer.

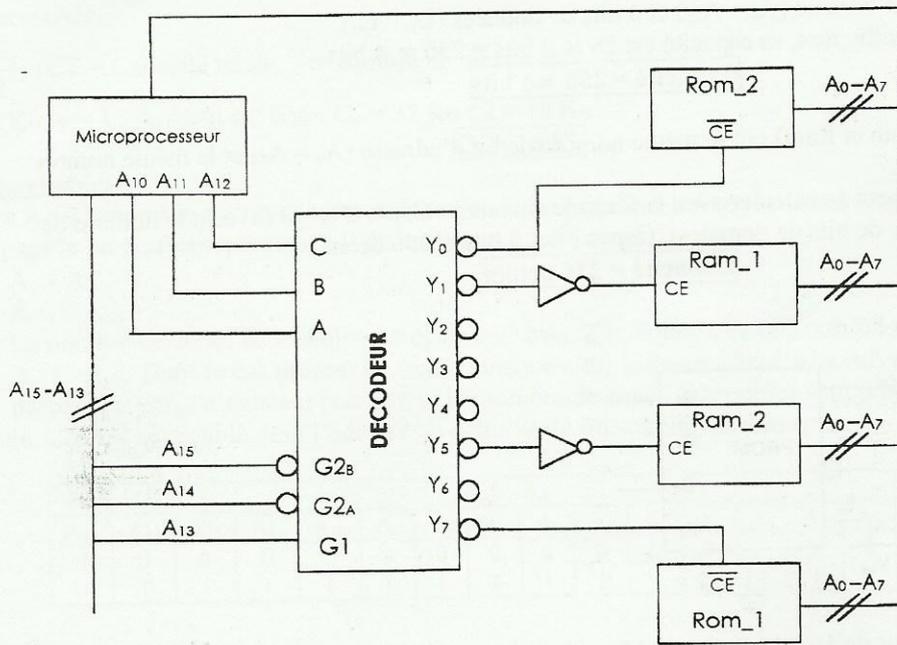
<b>0000 - 1AFF:</b>	<b>Libre</b>
<b>1B00 - 1FFF:</b>	<b>Rom_2</b>
<b>2000 - 3CFF:</b>	<b>Ram_1</b>
<b>3D00 - B0FF:</b>	<b>Libre</b>
<b>B100 - E9FF:</b>	<b>Ram_2</b>
<b>EA00 - FDFE:</b>	<b>Libre</b>
<b>FE00 - FEFF:</b>	<b>Rom_1</b>
<b>FF00 - FFFF:</b>	<b>Libre</b>

Contenu de la Prom

Les Rom sont actives à l'état bas ( $D_2 = 0$  et  $D_3 = 0$ ). Les Ram sont actives à l'état haut ( $D_0 = 1$  et  $D_1 = 1$ ).

Adresse $\mu P$	Adresse Prom	Contenu				Circuit
		$D_3$	$D_2$	$D_1$	$D_0$	
0000 - 1AFF	00 - 1A	1	1	0	0	Libre
1B00 - 1FFF	1B - 1F	0	1	0	0	Rom_2
2000 - 3CFF	20 - 3C	1	1	0	1	Ram_1
3D00 - B0FF	3D - B0	1	1	0	0	Libre
B100 - E9FF	B1 - E9	1	1	1	0	Ram_2
EA00 - FDFE	EA - FD	1	1	0	0	Libre
FE00 - FEFF	FE	1	0	0	0	Rom_1
FF00 - FFFF	FF	1	1	0	0	Libre

B/  
 1°) Schéma de câblage

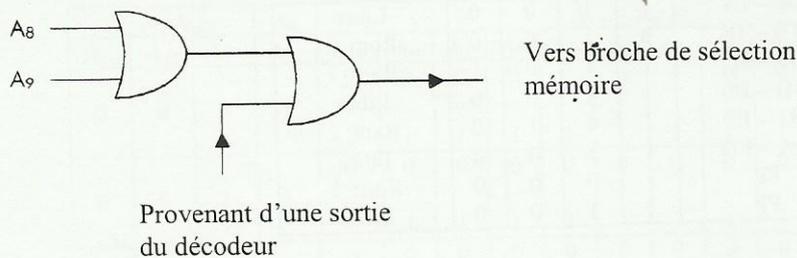


2°) Le décodeur ne permet pas un décodage complet, car certains bits d'adresses sont indéterminés. En effet, les bits  $A_{15}$  à  $A_{13}$  sont utilisés pour le décodeur, les bits  $A_0 - A_7$  par les blocs mémoires, mais les bits  $A_8$  et  $A_9$  ne sont pas utilisés et peuvent donc prendre n'importe quel état logique. Ce qui a pour effet de générer des adresses images, rendant le décodage partiel.

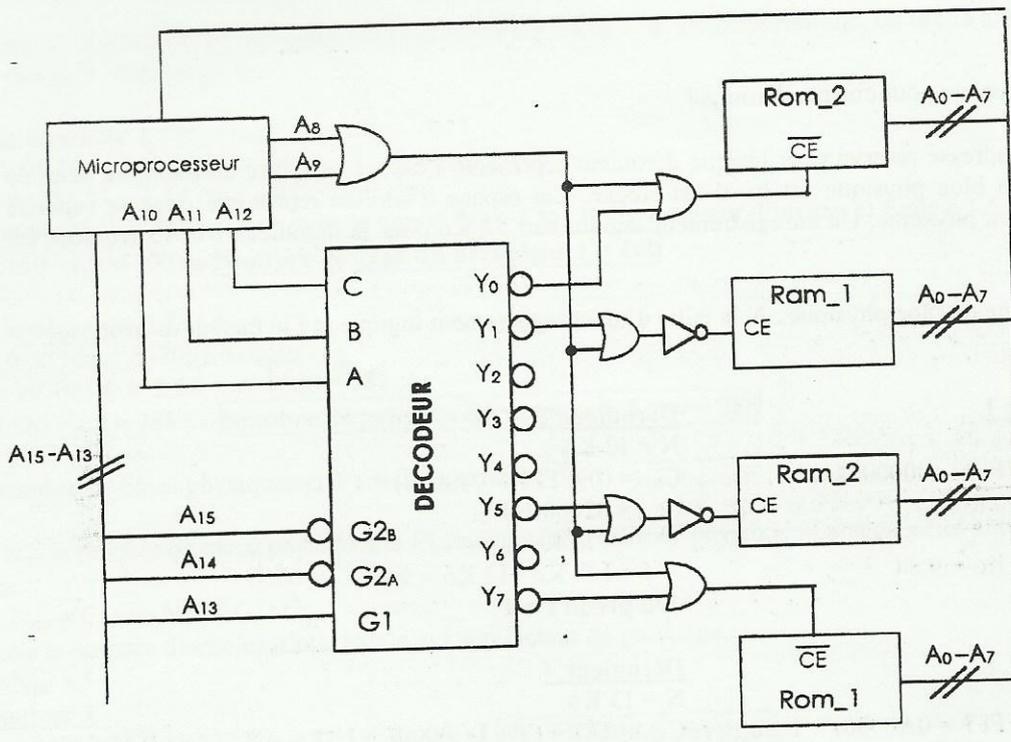
La table des adresses se présente sous la forme suivante :

$A_{15}$	$A_{14}$	$A_{13}$	$A_{12}$	$A_{11}$	$A_{10}$	$A_9$	$A_8$	$A_7$	$A_6$	$A_5$	$A_4$	$A_3$	$A_2$	$A_1$	$A_0$	Mémoires
0	0	1	0	0	0	X	X	0	0	0	0	0	0	0	0	ROM_2
0	0	1	0	0	1	X	X	0	0	0	0	0	0	0	0	RAM_1
0	0	1	1	0	1	X	X	0	0	0	0	0	0	0	0	RAM_2
0	0	1	1	1	1	X	X	0	0	0	0	0	0	0	0	ROM_1

Pour réaliser un décodage complet, il faut éliminer les adresses images en choisissant de fixer les bits  $A_8$  et  $A_9$  à 0 grâce au circuit suivant :



Le schéma de réalisation est le suivant :



**LES MEMOIRES DE MASSE**

**EXERCICE 1**

1°) Facteur de groupage pour chaque dérouleur

Chaque espace d'adresse réservé pour chaque dérouleur représente l'espace mémoire en mémoire centrale où toutefois qu'un bloc physique est lu, il est stocké. Cet espace d'adresse représente donc la capacité maximale d'un bloc physique. Un enregistrement logique fait 52 Ko pour le dérouleur 1 et 13 Ko pour les autres.

Soit  $C_{BP}$  la capacité du bloc physique ::  $N$  la taille d'un enregistrement logique et  $f$  le facteur de groupage.  
 $f = C_{BP}/N$

**Dérouleur 1**

$N = 52 \text{ Ko}$

$C_{BP} = (07FFFF - 000000) + 1$   
 $= 080000 \text{ H}$   
 $= 512$

$f = 512 \text{ Ko} / 52 \text{ Ko} = 9,84$

**On prend  $f = 9$**

**Dérouleur 2**

$N = 13 \text{ Ko}$

$C_{BP} = (09FFFF - 080000) + 1$   
 $= 020000 \text{ H}$   
 $= 128 \text{ Ko}$

$f = 128 \text{ Ko} / 13 \text{ Ko} = 9,84$

**On prend  $f = 9$**

**Dérouleur 3**

$N = 13 \text{ Ko}$

$C_{BP} = (0BFFFF - 0A0000) + 1$   
 $= 020000 \text{ H}$   
 $= 128 \text{ Ko}$

$f = 128 \text{ Ko} / 13 \text{ Ko} = 9,84$

**On prend  $f = 9$**

**Dérouleur 4**

$N = 13 \text{ Ko}$

$C_{BP} = (0DFFFF - 0C0000) + 1$   
 $= 020000 \text{ H}$   
 $= 128 \text{ Ko}$

$f = 128 \text{ Ko} / 13 \text{ Ko} = 9,84$

**On prend  $f = 9$**

2°) Capacité théorique et pratique

**Dérouleur 1**

Capacité théorique:  $C_t$

$C_t = L \times d$  ( $L$  = longueur de la bande et  $d$  = densité)

$C_t = 73\,000 \text{ cm} \times (1600 \text{ octets} / 2,54 \text{ cm}) = 43,85 \text{ Mo} \Rightarrow C_t = 43,85 \text{ Mo}$

Remarque: Cette capacité théorique est la même pour tous les autres dérouleurs

**Capacité pratique :  $C_p$**

Longueur d'un bloc physique :  $l_1 = f \times N / d = 9 \times 52 \times 1024 \times 2,54 \text{ cm} / 1600$

$l_1 = 760,8 \text{ cm}$

$C_p = L / (lg + 1) \times f \times N = 73000 / (760,8 + 1,5) \times 9 \times 52 \text{ Ko} = 45892609 \text{ caractères} \Rightarrow C_p = 43,76 \text{ Mo}$

**Dérouleur 2**

Longueur d'un bloc physique :  $l_2 = 9 \times 13 \times 1024 \times 2,54 \text{ cm} / 1600 \Rightarrow l_2 = 190,2 \text{ cm}$

**Capacité pratique**

$C_p = L / (lg + 1) \times f \times N = 73000 / (1,5 + 190,2) \times 9 \times 13 \text{ Ko} = 45623286 \text{ caractères} \Rightarrow C_p = 43,5 \text{ Mo}$

Remarque : Cette capacité est identique à celle des dérouleurs 3 et 4.

