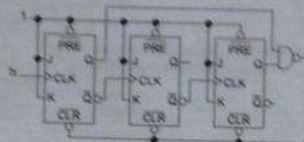
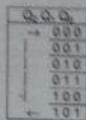


4.4 Exercice 4

4.4.1 Faire le schéma d'un compteur asynchrone modulo 8 à l'aide de bascules J-K. Un compteur 85 compte de 0 à 4. A l'état 5 nous activons la RAZ. Cet état ne dure que le temps de propagation dans la logique de detection et celle de RAZ. La configuration $Q_2, Q_1, Q_0 = 1, 1, 1$ n'arrive que pour l'état 5 (à 5) : $RAZ = Q_2 \cdot Q_1 \cdot Q_0$



5 TD5

5.1 Exercice 1

Un système à microprocesseur possède :

- un bus d'adresse de 16 bits ($A_{15} \dots A_0$),
- une bus de données de 8 bits ($D_7 \dots D_0$),
- des signaux de validation du circuit \overline{CE} et de lecture/écriture R/\overline{W} .

Nous souhaitons insérer les mémoires suivantes :

- une mémoire morte M_1 de 16K octets implantés à partir de 2000_{16} ,
- une mémoire morte M_2 de 8K octets implantés à partir de 8000_{16} ,
- une mémoire morte M_3 de 4K octets implantés juste après celle de 8K.

5.1.1 Élaborer les équations logiques de sélection de ces mémoires. Donner les schémas logiques en un nombre minimum de portes.

$$16K = 16 \cdot 1024 = 2^{14} = 4 \times 2^{12} = 4 \times 16^3 = [4000]_{16} \Rightarrow M_1 \text{ de } 2000_{16} \text{ à } 2000_{16} + 4000_{16} - 1 = 5FFF$$

$$8K = 8 \cdot 1024 = 2^{13} = 2 \times 2^{12} = 2 \times 16^3 = [2000]_{16} \Rightarrow M_2 \text{ de } 8000_{16} \text{ à } 8000_{16} + 2000_{16} - 1 = 9FFF$$

$$4K = 4 \cdot 1024 = 2^{12} = 1 \times 2^{12} = 1 \times 16^3 = [1000]_{16} \Rightarrow M_3 \text{ de } A000_{16} \text{ à } A000_{16} + 1000_{16} - 1 = AFFF$$

Circuit	\overline{CE}_1	\overline{CE}_2	\overline{CE}_3
M_1	$\overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot \overline{A_{12}} \cdot \overline{A_{11}} \cdot \overline{A_{10}} \cdot \overline{A_9} \cdot \overline{A_8} \cdot \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot \overline{A_4} \cdot \overline{A_3} \cdot \overline{A_2} \cdot \overline{A_1} \cdot \overline{A_0}$	$\overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot \overline{A_{12}} \cdot \overline{A_{11}} \cdot \overline{A_{10}} \cdot \overline{A_9} \cdot \overline{A_8} \cdot \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot \overline{A_4} \cdot \overline{A_3} \cdot \overline{A_2} \cdot \overline{A_1} \cdot \overline{A_0}$	$\overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot \overline{A_{12}} \cdot \overline{A_{11}} \cdot \overline{A_{10}} \cdot \overline{A_9} \cdot \overline{A_8} \cdot \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot \overline{A_4} \cdot \overline{A_3} \cdot \overline{A_2} \cdot \overline{A_1} \cdot \overline{A_0}$
M_2	$\overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot \overline{A_{12}} \cdot \overline{A_{11}} \cdot \overline{A_{10}} \cdot \overline{A_9} \cdot \overline{A_8} \cdot \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot \overline{A_4} \cdot \overline{A_3} \cdot \overline{A_2} \cdot \overline{A_1} \cdot \overline{A_0}$	$\overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot \overline{A_{12}} \cdot \overline{A_{11}} \cdot \overline{A_{10}} \cdot \overline{A_9} \cdot \overline{A_8} \cdot \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot \overline{A_4} \cdot \overline{A_3} \cdot \overline{A_2} \cdot \overline{A_1} \cdot \overline{A_0}$	$\overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot \overline{A_{12}} \cdot \overline{A_{11}} \cdot \overline{A_{10}} \cdot \overline{A_9} \cdot \overline{A_8} \cdot \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot \overline{A_4} \cdot \overline{A_3} \cdot \overline{A_2} \cdot \overline{A_1} \cdot \overline{A_0}$
M_3	$\overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot \overline{A_{12}} \cdot \overline{A_{11}} \cdot \overline{A_{10}} \cdot \overline{A_9} \cdot \overline{A_8} \cdot \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot \overline{A_4} \cdot \overline{A_3} \cdot \overline{A_2} \cdot \overline{A_1} \cdot \overline{A_0}$	$\overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot \overline{A_{12}} \cdot \overline{A_{11}} \cdot \overline{A_{10}} \cdot \overline{A_9} \cdot \overline{A_8} \cdot \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot \overline{A_4} \cdot \overline{A_3} \cdot \overline{A_2} \cdot \overline{A_1} \cdot \overline{A_0}$	$\overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot \overline{A_{12}} \cdot \overline{A_{11}} \cdot \overline{A_{10}} \cdot \overline{A_9} \cdot \overline{A_8} \cdot \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot \overline{A_4} \cdot \overline{A_3} \cdot \overline{A_2} \cdot \overline{A_1} \cdot \overline{A_0}$