3°) Nombre de Gaps approximatifs de la bande

Si l'espace occupé par les gaps ( $C_g$ ) avait été exploitable pour stocker des données, cet espace donnerait une capacité égale à la capacité théorique dont on soustrait la capacité pratique.

$C_g = C_t - C_p$

Pour le dérouleur 1,  $C_g = 43,85 \text{ Mo} - 43,76 \text{ Mo} = 0,09 \text{ Mo}$

Pour le dérouleur 2, 3 et 4,  $C_g = 43,85 \text{ Mo} - 43,5 \text{ Mo} = 0,35 \text{ Mo}$

Soit  $L_{gt}$ , la longueur totale des gaps. On peut écrire  $C_g = L_{gt} \times d$ . De cette formule, on tire la longueur totale des gaps.  $L_{gt} = C_g / d$ .

Pour le dérouleur 1 :

$$C_g = 0,09 \text{ Mo} = 94371,84 \text{ octets}$$

$$L_{gt} = 94371,84 \times 2,54 \text{ cm} / 1600 = 150 \text{ cm.}$$

On appelle  $N_g$  le nombre de gaps de la bande.  $N_g = L_{gt} / l_g$  ( $l_g$  longueur d'un gap et égale à 1,5 cm).

$$N_g = 150 / 1,5 = 100 \Rightarrow \boxed{\text{Nombre de gaps du dérouleur 1} = 100}$$

Pour le dérouleur 2,3 et 4 :

$$C_g = 0,35 \text{ Mo} = 367001,6 \text{ octets}$$

$$L_{gt} = 367001,6 \times 2,54 \text{ cm} / 1600 = 582 \text{ cm}$$

$$N_g = 582 / 1,5 = 388 \Rightarrow \boxed{\text{Nombre de gaps des dérouleurs 2, 3 et 4} = 388}$$

4°) Nombre de blocs physiques et d'articles par bande

Soit  $N_{BP}$  le nombre de blocs physiques et  $C_{BP}$  la capacité d'un bloc physique :  $C_p$  la capacité pratique d'une bande.

$$C_p = N_{BP} \times C_{BP} \Rightarrow N_{BP} = C_p / C_{BP}$$

Soit  $N_a$  le nombre d'articles d'une bande et  $f$  son facteur de groupage.

$$N_a = N_{BP} \times f$$

**Dérouleur 1**

$$M_{BP} = 44810,24 \text{ Ko} / 512 \text{ Ko} = 87,52 ; \text{ on prend } \boxed{N_{BP} = 88 \text{ blocs physique}}$$

$$N_a = 88 \times 9 \Rightarrow \boxed{N_a = 792 \text{ articles}}$$

**Dérouleur 2**

$$N_{BP} = 44544 \text{ Ko} / 128 \text{ Ko} = 348$$

$$\boxed{N_{BP} = 348 \text{ blocs}}$$

$$N_a = 348 \times 9$$

$$\boxed{N_a = 3192 \text{ articles}}$$

Ce résultat est identique à celui des dérouleurs 3 et 4.

## EXERCICE 2

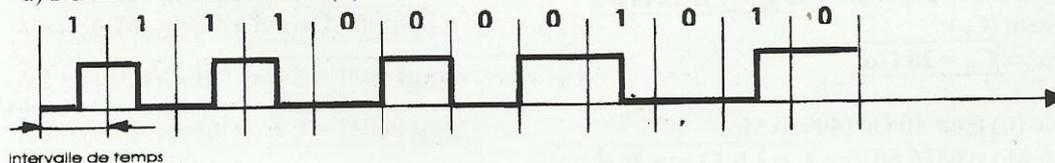
1°) MFM : Modified Frequency Modulation (Modulation de Fréquence Modifiée)

RLL : Run Length Limited (code à longueur limitée)

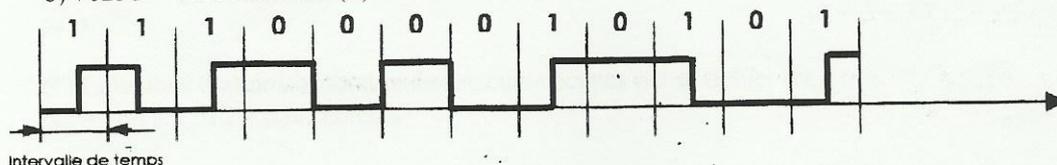
ARLL : Advanced Run Length (code à longueur limitée amélioré)

2°)

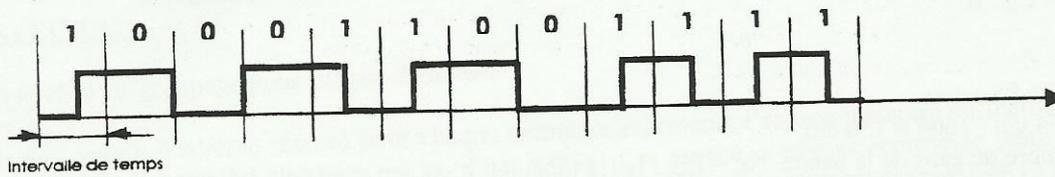
a) FOA16 = 111100001010(2)



b) 70258 = 111000010101(2)



c) 225510 = 100011001111(2)



### EXERCICE 3

1°) Le schéma de la figure représente le mode RAID 0 ou Stripping. En effet, chaque fragment du même fichier est écrit sur un disque différent.

2°) Chaque disque est un disque UDMA 100. Cela signifie que le taux de transfert de chaque disque est de 100 Mo/s.

3°) En mode RAID 0, pour 4 disques, le quart de la taille du fichier est écrit sur chaque disque. L'enregistrement du fichier se fait donc 4 fois plus rapidement que s'il avait été enregistré sur un seul disque. Chaque disque a un débit de 100 Mo/s. Pour enregistrer un fichier de 96 Go à ce débit, il faut mettre le temps suivant : Soit X ce temps.

Si le transfert de 100 Mo nécessite 1 seconde, pour enregistrer 96 Go, il faut mettre un temps  $X_0$  :  
 $X_0 = 96 \text{ Go} \times 1 \text{ s} / 100 \text{ Mo} = 983,04 \text{ secondes}$ . En mode RAID 0 et pour 4 disques,  $X = X_0 / 4 = 983,04 / 4$   
 $X = 245,76 \text{ s}$  donc **Temps = 4 mn 5 s**

En mode RAID 1 :

Le fichier doit se retrouver intégralement sur chaque des 4 disques. Or un disque fait 40 Go. Le fichier de 96 Go ne pourra pas tenir sur un disque, le disque arrivant à saturation, seulement 40 Go du fichier pourront être enregistrés. Cela représente un temps égale à :  
 $X = 40 \text{ Go} \times 1 \text{ s} / 100 \text{ Mo} = 409,6 \text{ s}$ . Cela donne un temps de **6 mn 50 s**

4°) Que l'on soit en mode RAID 0 ou en mode RAID 1, la sauvegarde sur la bande se fait de la façon suivante :

- les données sont lues sur les disques et chargées en mémoire centrale de l'ordinateur.
- De la mémoire centrale, les données sont transférées vers la bande avec un débit de 6 Mo/s.

Seulement, en mode RAID 1, les 4 disques de 40 Go ont le même contenu. La sauvegarde du contenu d'un disque est le même que pour les autres. On sauvegarde donc 40 Go une seule fois.

En mode RAID 0, les 4 disques formant un seul disque, ne contiennent pas les mêmes informations. C'est l'association des 4 contenus qui forment la totalité des informations à sauvegarder. On sauvegarde donc  $40 \text{ Go} \times 4 = 160 \text{ Go} = 163 \text{ 840 Mo}$ .

#### Mode RAID 0

\* Temps de sauvegarde ( $t_s$ ) :

$$t_s = 163 \text{ 840 Mo} \times 1 \text{ s} / 6 \text{ Mo} = 27306,66 \text{ s} \Rightarrow \mathbf{t_s = 7 \text{ h } 35 \text{ m } 6 \text{ s}}$$

\* Espace de bande restant ( $C_R$ ):

$$C_R = 180 \text{ Go} - 160 \text{ Go} \Rightarrow \mathbf{C_R = 20 \text{ Go}}$$

#### Mode RAID 1

\* Temps de sauvegarde ( $t_s$ ) pour 40 Go (40960 Mo)

$$t_s = 40960 \text{ Mo} \times 1 \text{ s} / 6 \text{ Mo} = 6826,66 \text{ s} \Rightarrow \mathbf{t_s = 1 \text{ h } 53 \text{ mn } 46 \text{ s}}$$

\* Espace de bande restant ( $C_R$ ):  $C_R = 180 \text{ Go} - 40 \text{ Go} = 140 \text{ Go} \Rightarrow \mathbf{C_R = 140 \text{ Go}}$

### EXERCICE 4

1°) Définition du facteur d'entrelacement

Si le contrôleur d'un disque permet de lire un seul secteur sur X secteurs contigus, on dit que le disque a un facteur d'entrelacement de 1 : X. En d'autres termes, le facteur d'entrelacement représente le nombre de tours de disque nécessaire pour lire une piste entière.

2°)

Disque 1 : facteur d'entrelacement = 1 : 1  
 Disque 2 : facteur d'entrelacement = 1 : 3  
 Disque 3 : facteur d'entrelacement = 1 : 6

3°)

Disque 1

Un tour du disque suffit pour lire une piste entière. Le disque tourne à 3600 tours par minute (60 secondes). On cherche donc le temps mis pour faire un tour du disque. Soit  $t_1$  ce temps.  $t_1 = 60/3600 \Rightarrow t_1 = 16,67 \text{ ms}$ .

Disque 2

Il faut 3 tours pour lire une piste entière :  $t_2 = (60 / 3600) \times 3 = 0,05 \Rightarrow t_2 = 50 \text{ ms}$

Disque 3

Il faut 6 tours pour lire une piste entière :  $t_3 = (60 / 3600) \times 6 = 0,1 \Rightarrow t_3 = 100 \text{ ms}$

4°) Si la lecture d'une piste entière met Y ms, calculer la vitesse de rotation revient à chercher le nombre de tours possibles en une minute.

Disque 1

La lecture d'une piste entière nécessite un tour et met Y ms. Si on cherche le nombre de tours fait en 1 mn, ce sera la vitesse du disque.

On convertit Y ms en Y mn.

$$Y \text{ ms} = Y \cdot 10^{-3} \text{ s} = (Y/30) \cdot 10^{-3} \text{ mn} = (Y/6) \cdot 10^{-1} \text{ mn} \Rightarrow Y \text{ ms} = (Y/6) \cdot 10^{-4} \text{ mn}$$

La vitesse de rotation V1 du disque vaut alors :

$$V1 = 1 / ((Y/6) \cdot 10^{-4}) \Rightarrow V1 = (6 \cdot 10^4 / Y) \text{ tpm}$$

Disque 2

La lecture d'une piste entière met 3 tours

$$V2 = 3 / ((Y/6) \cdot 10^{-4}) \Rightarrow V2 = (18 \cdot 10^4 / Y) \text{ tpm}$$

Disque 3

La lecture d'une piste entière met 6 tours

$$V3 = 6 / ((6/Y) \cdot 10^{-4}) \Rightarrow V3 = (36 \cdot 10^4 / Y) \text{ tpm}$$

5°) Application numérique

Pour Y = 24 ms

$$V1 = (6/24) \times 10^4 \Rightarrow V1 = 2500 \text{ tpm}$$

$$V2 = (18/24) \times 10^4 \Rightarrow V2 = 7500 \text{ tpm}$$

$$V3 = (36/24) \times 10^4 \Rightarrow V3 = 15000 \text{ tpm}$$

En conclusion, plus le facteur d'entrelacement est élevé, il faut aussi que la vitesse du disque soit grande pour garder le même temps de lecture d'une piste entière avec les disques ayant un facteur d'entrelacement petit.

6°) Calculons d'abord le nombre de secteurs occupés par le fichier ( $N_{sf}$ ) :

$$N_{sf} = 168 \text{ K} / 512 = 336 \text{ secteurs}$$

Le disque 2 a un facteur d'entrelacement de 1 : 3. Il faut donc 3 tours pour lire une piste entière. Calculons le nombre de pistes occupées par le fichier ( $N_{pf}$ ) :

$$N_{pf} = 336 / 63 = (5 + 1/3)$$

Le fichier occupe donc 5 pistes entières plus le tiers d'une piste.

Pour lire le fichier, il faut comme nombre de tours ( $N_f$ ) :

$$N_f = 3 \times (5 + 1/3) = 15 + 1 = 16 \text{ tours. ; } \boxed{N_f = 16 \text{ tours}}$$

Le temps mis pour effectuer la lecture du fichier en 16 tours est le suivant :

$$t = (60 / 3600) \times 16 = 0,266 \text{ s} \Rightarrow \boxed{t = 266 \text{ ms}}$$

### EXERCICE 5

1°) Nombre de secteurs par cluster ( $N_s$ ) :

Le disque possède 512 Ko par secteur. Or un secteur fait 512 octets. Le nombre de secteurs par cluster vaut :

$$N_s = 512 \text{ Ko} / 512 = 1024 \Rightarrow \boxed{N_s = 1024}$$

2°) Capacité disque perdu ( $C_{dp}$ )

La taille du fichier fait 1,5 Ko. Ce fichier peut donc occuper au maximum 1,5 Ko / 512 octets secteurs, soit 3 secteurs.

Un secteurs fait 1024 secteurs. Ce fichier tient donc dans un seul cluster. L'espace disque perdu vaut :

$$C_{dp} = 1024 - 3 = 1021 \text{ secteurs avec } 1 \text{ secteur} = 512 \text{ octets}$$

$$C_{dp} = 1021 \times 512 = 510,5 \text{ Ko} \Rightarrow \boxed{C_{dp} = 510,5 \text{ Ko}}$$

Cette capacité peut être utilisable à condition que l'on augmente la taille du fichier qui pourra ainsi utiliser cet espace. En effet, aucun autre fichier que celui qui l'a occupé ne peut utiliser un cluster déjà occupé par un autre fichier.

3°) Le fichier tient dans un seul cluster. Si on détermine le nombre de clusters du disque, ce nombre correspondra bien à P, nombre de fois que l'on pourra dupliquer le fichier pour amener le disque à saturation.

Le disque a une capacité de 84 Go avec 512 Ko/cluster = 0,5 Mo/cluster. Soit Ncl le nombre de cluster du disque.

$$N_{cl} = 84 \times 1024 \text{ Mo} / 0,5 \text{ Mo} \Rightarrow \boxed{N_{cl} = 172032 \text{ clusters}}$$

Les clusters 0 et 1 d'un disque sont toujours occupés par la zone système du disque. La zone de données commence au cluster  $n^{\circ}2$ .

Il faut donc soustraire 2 clusters des 172032.  $\boxed{P = 172030}$

Cette duplication n'est possible que si à chaque copie, on donne un nouveau nom au fichier.

4°) L'espace disque perdu quand le fichier a été enregistré 1 fois vaut 510,5 Ko

S'il est dupliqué 172030 fois, la capacité totale perdue ( $C_{TP}$ ) vaut :

$$C_{TP} = 510,5 \text{ Ko} \times 172030 = 87821315 \text{ Ko} \quad \boxed{C_{TP} = 83,75 \text{ Go}}$$

La quasi totalité de l'espace disque est perdu. C'est en effet l'inconvénient de mettre sur un disque de grande taille dont les clusters sont volumineux des fichiers de petites tailles.

### EXERCICE 6

1°) Une entrée du répertoire racine tient sur 32 octets. On peut donc lire sur la figure 1, 9 entrées pour le répertoire racines.

Un secteur fait 512 octets. Le secteur 19 peut donc contenir  $512 / 32 = 16$  entrées.

L'offset de départ de la dernière entrée est : 000001E0H

2°) Les fichiers effacés ont leur premier octet de l'entrée du répertoire racine code E5. D'après la figure 2, il existe 4 fichiers dont 2 ne commencent pas par E5. Il y a donc 2 fichiers qui ne sont pas effacés.

Caractéristiques de ces fichiers :

Ces fichiers occupent les entrées 2 et 4 du répertoire racine.

**a) Nom + extension (octet 0 à 10)**

D'après la figure 2. le nom occupe les 8 premiers octets et l'extension les 3 octets suivants d'une entrée du répertoire racine. Sur la figure 1, on dit :

Fichier 1 : 41 57 41 44 4A 49 20 20 – 58 4C 53  
 A W A D J I X L S

**Nom du fichier : AWADJI.XLS** Ce fichier a été créé sous MS Excel.

Fichier 2 :

On lit : 42 54 53 20 20 20 20 20 – 44 4F 43  
 B T S D O C

**Nom du fichier 2 : BTS.DOC** Ce fichier a été créé sous MS WINWORD.

**b) Attribut (octet 11)**

**Fichier 1**

Octet 11 = 20 H = 0010000<sub>(2)</sub>  
 Le bit 5 vaut 1. Le fichier 1 est un fichier archive d'après la figure 2.

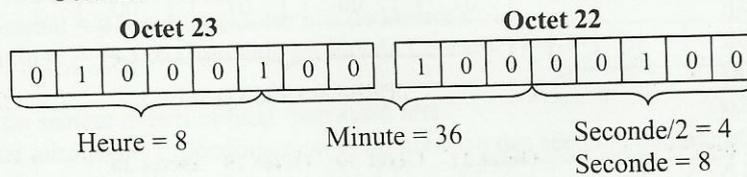
**Fichier 2**

Octet 11 = 22 H = 00100010<sub>(2)</sub>  
 Les bits 1 et 5 valent 1. Le fichier 2 est un archive caché et archive d'après la figure 2.

**c) La date et l'heure de création (Octet 22 et octet 23)**

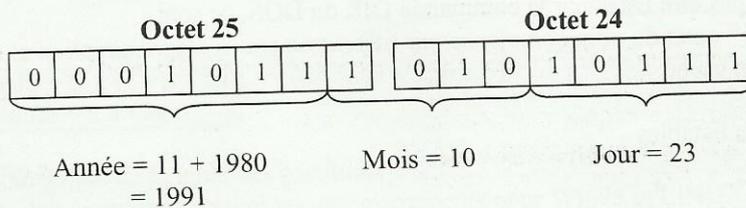
**Fichier 1**

Heure de création : Octet 22 = 84 H  
 Octet 23 = 44 H



**Heure de création du fichier 1 : 8h 36 mn 8s**

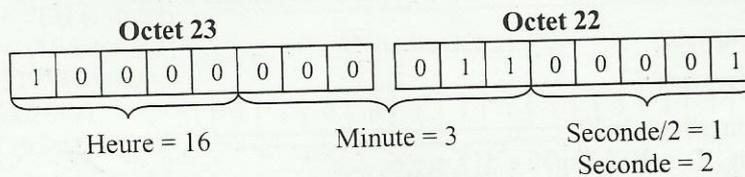
Date de création : Octet 24 = 57  
 Octet 25 = 17



**Date de création du fichier 1 : 23 octobre 1991**

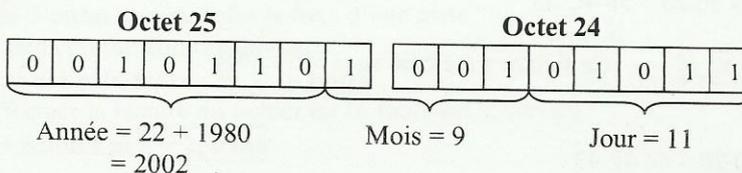
**Fichier 2**

Heure de création : Octet 22 = 61H  
 Octet 23 = 80H



**Heure de création du fichier 2 : 16 h 3 mn 2s**

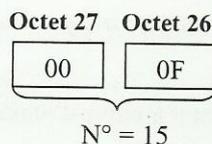
Date de création :      Octet 24 = 2BH  
                                  Octet 25 = 2DH



**Date de création du fichier 2 : 11 septembre 2002**

Le numéro du premier cluster (octet 26 et octet 27)

Fichier 1      Octet 26 = 0F  
                          Octet 27 = 00



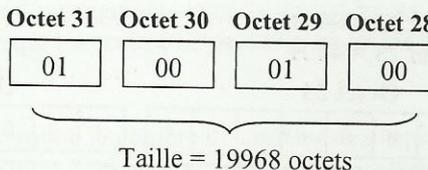
**Numéro du premier cluster du fichier 1 = 15**

Fichier 2      Octet 26 = 00  
                          Octet 27 = 01

**Numéro du premier cluster du fichier 2 = 256**

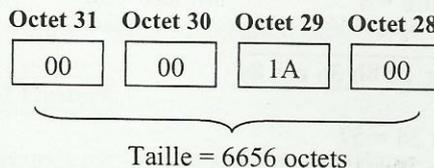
**\* Taille des fichiers (Octets 28, 29, 30 et 31)**

Fichier 1 :      Octet 28 = 00  
                          Octet 29 = 4E  
                          Octet 30 = 00  
                          Octet 31 = 00



**Taille du fichier 1 = 19,5 Ko**

Fichier 2 :      Octet 28 = 00  
                          Octet 29 = 1A  
                          Octet 30 = 00  
                          Octet 31 = 00



**Taille du fichier 2 = 6,5 Ko**

3°) Les fichiers qui ne peuvent pas être listés par la commande DIR du DOS, ce sont :

- les fichiers effacés (le 1<sup>er</sup> octet du nom commence par ES)
- le fichier ayant l'attribut caché

Offset de départ des fichiers non listables

1<sup>er</sup> fichier : 00000000 H

2<sup>ème</sup> fichier : 00000040 H

3<sup>ème</sup> fichier : 00000060 H

4°) Le fichier commence au cluster n°15. Or le vidage de la FAT donne un fichier dont le chaînage commence au cluster n°15.

Le chaînage appartient donc au fichier 1.