$$\overline{CE}_{M1} = \overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot \overline{A_{12}} \cdot \overline{A_{11}} \cdot \overline{A_{10}} \cdot \overline{A_9} \cdot \overline{A_8} \cdot \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot \overline{A_4} \cdot \overline{A_3} \cdot \overline{A_2} \cdot \overline{A_1} \cdot \overline{A_0}$$

$$\overline{CE}_{M2} = \overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot \overline{A_{12}} \cdot \overline{A_{11}} \cdot \overline{A_{10}} \cdot \overline{A_9} \cdot \overline{A_8} \cdot \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot \overline{A_4} \cdot \overline{A_3} \cdot \overline{A_2} \cdot \overline{A_1} \cdot \overline{A_0}$$

$$\overline{CE}_{M3} = \overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot \overline{A_{12}} \cdot \overline{A_{11}} \cdot \overline{A_{10}} \cdot \overline{A_9} \cdot \overline{A_8} \cdot \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot \overline{A_4} \cdot \overline{A_3} \cdot \overline{A_2} \cdot \overline{A_1} \cdot \overline{A_0}$$

5.2 Exercice 2

Soit un système à base d'un microprocesseur travaillant sur 8 bits de données et 16 bits d'adresse. Il accède à une mémoire de 8 kilooctets lorsque ses adresses sont comprises entre 5600_{16} et $7FFF_{16}$.

5.2.1 Donner le schéma logique du décodage d'adresses en un nombre minimum de portes.

5.2.2 Proposer un schéma à l'aide d'un décodeur réalisant le même décodage.

Taille mémoire : $8K = 8 \times 2^{10} = 2 \times 2^{11} = 2^{12} = [2000]_{16} \Rightarrow$ Mémoire @ sur 13 bits. μ Processeur @ sur 16 bits.