5°) Taille d'un cluster (T<sub>C</sub>)

Le fichier a une taille de 19,5 Ko.

Le nombre de clusters du chaînage est de 39.

La taille d'un cluster (T<sub>C</sub>) vaut ; T<sub>C</sub> = 19,5 Ko/39 = 512 octets.

Un secteur fait 512 octets. Il y a donc un secteur par cluster.

**Conclusion : 1 secteur = 1 cluster = 512 octets**

6°) C'est seulement sur les disquettes que l'on a 1 secteur par cluster. Le support sur lequel le fichier est enregistré ne peut être qu'une disquette.

**EXERCICE 7**

1°) Numéro du cylindre de départ de la partition LINUX

La partition DOS fait 320 cylindres (0 à 319). X représente le numéro du cylindre de départ de la partition LINUX. X vaut 320. La partition LINUX occupe 576 cylindres allant de 320 à 895.

2°) Taille de chaque partition

**Partition DOS**

Nombre de cylindre (C) = 320 (0 à 319) Capacité = C x H x S x 512 = 320 x 126 x 63 x 512

Nombre de têtes (H) = 126 (1 à 126)

Nombre de secteurs (S) = 63 (1 à 63) ⇒ **Capacité = 1,21 Go**

**Partition LINUX**

Nombre de cylindres = 576 (320 à 895) Capacité = C x H x S x 512 = 576 x 127 x 63 x 512

Nombre de têtes = 127 (0 à 126)

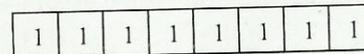
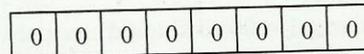
Nombre de secteurs = 63 (1 à 63) ⇒ **Capacité = 2,2 Go**

3-a) En consultant la figure 2, on constate que :

- Le nombre de tête de départ et final tient sur un octet

N° de tête départ minimal = 0 (on met tous les bits de l'octet à 0).

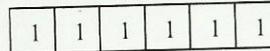
N° de tête finale maximal = 255 (on met tous les bits de l'octet à 1).



- Le nombre du secteur départ et final tient sur 6 bits

N° secteur départ minimal = 1 (généralement la numérotation des secteurs commence à 1 et non à 0)

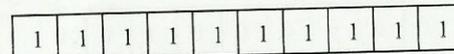
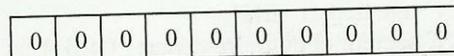
N° secteur final maximal = 63 (on met les 6 bits à 1)



- Le numéro du cylindre départ et final tient sur 10 bits

N° cylindre départ minimal = 0 (on met les 10 bits à 0)

N° cylindre final maximal = 1023 (on met les 10 bits à 1)



On peut déduire la taille maximale d'une partition :

Partition maximale = C x H x S x 512 = 1024 x 63 x 256 x 512

**Partition maximale = 8,4 Go**

3-b) Vidage hexadécimal de la table des partitions

- la table des partitions contient les enregistrements pour Win98 et LINUX.

- le système démarre sous Win98. C'est donc Win98 qui est la partition active (c'est une partition principale de type FAT 32).

**Partition Win98 :**

Tête départ = 1 = 01H (1 octet)

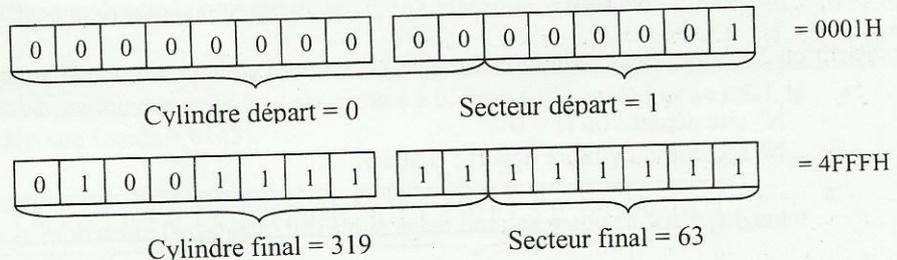
Tête finale = 126 = 7EH (1 octet)

Cylindre départ = 0 (10 bits)

Cylindre finale = 319 (10 bits)

Secteur départ = 1 (6 bits)

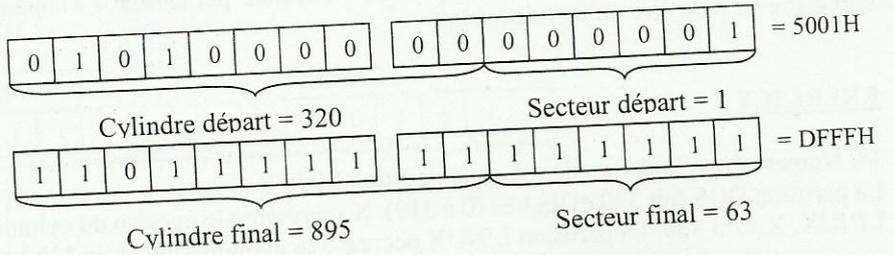
Secteur final = 63 (6 bits)



ID system (1 octet = 0B (partition principale de type FAT 32 créée par Win98))

**Partition LINUX**

- Tête départ (1 octet) = 0 (00H)
- Tête finale (1 octet) = 126 (7EH)
- Cylindre départ (10 bits) = 320
- Cylindre final (10 bits) = 895
- Secteur départ (6 bits) = 1
- Secteur final (6 bits) = 63



ID system (1 octet) = 08 (partition Non DOS)  
 En définitive, le vidage Hexadécimal de la table des partitions sera le suivant :

**80 01 00 01 0B 7E 4F FF**  
**00 00 50 01 08 7E DF FF**

4°) Offset de fin de la table des partitions précédente

Cette table tient sur 16 octets, le premier octet ayant l'offset 1BE, l'offset final est donc 1CD.

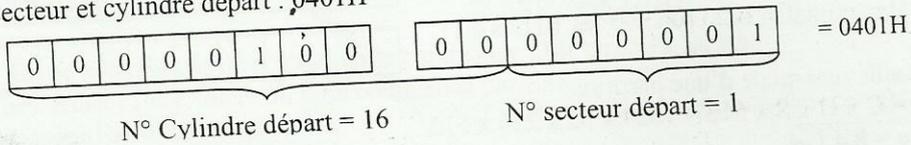
5-a) Les enregistrements concernant une partition tiennent sur 64 bits (8 octets). Le vidage présenté contient 16 octets. Il y a donc 2 partitions qui existent sur le disque.  
 Les octets 1 et 5 des enregistrements de chaque partition précisent si la partition est active ou non et donne le type de partition.

**Partition 1** : Octet 1 = 00 (il s'agit d'une partition non système)  
 Octet 5 = 0B (il s'agit d'une partition de type FAT 32 (Win9X))

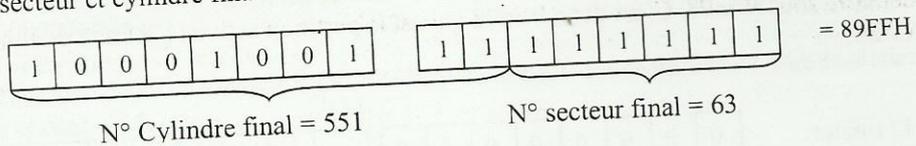
**Partition 2** : Octet 1 = 80 (il s'agit d'une partition système)  
 Octet 5 = 07 (il s'agit d'une partition de type NTFS : Windows NT ou Windows 2000)

5-b) Partition 1

- N° tête départ : 01H = 1
- N° secteur et cylindre départ : 0401H

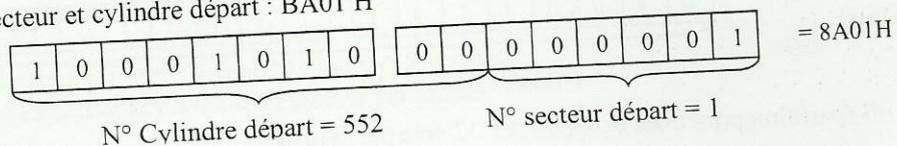


- N° tête finale : 05H = 5
- N° secteur et cylindre final = 89FF



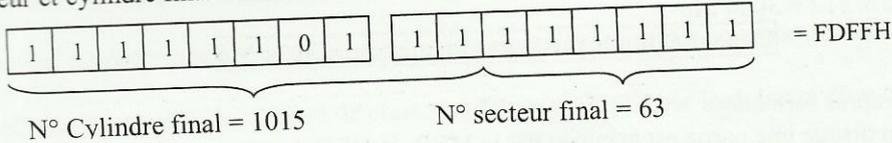
Partition 2

- N° tête départ : 00 H = 0
- N° secteur et cylindre départ : BA01 H



- N° tête final : 05H = 5

- N° secteur et cylindre final : FDFH



5-c) Le système qui démarre à la mise sous tension du PC est le Windows 2000 (ou Windows NT), car c'est la deuxième identifiée par ID System = 07 qui est la partition active.

6°) Taille des partitions

Partition 1

$\left. \begin{array}{l} \text{N}^\circ \text{ tête de départ} = 1 \\ \text{N}^\circ \text{ tête finale} = 5 \end{array} \right\} \Rightarrow \text{nombre de secteurs} = 5$

$\left. \begin{array}{l} \text{N}^\circ \text{ secteur départ} = 1 \\ \text{N}^\circ \text{ secteur final} = 63 \end{array} \right\} \Rightarrow \text{nombre de secteurs} = 63$

$\left. \begin{array}{l} \text{N}^\circ \text{ cylindre départ} = 16 \\ \text{N}^\circ \text{ cylindre final} = 551 \end{array} \right\} \Rightarrow \text{nombre de secteurs} = 536$

Taille de la partition 1 =  $C \times H \times S \times 512 = 536 \times 5 \times 63 \times 512 = 82,4 \text{ Mo}$

**Taille de la partition 1 = 82,4 Mo**

Partition 2

$\left. \begin{array}{l} \text{N}^\circ \text{ tête de départ} = 0 \\ \text{N}^\circ \text{ tête finale} = 5 \end{array} \right\} \Rightarrow \text{nombre de secteurs} = 6$

$\left. \begin{array}{l} \text{N}^\circ \text{ secteur départ} = 1 \\ \text{N}^\circ \text{ secteur final} = 63 \end{array} \right\} \Rightarrow \text{nombre de secteurs} = 63$

$\left. \begin{array}{l} \text{N}^\circ \text{ cylindre départ} = 552 \\ \text{N}^\circ \text{ cylindre final} = 1015 \end{array} \right\} \Rightarrow \text{nombre de secteurs} = 464$

Taille de la partition 2 =  $C \times H \times S \times 512 = 464 \times 6 \times 63 \times 512 = 85,6 \text{ Mo}$

**Taille de la partition 2 = 85,6 Mo**

**EXERCICE 8**

1°) Le paramètre Landz représente la piste d'atterrissage. Cette piste est la dernière piste située vers le centre du disque. Elle ne contient aucune donnée et c'est au dessus d'elle que les têtes de lecture / écriture seront parquées à l'arrêt du disque.

Quand on écrit Landz = 6143, ce chiffre représente le numéro de la dernière piste ou du dernier cylindre du disque.

Le paramètre cylindre = 6144 représente le nombre total de cylindres. Le paramètre Landz représente le numéro du dernier cylindre. Or la numérotation des cylindres commence à 0. Pour 6144 cylindres au total, la numérotation ira de 0 à 6143 (d'où découle Landz = 6143).

2°) Capacité du disque avant formatage

La capacité du disque avant formatage représente la capacité théorique et est donnée par la formule suivant :

$$C_1 = C \times H \times S \times \text{Capacité d'un secteur}$$

Avec : C = nombre de cylindres

H = nombres de têtes

S = nombre de secteurs par piste  
 $C_t = 6144 \times 16 \times 63 \times 512 = 3024 \text{ Mo}$

**Capacité du disque avant formatage = 3024 Mo**

Capacité du disque après formatage

Après formatage du disque une partie est occupée par le MBR, le DRB, les deux copies de la FAT et le répertoire racine. Pour trouver la capacité après formatage, il faut soustraire de la capacité théorique l'espace occupé par les informations enregistrées sur le disque après formatage.

- Le MBR occupe un secteur, soit 512 octets = 0,5 Ko
- L'amorce (DBR) occupe 5 secteurs, soit 2560 octets = 2,5 Ko
- Une copie de la FAT occupe :
  - Pour une FAT 16, une entrée de la FAT est sur 16 bits, soit 2 octets.
  - Pour un disque de 3024 Mo et 16 secteurs par cluster, on aura : 387072 clusters, soit 387072 entrée dans la FAT.
  - Une entrée occupant 2 octets, la première copie de la FAT occupera 774144 octets pour les 387072 entrées ; soit 756 Ko. Les 2 copies de la FAT occuperont 1512 Ko.
- Le répertoire racine occupe  $2043 \times 32 \text{ octets} = 65536 \text{ octets}$  ; soit 64 Ko

L'espace occupé par la zone système est la suivante :  $(0,5 \text{ Ko} + 2,5 \text{ Ko} + 1512 \text{ Ko} + 64 \text{ Ko}) = 1579 \text{ Ko}$

Espace occupé par la zone système = 1,54 Mo

La capacité du disque après formatage est  $(3024 \text{ Mo} - 1,54 \text{ Mo})$  ; soit 3022,46 Mo.

### EXERCICE 9

1°) Capacité théorique

$$C_t = L \times d$$

Avec : L = longueur de la bande en cm

d = densité exprimée en caractères ou octets/cm

$$L = 730 \text{ m} = 73000 \text{ cm}, d = 1260 \times 4 \text{ bits / cm} = 630 \text{ octets / cm}$$

$$C_t = 73000 \times 630 \text{ octets} \Rightarrow C_t = 43,86 \text{ Mo}$$

Capacité pratique

$$C_p = L / (1 + lg) \times f \times N$$

Avec : lg = 1,5 cm

f = 5 articles par bloc

N = 150 caractères par article

On cherche l que l'on ne connaît pas.

$$l = f \times N / d = 5 \times 150 / 630 = 1,2 \text{ cm}$$

$$C_p = 73000 / (1,2 + 1,5) \times 5 \times 150 \Rightarrow C_p = 19,33 \text{ Mo}$$

2°) Le facteur de groupage vaut maintenant 8

La capacité théorique ne change pas, car le facteur de groupage n'influe pas sur la capacité théorique.

$$C_t = 43,86 \text{ Mo}$$

Capacité pratique

Longueur d'un bloc physique :

$$l = f \times N / d = 8 \times 150 / 630 = 1,9 \text{ cm}$$

$$C_p = 73000 / (1,9 + 1,5) \times 8 \times 150 \Rightarrow C_p = 24,57 \text{ Mo}$$

Conclusion

Le facteur de groupage est grand, moins de gap il y a, et plus grande est la capacité pratique de la bande.

### EXERCICE 10

1°) Nombre de secteurs par piste (Nsc)

1 cluster fait 32 Ko = 32768 octets

1 secteur fait 512 octets

$$Nsc = 32768 / 512 = 64 \text{ secteurs}$$

### Un cluster fait 64 secteurs

2°) Si un seul secteur est défectueux dans un cluster, tout le cluster est considéré comme défectueux par le SE.

L'espace perdu (Ep) est donc égal au nombre de clusters affectés multiplié par la capacité d'un cluster.

Nombre de cluster affectés = 4

Ep = 4 × 32 Ko = 128 Ko

**Espace perdu = 128 Ko**

3°) On peut préconiser de diminuer le nombre de secteur par cluster. Pour les disques durs, le plus petit nombre de secteurs par cluster est de 2. Seules les disquettes acceptent 1 secteur par cluster.

Si on admet donc qu'un cluster fait 2 secteurs soit 1 Ko, pour obtenir l'espace minimum perdu, on doit admettre que deux secteurs défectueux se retrouvent dans le même cluster.

Le total de secteur défectueux vaut 17, cela donne 8 clusters dont les 2 secteurs sont défectueux et 1 cluster avec un seul secteur défectueux. L'espace perdu revient à 9 clusters à éliminer.

Cela donnera 9 × 1 Ko = 9 Ko.

**Espace minimum perdu = 9 Ko**

Pour trouver l'espace perdu maximum, on doit admettre qu'il y a un secteur défectueux par cluster.

Pour 17 secteurs défectueux, 17 clusters seront affectés, soit 17 × 1 Ko = 17 Ko.

**Espace maximum perdu = 17 Ko**

### EXERCICE 11

Donnons le chaînage de tous les fichiers

ORDRES.DAT : 10 → 41 → 43 → 44 → 45 → 46 → 47 → 48 → 49 → 50 → 100  
 MSC DEX.SYS : 02 → 03 → 29 → 30 → 80 → 81  
 CERCLE.PAS : 04  
 RACAP.XLS : 07 → 06 → 05  
 PROJET.PPT : 57 → 76 → 58 → 59 → 57  
 LISEZMOI.TXT : 62 → 63 → 64 → 65 → 66 → 67 → 68 → 69 → 70 → 71 → 72  
 MISSION.DAT : 94 → 93 → 92 → 91 → 90 → 89 → 88 → 87 → 86 → 85 → 84 → 83 → 82  
 MS TWIN.DLL : 26 → 27 → 04  
 ASSIST.COM : 08 → 09 → 10 → 11 → 12 → 13 → 14 → 15  
 DOLLAR.OVL : 37 → 36 → 35 → 34 → 33  
 ESSAI.TXT : 16 → 17 → 18 → 19 → 20 → 21 → 22 → 23 → 24 → 25 → 56 → 55 → 54 → 53  
 → 52 → 51 → 77 → 78 → 79 → 28 →

SOUTIEN.DOC : 38

En analysant chaque chaînage, on peut conclure

- le fichier PROJET.PPT a une chaîne circulaire
- les fichiers CERCLES.PAS et TWIN.DLL sont croisés

2°) La chaîne 31 → 32 → 60 → 61 est un ensemble de clusters perdus. Les clusters 73, 75 et 97 sont des clusters perdus.

L'espace disque que l'on pourrait récupérer à la suppression du contenu de ces clusters est :

Nombre de clusters = 7

Un cluster fait 512 octets

**Espace récupérable = 512 octets x 7 = 3584 octets = 3,5 Ko**

3°) Les fichiers fragmentés sont les suivants :

- ORDRE.DAT
- MSCDEX.SYS
- ESSAI.TXT

4°) Les fichiers ne contenant pas d'erreur sont les fichiers non fragmentés et les fichiers fragmentés (la fragmentation n'est pas une erreur, elle naît dans la manipulation normale des fichiers).  
 Un cluster fait  $16 \times 512 = 8 \text{ Ko}$ .

Nom du fichier	Taille du fichier en octets	Capacité du cluster occupée en octets	Capacité du disque perdu en octets
ORDRES.DAT	91120	$12 \times 8192 = 98304$	7184
MSC DEX.SYS	48072	$6 \times 8192 = 49152$	1080
RACAP.XLS	16791	$3 \times 8192 = 24576$	7785
LISEZ-MOI.TXT	106240	$13 \times 8192 = 106496$	256
MISSION.DAT	99442	$13 \times 8192 = 106496$	7054
ASSIST.COM	61952	$8 \times 8192 = 65536$	3584
DOLLAR.OVL	33191	$8 \times 8192 = 65536$	7769
SOUTIEN.DOC	225	$1 \times 8192 = 8192$	7967

**Espace disque total perdu : 42679 octets soit 41,67 Ko**

### EXERCICE 12

1°) Le disque possède :

- 10 pistes par face soit 10 cylindres numérotés cylindre 0 à cylindre 9 et piste 0 à piste 9 ( $P_0$  à  $P_9$ )
- 2 plateaux, soit 4 têtes numérotées ( $T_0$  à  $T_3$ )

2°) La capacité d'un cylindre

Capacité d'une piste =  $18 \text{ secteurs} \times 512 = 9 \text{ Ko}$

Capacité d'un cylindre =  $9 \text{ Ko} \times 4 \text{ têtes} = 36 \text{ Ko}$

Nombre de cylindres ( $N_{ycl}$ ) occupés par le fichier

Le fichier fait 2520 Ko

$N_{ycl} = 2520 / 36 \text{ Ko} = 7$

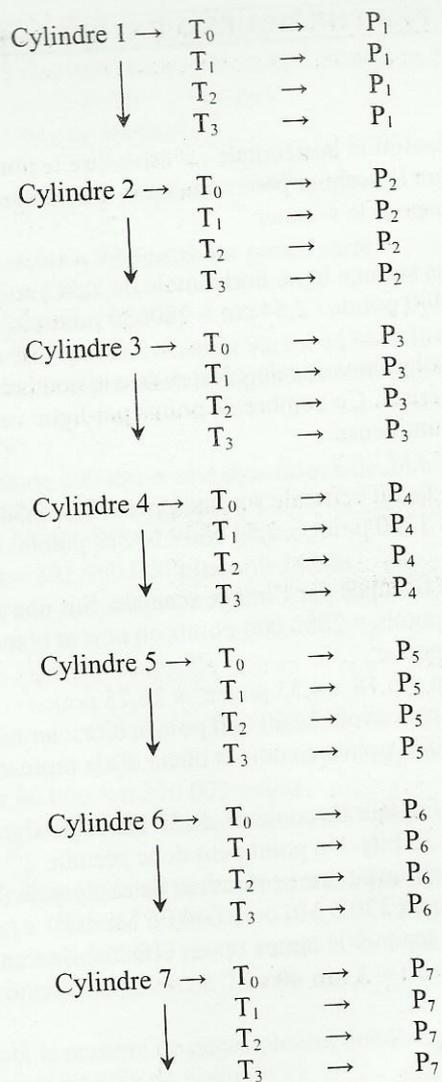
**Nombre de cylindres occupés par le fichier = 7**

3°) Cheminement de la lecture du fichier

Le fichier occupe 7 cylindres (ce sont les cylindres 1, 2, 3, 4, 5, 6 et 7, le cylindre 0 étant occupé par la zone système).

Le contrôleur pour lire un fichier, lit un cylindre entier avant de passer au cylindre suivant. La lecture d'un cylindre commence par la tête 0, puis la tête 1 jusqu'à la dernière tête (une tête lisant une partie).

Ce processus est décrit sur le schéma suivant :



## LES PERIPHERIQUES D'ENTREE/SORTIE

### EXERCICE 1

1°) Le chiffre 2400 représente la résolution horizontale : c'est-à-dire le nombre de points sur une ligne horizontale de 1 pouce (2,54 cm) que le scanner peut numériser. Ce nombre de points horizontaux dépend du nombre de cellules CDD dont dispose le scanner.

2°) Le scanner numérise 2400 points sur une ligne horizontale de 2,54 cm. Sur une ligne horizontale de 297 cm, il peut numériser :  $29,7 \text{ cm} \times 2400 \text{ points} / 2,54 \text{ cm} = 280630 \text{ points}$ .

3°) Le chiffre 1200 représente la résolution verticale, c'est-à-dire le nombre de points sur une ligne verticale de 2,54 cm que le scanner peut numériser. Ce nombre de points par ligne verticale dépend du pas du moteur pas à pas qui fait avancer la source lumineuse.

4°) Le scanner balaie 1200 lignes de façon verticale sur une distance de 2,54 cm. Sur une ligne verticale de 210 cm, il peut numériser :  $210 \text{ cm} \times 1200 \text{ points} / 2,54 \text{ cm} = 99212 \text{ points}$ .

5°) Calculons la quantité de mémoire occupée par l'image scannée. Sur une superficie de 1 pouce<sup>2</sup>, le scanner peut numériser  $1200 \times 2400 \text{ points} = 2880000 \text{ points}$  en noir et blanc.

Convertissons la taille de l'image en pouce<sup>2</sup>.

$157 \text{ mm} \times 110 \text{ mm} = 15,7 \text{ cm} \times 11 \text{ cm} = 6,18 \times 4,33 \text{ pouce}^2 = 26,75 \text{ pouce}^2$ .

Si sur une superficie de 1 pouce<sup>2</sup>, on peut avoir 2 800 000 points, alors sur une superficie de 26,75 pouce<sup>2</sup>, on aura :  $26,75 \times 2\,800\,000 = 77\,040\,000 \text{ points}$  en noir et blanc. Cela représente 77 040 000 bits en mémoire.

Si l'image est numérisée avec une profondeur de couleurs de 36 bits, cela signifie que pour représenter en mémoire un point numérisé, on utilise 36 bits. Un point peut donc prendre  $2^{36}$  couleurs possibles.

*La taille de l'image numérisée en couleur avec une profondeur de couleur de 36 bits est :*

*$77\,040\,000 \times 36 = 2\,773\,440\,000 \text{ bits}$ , soit 330,6 Mo ou 2 644,95 Mbits.*

*Avec un débit de 12 Mbits/s, il faut au scanner le temps t pour effectuer le transfert en mémoire.*

$t = 2644,95 \text{ Mbits} / 12 \text{ Mbits} = 220,4 \text{ s} \Rightarrow t = 3 \text{ mn } 40 \text{ s}$ .

### EXERCICE 2

1°) Un point est codé sur 8 bits par couleur primaire. Pour les trois couleurs (rouge, vert, bleu), un point sera codé sur 24 bits.

Soit ( $C_{pc}$ ) la capacité mémoire par pouce carré pour une image en noir et blanc.

$C_{pc} = 300 \times 600 = 180\,000 \text{ points}$ .

La taille de l'image vaut :  $21 \text{ cm} \times 29,7 \text{ cm} = 8,26 \times 11,7 \text{ pouce}^2 = 96,6 \text{ pouce}^2$ .

Si un pouce<sup>2</sup> occupe 180 000 bits, 96,6 pouce<sup>2</sup> occuperont :  $96,6 \times 180\,000 = 17\,388\,000 \text{ points}$  en noir et blanc.

Pour obtenir la capacité mémoire pour une image en couleur dont un point nécessite 8 bits par couleur primaire, soit 24 bits pour les trois couleurs primaires, on multiplie le nombre de points par la profondeur de couleur, ici égale à 24 bits.

$C_p = 17\,388\,000 \times 24 \text{ bits} = 417\,312\,000 \text{ bits}$ , soit  $C_p = 49,7 \text{ Mo}$  **Capacité = 49,7 Mo**

2°) La résolution verticale fait 600 ppp c'est-à-dire que sur une distance verticale de 2,54 cm, le moteur pas à pas fait 600 pas.

Le pas du moteur vaut donc  $25,4 \text{ mm} / 600 = 0,04 \text{ mm}$

**Pas = 0,04 mm**

3°) Pour une résolution verticale de 600 ppp, le moteur s'arrête et redémarre 600 fois sur une distance de 2,54 cm soit 1 pouce. La longueur de la feuille A4 vaut 29,7 cm, soit 11,7 pouce.

Pour une longueur de 11,7 pouces, le moteur s'arrêtera et redémarrera  $11,7 \times 600$  fois, soit 7020 fois.

**Le moteur s'arrête et redémarre 7020 fois pour numériser la feuille A4.**

4°) Le moteur s'arrête et redémarre 7020 fois pour numériser toute l'image en 30 s.  
En une seconde, il s'arrêtera et redémarrera  $7020/30 = 234$  fois/s.  
**Le moteur s'arrête et redémarre 234 fois par seconde.**

### EXERCICE 3

1°) La résolution est de 72 dpi, soit 72 points x 72 points par pouce carré.  
Le nombre de points par pouce carré vaut 5184.  
Superficie de l'image :  $11 \times 8,5 = 93,5$  pouce<sup>2</sup>.  
Pour un pouce<sup>2</sup>, on a 5184 points. Pour 93,5 pouce<sup>2</sup>, on aura  $93,5 \times 5184 = 484704$  points.  
Pour une profondeur de couleur de 8 bits, la taille du fichier générée sera :  $484704 \times 8 \text{ bits} = 3877632$  bits.  
**Taille du fichier = 473,3 Ko**

2°) On prend maintenant une résolution de 300 dpi et une dynamique de 24 bits.  
Nombre de points par pouce<sup>2</sup> =  $300 \times 300 = 90\,000$  points  
Nombre de points de l'image =  $93,5 \times 90\,000 = 8\,415\,000$  points  
Taille du fichier =  $8\,415\,000 \times 24 \text{ bits} = 201\,960\,000$  bits, soit 24 Mo.  
**Taille du fichier = 24 Mo**

### EXERCICE 4

Superficie de l'image en pouce<sup>2</sup> :  $254 \text{ mm} \times 180 \text{ mm} = 25,4 \text{ cm} \times 18 \text{ cm} = 10 \times 7,08 \text{ pouce}^2 = 70,8 \text{ pouces}^2$ .  
Nombre de points exprimés par pouce carré :  $300 \times 300 = 90\,000$  points/pouce<sup>2</sup>  
Nombre de points de l'image :  $70,3 \times 90\,000 = 6\,372\,000$  points.  
Or un point est matérialisé par une goutte d'encre. L'impression de l'image utilise donc 6 372 000 gouttes d'encre.  
Le volume d'une goutte est tel que un litre (1000ml) peut en contenir jusqu'à 3 milliards.  
Le volume d'une goutte est donc égal à  $1000 \text{ ml} / 3\,000\,000\,000 = 10^{-6} / 3 \text{ ml}$   
Le volume d'encre pour l'impression des 6 372 000 points est de  $10^{-6} / 3 \text{ ml} \times 6\,372\,000 = 2,1 \text{ ml}$   
Le volume d'encre utilisé pour imprimer l'image est de 2,1 ml.

2°) La vitesse d'impression représente le nombre de pages pleines que l'imprimante peut imprimer en une minute. Une ligne de 80 caractères contient  $80 \times 64 \text{ points} = 5120$  points.  
On peut imprimer 40 lignes de 80 caractères sur une page. Cela donne sur une page  $11520 \times 40 \text{ points} = 460\,800$  points.  
Les gouttes d'encre sont protégées à une vitesse 20 000 gouttes par seconde. Pour imprimer la page, il met  $460\,800 / 20\,000 = 23$  s.  
Si en 23 s, une page est imprimée, en une minute, on imprimera  $60/23 = 2,6$  pages environ.  
**Vitesse d'impression = 3 ppm**

3°) Le document fait 98 pages. Pour une vitesse de 3 ppm, on mettra  $98/3 = 32,67$  minutes, soit 32 min 40 s.

4°) La zone imprimable peut contenir au maximum 460 800 points, soit 460 800 gouttes d'encre dont une seule a un volume de  $10^{-6} / 3 \text{ ml}$ .  
Le volume d'encre par page est donc égal à  $(10^{-6} \text{ ml} / 3) \times 460\,800 = 153,6 \cdot 10^{-3} \text{ ml}$   
Pour les 98 pages, le volume total sera :  $153,6 \cdot 10^{-3} \text{ ml} \times 98 = 15 \text{ ml}$ .  
**Le volume total d'encre utilisé pour l'impression des 98 pages est de 15 ml.**

### EXERCICE 5

- 1°)
- ⑥ Nettoyage et neutralisation du tambour
  - ② Ecriture optique de l'image sur le tambour
  - ⑤ Fixation de l'image sur le papier

- ④ Transfert de l'image sur le papier
- ③ Développement de l'image
- ① Chargement électrostatique du tambour.

2°)

- La caractéristique 1200 x 1200 dpi représente la résolution, c'est-à-dire le nombre de points imprimés par pouces<sup>2</sup>.
- La caractéristique 16 ppm représente la vitesse d'impression de l'imprimante, soit 16 pages imprimées par minute.

3°) Superficie de l'image en pouce<sup>2</sup> = 25,4 cm x 17,78 cm = 10 x 7 pouce<sup>2</sup>

Le nombre de points par pouce carré vaut : 1200 x 1200 = 1 440 000 points.

Nombre de points de l'image : 1 440 000 x 70 = 100 800 000 points, soit 100 800 000 bits. Cela donne 12 Mo.

Cette image ne pourra pas être entièrement imprimée. Seulement le tiers pourra être imprimé. En effet, la capacité de la mémoire tampon est 3 fois plus petite que la taille de l'image.

### EXERCICE 6

1°) L'économiseur d'écran permet de réduire la dégradation de la couche de phosphore se trouvant au fond de l'écran. En effet, quand le moniteur affiche pendant longtemps une même image, les électrons du faisceau lumineux bombardent les mêmes points (pixels) de l'écran. Ce qui accélère la dégradation de la couche phosphorescente.

L'écran de veille est un petit programme sous Windows qui va couper le faisceau lumineux et faire apparaître des images aléatoires pour réduire la dégradation, ou permettre une dégradation uniforme à la longue.

2°) La fonction d'économie d'énergie a une importance capitale pour les ordinateurs portables. En effet, l'ordinateur portable doit pouvoir fonctionner en autonomie grâce à sa batterie.

La fonction d'économie d'énergie va permettre d'arrêter le fonctionnement de certains éléments (écran, disque dur) et même de mettre tout l'ordinateur en veille.

Cela permet de réduire la consommation électrique et donc d'augmenter la durée d'autonomie de la batterie. La fonction d'économiseur d'écran n'a aucun intérêt pour les portables à écran LCD. Cette fonction est faite pour les ordinateurs à écran de type CRT (Cathod Ray Tube) ou écran à tube cathodique.

### EXERCICE 7

1°)

- Fréquence verticale ou fréquence trame : c'est le nombre de renouvellements de l'image par seconde
- Fréquence horizontale : c'est le nombre de lignes d'écran balayés à une seconde. Cela représente la vitesse de déplacement du faisceau d'électron d'un bord à l'autre de l'écran.
- Bande passante : c'est le nombre de pixels que le faisceau lumineux balaie en une seconde.

2°)

La fréquence verticale est égale à X.

Fréquence horizontale

Un écran fait 480 lignes. Si l'écran est balayé X fois par seconde, le nombre de lignes balayées par seconde vaut 480X. La fréquence horizontale vaut donc 480X Hz.

Bande passante

Le nombre de pixels d'un écran vaut 640 x 480, soit 307200 pixels. Si l'écran est balayé X fois par seconde, le nombre de pixels balayés par seconde vaut 307200X.

La bande passante vaut 307200X Hz.

3°) Application numérique :  $X = 80 \text{ Hz}$ .  
Fréquence horizontale ( $F_H$ ) =  $480 \times 80 = 38400 \text{ Hz}$   
 $F_H = 38,4 \text{ KHz}$   
Bande passante (BP) =  $307200 \times 80 = 24576000 \text{ Hz}$   
 $BP = 24,57 \text{ MHz}$ .

### EXERCICE 8

1°) On emploie également les termes **MULTISYNC**, ou **multi-fréquence** pour désigner le terme AUTSCAN.

Un moniteur AUTOSCAN est un moniteur capable de changer de fréquence horizontale si l'on modifie sa résolution. Pour ce faire, il recherche automatiquement la fréquence horizontale adéquate par rapport à la nouvelle résolution qui lui a été imposée.

En effet, pour une même fréquence verticale, si l'on modifie la résolution de l'écran, on modifie le nombre de lignes de l'écran. Or, la fréquence horizontale, c'est le produit du nombre de lignes par la fréquence verticale. En modifiant donc la résolution, on modifie la fréquence horizontale.

2°) Résolutions que le moniteur peut afficher  
 Soient  $F_V$  la fréquence verticale,  $F_H$  la fréquence horizontale et  $N$  le nombre de lignes de l'écran. Calculons le nombre de lignes d'écran affichables pour les deux fréquences extrêmes 31,5 KHz et 64 KHz.

$$F_H = F_V \times N_L \Rightarrow N = F_H / F_V$$

Pour  $F_H = 31,4 \text{ KHz}$  et  $F_V = 72 \text{ Hz}$

$$N_{\text{inf}} = 31500/72 = 437,5$$

Pour  $F_H = 64 \text{ KHz}$  et  $F_V = 72 \text{ Hz}$

$$N_{\text{sup}} = 64000/72 = 888,88$$

Le moniteur peut balayer une plage de lignes comprise entre 437 lignes et 888 lignes. Toutes résolutions dont le nombre de lignes est en dehors de cette plage ne pourra pas être affichée.

**Le moniteur pourra donc afficher les résolutions suivantes : 640 x 400, 1024 x 768 et 800 x 600**

3°) Les résolutions 1600 x 1200, 1024 x 1024 et 1280 x 1024 ne pourront pas être affichées correctement. La solution pour arriver à les afficher consiste à faire fonctionner le moniteur en mode entrelacé. Dans ce mode, la carte graphique pilote le faisceau d'électron de sorte que celui-ci balaie d'abord les lignes paires pour un premier balayage. Les lignes impaires sont balayées dans un deuxième passage. Il faut donc deux passages du faisceau pour balayer un écran entier. Cela revient à balayer la moitié des lignes en une passe. Cela fait que :

La résolution    1600 x 1200 devient 1600 x 600  
                           1024 x 1024 devient 1024 x 512  
                           1280 x 1024 devient 1024 x 512

### EXERCICE 9

1°) Le pas est la distance séparant deux pixels. Pour une résolution de 1024 x 768, une ligne d'écran contient 1024 pixels. Le rapport de la longueur de cette ligne par le nombre de pixels sur la ligne donne le pas. Soit  $D$  la diagonale de l'écran,  $L$  la longueur d'une ligne d'écran.

$$L = 4/5 \times D$$

Calculons la longueur  $L$  pour chaque écran

Ecran 14:  $L = 4/5 \times 14 \times 2,54 \text{ cm} = 28,44 \text{ cm} = 284,4 \text{ mm}$

Ecran 15:  $L = 4/5 \times 15 \times 2,54 \text{ cm} = 30,48 \text{ cm} = 304,8 \text{ mm}$

Ecran 17:  $L = 4/5 \times 17 \times 2,54 \text{ cm} = 34,54 \text{ cm} = 345,4 \text{ mm}$

**Pas de chaque écran**

Ecran 14: pas =  $284,4 / 1024 = 0,27 \text{ mm}$

Ecran 15: pas =  $304,8 / 1024 = 0,3 \text{ mm}$

Ecran 17: pas =  $345,4 / 1024 = 0,33 \text{ mm}$

2°) On impose un pas = 0,28 mm

On peut calculer la longueur (Lo) d'une ligne d'écran connaissant le pas et le nombre de points par ligne. On comparera ensuite cette longueur à celle des différents écrans. Si une longueur de ligne d'un moniteur est inférieure à Lo, alors ledit moniteur ne pourra pas afficher la résolution concernée avec le pas indiqué.

$$L_o = 0,28 \text{ mm} \times 1200 = 336 \text{ mm} = 33,6 \text{ cm}$$

Ce résultat signifie que pour afficher 1200 points séparés d'une distance de 0,28 mm les uns des autres, le moniteur doit avoir une ligne d'une longueur de 33,6 cm au moins.

Les moniteurs 14 pouces et 15 pouces ont respectivement une longueur de lignes de 28,44 cm et 30,48 cm. Ils ne peuvent donc pas afficher la résolution 1200 x 800 avec un pas de 0,28. Seul le moniteur de 17 pouces avec une longueur de ligne de 34,54 cm peut afficher une résolution de 1200 x 800 avec un pas de 0,28 mm.

### EXERCICE 10

1°) Taille physique de l'écran

La diagonale D de l'écran représente sa taille. La longueur d'une ligne de l'écran est notée L et vaut :

$$L = 4/5 \times D \Rightarrow D = L \times 5/4$$

$$L = 286,72 \text{ mm} \Rightarrow D = 286,72 \text{ mm} \times 5/4 = 358,4 \text{ mm} \text{ soit environ } 14 \text{ pouces.}$$

2°) Résolution maximale

La résolution est le produit du nombre de points sur une ligne horizontale par le nombre de lignes.

Le nombre de points sur une ligne horizontale ( $N_{PH}$ ) est le quotient de la longueur d'une ligne physique de l'écran (L) et du pas.

$$N_{PH} = L / \text{PAS} = 286,72 / 0,28 = 1024 \Rightarrow N_{PH} = 1024 \text{ pixels}$$

Nombre de lignes de l'écran ( $N_L$ )

Soit  $F_H$  la fréquence horizontale,  $F_V$  la fréquence verticale

$$F_H = F_V \times N_L \Rightarrow N_L = F_H / F_V \Rightarrow N_L = 53760/70 = 768 \text{ lignes}$$

**NB:** la fréquence horizontale est le nombre de lignes d'écran balayé par seconde.

De ces calculs, on tire la résolution maximale : **Résolution maximale = 1024 x 768**

3°) En mode entrelacé, un passage du faisceau d'électron balaie la moitié des lignes. L'autre moitié est balayée à la prochaine passe.

Le nombre de points balayés à chaque passe est 1024 x (768 / 2), soit 393216 pixels.

Si ce nombre est balayé 70 fois par seconde, la bande passante (BP) du moniteur en mode entrelacé est le suivant :

$$BP = 393216 \times 70 = 27525120 \text{ Hz} \Rightarrow BP = 27,52 \text{ MHz}$$

### EXERCICE 11

1°) La norme Truecolor permet un affichage en 16,7 millions de couleur, soit 24 niveaux de couleur par pixel. Les 24 bits par pixel concernent les 3 couleurs primaires. Par couleur primaire, il faut donc 8 bits. Chaque DAC (un pour le rouge, un pour le vert et un pour le bleu) reçoit alors 8 bits pour quantifier chaque couleur primaire.

2°) Calcul de la mémoire vidéo en Truecolor

Soit R la résolution à afficher et  $N_c$  la profondeur de couleur et  $C_{VRAM}$  la capacité de la mémoire vidéo

$$C_{VRAM} = R \times N_c$$

Dans le cas présent R = 1600 x 1200 et  $N_c = 24$  bits

$$C_{VRAM} = 1600 \times 1200 \times 24 \text{ bits} = 46\,080\,000 \text{ bits} \Rightarrow C_{VRAM} = 5,5 \text{ Mo}$$

3°) Cherchons les niveaux de couleur pour chaque nombre de couleurs données.

256 couleurs nécessitent 8 niveaux de couleur par pixel ( $256 = 2^8$ )

131072 couleurs nécessitent 17 niveaux de couleurs par pixel ( $131072 = 2^{17}$ )

65536 couleurs nécessitent 16 niveaux de couleurs par pixel ( $65536 = 2^{16}$ )

On multiplie ensuite chaque résolution par chaque profondeur de couleur pour trouver la capacité de la Ram vidéo correspondante.

La résolution et le niveau de couleur pour lesquels la valeur de la Ram vidéo trouvée sera inférieure à 1 Mo ne pourront pas être affichés.

Résolutions Couleurs	800 x 600	1280 x 1024	640 x 480	1024 x 768
256 (niveau de couleur = 8)	<b>468,75 Ko</b>	<b>1,25 Ko</b>	<b>300 Ko</b>	<b>768 Ko</b>
131072 (niveau de couleur = 17)	<b>0,97 Ko</b>	<b>2,65 Ko</b>	<b>637,5 Ko</b>	<b>1,6 Ko</b>
65536 (niveau de couleur = 16)	<b>937,5 Ko</b>	<b>2,5 Mo</b>	<b>600 Ko</b>	<b>1,5 Ko</b>

Pour 1 Mo de Ram vidéo, on peut afficher :

- en 256 couleurs, les résolutions suivantes : 800 x 600, 640 x 480 et 1024 x 768
- en 131072 couleurs, les résolutions suivantes : 800 x 600 et 640 x 480
- en 65536 couleurs, les résolutions suivantes : 800 x 600 et 640 x 480

### EXERCICE 12

1°) Déterminons les niveaux de couleur pour chaque nombre de couleurs.

Nombre de couleurs	Niveaux de couleur
16 = (2 <sup>4</sup> )	4 bits
256 = (2 <sup>8</sup> )	8 bits
65536 = (2 <sup>16</sup> )	16 bits
16,7 millions = (2 <sup>24</sup> )	24 bits
4 milliards = (2 <sup>32</sup> )	32 bits

On fait ensuite le produit de chaque résolution par la profondeur de couleur. Le résultat trouvé sera comparé à chaque capacité mémoire vidéo standard afin de choisir pour le couple [Résolution - couleur], la capacité mémoire standard qui sera immédiatement supérieur à la valeur calculée.

Résolutions Couleurs	640 x 480	800 x 600	1024 x 768	1028 x 1024	1600 x 1200	2048 x 2048
16	256 Ko	256 Ko	512 Ko	1 Mo	1 Mo	1 Mo
256	512 Ko	512 Ko	1 Mo	2 Mo	2 Mo	8 Mo
65536	1 Mo	1 Mo	2 Mo	4 Mo	4 Mo	16 Mo
16,7 millions	1 Mo	2 Mo	4 Mo	4 Mo	8 Mo	16 Mo
4 milliards	2 Mo	2 Mo	4 Mo	8 Mo	8 Mo	32 Mo

2°) Si le couple [Résolution - Couleur] est supérieur à la mémoire vidéo, l'image est brouillée.

Généralement, quand cela arrive sous Windows, il faut redémarrer l'ordinateur, mais en mode sans échec.

Dans ce mode, le système prend par défaut une résolution de 640 x 480 en 16 couleurs. C'est le standard d'affichage pour les cartes graphiques VGA sous DOS.

BTS SESSION 2000  
INFORMATIQUE INDUSTRIELLE ET MAINTENANCE

**TECHNOLOGIES DES ORDINATEURS**

**EXERCICE**

1°) Facteur de groupage

Le facteur de groupage est le nombre d'articles par bloc physique. La taille d'un article est de 240 caractères ou octets. Il reste à trouver la taille d'un bloc physique. On sait que les zones de travail du programme dont le code objet fait 23 Ko, valent 2,5 Ko. En d'autres termes, l'espace mémoire réservé aux données que le programme doit traiter occupe 2,5 Ko. Les données à traiter sont les données lues sur la bande et ce, bloc physique par bloc physique.

Un bloc physique fait donc 2,5 Ko maximum pour pouvoir être chargé dans un espace mémoire de 2,5 Ko. Soit  $f$ , le facteur de groupage.  $2,5 \text{ Ko} = 2560 \text{ octets}$ .

$f = \text{taille du bloc physique} / \text{taille d'un article}$

$$f = 2560 / 240 = 10,66 \Rightarrow \boxed{f = 10}$$

2°) Longueur de bande occupée par le fichier

Le fichier fait 25 000 articles en raison de 240 caractères par article. Le fichier a donc une taille de  $25\,000 \times 240 = 6\,000\,000$  de caractères.

Soit  $C_{pf}$  (capacité du fichier) et  $L_f$  (longueur de bande occupée par le fichier) et  $d$  la densité d'enregistrement.

$$C_{pf} = L_f \times d \Rightarrow L_f = C_{pf} / d$$

$$L_f = 6\,000\,000 \times 2,54 \text{ cm} / 1600 \Rightarrow \boxed{L_f = 9525 \text{ cm}}$$

3°) La capacité du disque (diskpack) est donnée par la formule suivante :

$$\text{Cap} = C \times H \times S \times 2 \text{ Ko}$$

$C$  = nombre de cylindres = nombre de pistes par face

$H$  = nombre de tête = nombre de face (il y a 2 faces par disque)

$S$  = nombre de secteurs par pistes

$2 \text{ Ko}$  = capacité d'un secteur

$$\text{Cap} = 200 \times (11 \times 2) \times 10 \times 2 \text{ Ko} = 88\,000 \text{ Ko}$$

4°) La précision « un bloc ne peut pas être partagés entre 2 secteurs » est de taille, car cela signifie q'un bloc physique doit pouvoir tenir dans un secteur de 2 Ko. Pour 240 caractères par article, le nouveau facteur de groupage pour enregistrer ce fichier est :

$$f = 2 \text{ Ko} / 240 = 2048 / 240 = 8,53, \boxed{\text{on prend } f = 8}$$

5°) Nombre de cylindres occupés par le fichier

- Capacité d'une piste = (capacité secteur x nombre secteur par piste) =  $10 \times 2 \text{ Ko} = 20 \text{ Ko}$

- Nombre de têtes = Nombre de disques x 2 =  $11 \times 2 = 22$  (il y a 2 têtes par disques)

- Capacité d'un cylindre = capacité d'une piste x nombre de têtes =  $22 \times 20 \text{ Ko} = 440 \text{ Ko}$

- La taille du fichier est de 6 000 000 caractères

- Nombre de cylindres ( $N_{cy}$ ) occupés par le fichier = taille du fichier / capacité d'un cylindre

$$N_{cy} = 6\,000\,000 / 440 \text{ Ko} = 13,64. \boxed{\text{On prend 14 cylindres}}$$

**PROBLEME**

**Première partie**

1°) Pour une FAT 12 bits, la taille d'une partition MS DOS sera :

- La FAT12 donne  $2^{12}$  entrées de FAT ou  $2^{12}$  clusters
- Un cluster est égal à 16 secteurs
- Un secteur est égal à 512 octets

Taille partition =  $2^{12} \times 16 \times 512 \Rightarrow$  **Taille partition = 32 Mo**

2°) Type de FAT pour un disque de 256 Mo

- un cluster = 8 secteurs
- un cluster = 512 octets

$2^{\text{Fat}} \times 8 \times 512 = 256 \text{ Mo}$

$2^{\text{FAT}} \times 2^{12} = 2^8 \times 2^{20} \Rightarrow 2^{\text{FAT}} = 2^{16} \Rightarrow \text{FAT} = 16$

**On a donc une FAT 16 bits**

3°) Nombre de secteurs par cluster

On a précédemment une FAT 16.

$2^{16} \times \text{nombre de secteurs par cluster} \times 512 = 384 \text{ Mo}$

Nombre de secteurs par cluster =  $384 \text{ Mo} / 2^{16} \times 512 \text{ octets} = 384 / 32$

Nombre de secteur par cluster = 12

**Deuxième partie**

1°) Pour déterminer X, on cherche le nombre de clusters occupés par chaque fichier et on en fait la somme.

- Soient  $X_1, X_2, X_3$  le nombre respectif de clusters occupés par les fichiers File 0001, File 0002 et File 003.
- Un cluster fait 8 secteurs soit  $8 \times 512 = 4096 \text{ octets} = 4 \text{ Ko}$

$X_1 = 29,75 \text{ Ko} / 4 \text{ Ko} = 7,44 \text{ soit } X_1 = 8 \text{ clusters}$

$X_2 = 2,3 \text{ Ko} / 4 \text{ Ko} = 0,57 \text{ soit } X_2 = 1 \text{ cluster}$

$X_3 = 16,25 \text{ Ko} / 4 \text{ Ko} = 4,06 \text{ soit } X_3 = 5 \text{ clusters}$

$X = X_1 + X_2 + X_3 = 8 + 1 + 5 = 14 \Rightarrow$  **X = 14 clusters**

2°) Représentation des 32 premières entrées de la FAT.

La première entrée porte le n°2

2	3	4	5	6	7	8	9
0004	FFF7	0005	0012	FFF7	FFF7	FFF8	FFF7
10	11	12	13	14	15	16	17
000C	FFF7	000D	000E	0011	FFF7	FFF7	FFF8
18	19	20	21	22	23	24	25
0013	0014	0015	0016	FFF8	0000	0000	0000
26	27	28	29	30	31	32	33
0000	0000	FFF7	FFF7	FFF7	FFF7	FFF7	FFF7

Description

**File0001.CHK**

- occupe 8 clusters
- 1<sup>er</sup> fichier enregistré
- Il est fragmenté en 2 morceaux
- 1<sup>er</sup> fragment 12 Ko soit 3 clusters
- 2<sup>e</sup> fragment commence après le 3<sup>e</sup> fichier

**File0002.CHK**

- occupe 1 cluster
- enregistre le 1<sup>er</sup> fragment de file 0001

**File0003.CHK**

- occupe 5 clusters
- commence après le 2<sup>e</sup> fichier
- il n'est pas fragmenté

**Clusters défectueux**

3, 6, 7, 9, 11, 15, 16, 28, 29, 30, 31, 32, 33.

3°) Cluster de début et de fin

File0001.CHK

Début: 2

Fin: 22

File0002.CHK

Début: 8

Fin: 8

File0003.CHK

Début: 10

Fin: 17

4°) Espace disque perdu

File0001.CHK

Taille = 29,75 Ko

Nombre clusters = 8

Taille maxi = 8 x 4 Ko = 32 Ko

Espace perdu = 32 Ko - 29,75 Ko

Espace perdu = 2,25 Ko

File0002.CHK

Taille = 2,3 Ko

Nombre clusters = 1

Taille maxi = 1 x 4 Ko = 4 Ko

Espace perdu = 4 Ko - 2,3 Ko

Espace perdu = 1,7 Ko

File0003.CHK

Taille = 16,25 Ko

Nombre clusters = 5

Taille maxi = 5 x 4 Ko = 20 Ko

Espace perdu = 20 Ko - 16,25 Ko

Espace perdu = 2,25 Ko

**Troisième partie**

1°) Type de FAT :

Le type de FAT est la FAT 16 car chaque entrée est codée sur 16 bits.

2°) On compte le nombre de fin de fichier (FFF8). On trouve 5 fichiers.

3-a) Cluster de départ et de fin

NB : Les 5 fichiers seront identifiés par F1, F2, F3, F4 et F5

F1

Début = 2

Fin = 22

Nombre-clusters = 3

F2

Début = 7

Fin = 29

Nombre clusters = 3

F3

Début = 8

Fin = 15

Nombre clusters = 3

F4

Début = 18

Fin = 19

Nombre clusters = 3

F5

Début = 21

Fin = 21

Nombre clusters = 3

3-b) Taille maximale de chaque fichier

Un cluster fait 1,5 Ko.

F1 : Taille maxi = 3 x 1,5 Ko = 4,5 Ko

F2 : Taille maxi = 5 x 1,5 = 7,5 Ko

F3 : Taille maxi = 7,5 Ko

F4 : Taille maxi = 2 x 1,5 Ko = 3 Ko

F5 : Taille maxi = 1 x 1,5 Ko = 1,5 Ko

4°) Nombre de fichiers non fragmentés

On a 3 fichiers (F3, F4, F5) qui ne sont pas fragmentés. Ils occupent des clusters contigus.

BTS SESSION 2003  
INFORMATIQUE INDUSTRIELLE ET MAINTENANCE

**TECHNOLOGIES DES ORDINATEURS**

**EXERCICE 1**

1°) Entrées, sorties et E/S

	Fig. 1	Fig. 2	Fig. 3	Fig. 4
Nombre Entrées dédiées	12	14	08	12
Nombre Sorties dédiées	04	02	06	08
Nombre Entrées / Sorties	04	06	08	08

2°) Dans la structure de sortie des PAL, les sorties sont prioritaires dans la référence si le PAL possède à la fois des sorties à registres et des sorties combinatoires.

Fig. 1	Fig. 2	Fig. 3	Fig. 4
PAL 20R4	PAL 20L8	PAL 16R6	PAL 2058

3°)

Famille PAL 24 pattes	Fig. 1, Fig. 2, Fig. 4
Famille PAL 20 pattes	Fig. 3