$$5600_{16} + [2000]_{16} = 7600_{16} \Rightarrow$$

$$7600_{16} + [2000]_{16} = 9600_{16} \Rightarrow$$

$$9600_{16} + [2000]_{16} = B600_{16} \Rightarrow$$

$$B600_{16} + [2000]_{16} = D600_{16} \Rightarrow$$

$$D600_{16} + [2000]_{16} = F600_{16} \Rightarrow$$

$$F600_{16} + [2000]_{16} = 1600_{17} \Rightarrow$$

$$1600_{17} + [2000]_{16} = 3600_{18} \Rightarrow$$

$$3600_{18} + [2000]_{16} = 5600_{19} \Rightarrow$$

$$5600_{19} + [2000]_{16} = 7600_{20} \Rightarrow$$

$$7600_{20} + [2000]_{16} = 9600_{21} \Rightarrow$$

$$9600_{21} + [2000]_{16} = B600_{22} \Rightarrow$$

$$B600_{22} + [2000]_{16} = D600_{23} \Rightarrow$$

$$D600_{23} + [2000]_{16} = F600_{24} \Rightarrow$$

$$F600_{24} + [2000]_{16} = 1600_{25} \Rightarrow$$

$$1600_{25} + [2000]_{16} = 3600_{26} \Rightarrow$$

$$3600_{26} + [2000]_{16} = 5600_{27} \Rightarrow$$

$$5600_{27} + [2000]_{16} = 7600_{28} \Rightarrow$$

$$7600_{28} + [2000]_{16} = 9600_{29} \Rightarrow$$

$$9600_{29} + [2000]_{16} = B600_{30} \Rightarrow$$

$$B600_{30} + [2000]_{16} = D600_{31} \Rightarrow$$

$$D600_{31} + [2000]_{16} = F600_{32} \Rightarrow$$

$$F600_{32} + [2000]_{16} = 1600_{33} \Rightarrow$$

$$1600_{33} + [2000]_{16} = 3600_{34} \Rightarrow$$

$$3600_{34} + [2000]_{16} = 5600_{35} \Rightarrow$$

$$5600_{35} + [2000]_{16} = 7600_{36} \Rightarrow$$

$$7600_{36} + [2000]_{16} = 9600_{37} \Rightarrow$$

$$9600_{37} + [2000]_{16} = B600_{38} \Rightarrow$$

$$B600_{38} + [2000]_{16} = D600_{39} \Rightarrow$$

$$D600_{39} + [2000]_{16} = F600_{40} \Rightarrow$$

$$F600_{40} + [2000]_{16} = 1600_{41} \Rightarrow$$

$$1600_{41} + [2000]_{16} = 3600_{42} \Rightarrow$$

$$3600_{42} + [2000]_{16} = 5600_{43} \Rightarrow$$

$$5600_{43} + [2000]_{16} = 7600_{44} \Rightarrow$$

$$7600_{44} + [2000]_{16} = 9600_{45} \Rightarrow$$

$$9600_{45} + [2000]_{16} = B600_{46} \Rightarrow$$

$$B600_{46} + [2000]_{16} = D600_{47} \Rightarrow$$

$$D600_{47} + [2000]_{16} = F600_{48} \Rightarrow$$

$$F600_{48} + [2000]_{16} = 1600_{49} \Rightarrow$$

$$1600_{49} + [2000]_{16} = 3600_{50} \Rightarrow$$

$$3600_{50} + [2000]_{16} = 5600_{51} \Rightarrow$$

$$5600_{51} + [2000]_{16} = 7600_{52} \Rightarrow$$

$$7600_{52} + [2000]_{16} = 9600_{53} \Rightarrow$$

$$9600_{53} + [2000]_{16} = B600_{54} \Rightarrow$$

$$B600_{54} + [2000]_{16} = D600_{55} \Rightarrow$$

$$D600_{55} + [2000]_{16} = F600_{56} \Rightarrow$$

$$F600_{56} + [2000]_{16} = 1600_{57} \Rightarrow$$

$$1600_{57} + [2000]_{16} = 3600_{58} \Rightarrow$$

$$3600_{58} + [2000]_{16} = 5600_{59} \Rightarrow$$

$$5600_{59} + [2000]_{16} = 7600_{60} \Rightarrow$$

$$7600_{60} + [2000]_{16} = 9600_{61} \Rightarrow$$

$$9600_{61} + [2000]_{16} = B600_{62} \Rightarrow$$

$$B600_{62} + [2000]_{16} = D600_{63} \Rightarrow$$

$$D600_{63} + [2000]_{16} = F600_{64} \Rightarrow$$

$$F600_{64} + [2000]_{16} = 1600_{65} \Rightarrow$$

$$1600_{65} + [2000]_{16} = 3600_{66} \Rightarrow$$

$$3600_{66} + [2000]_{16} = 5600_{67} \Rightarrow$$

$$5600_{67} + [2000]_{16} = 7600_{68} \Rightarrow$$

$$7600_{68} + [2000]_{16} = 9600_{69} \Rightarrow$$

$$9600_{69} + [2000]_{16} = B600_{70} \Rightarrow$$

$$B600_{70} + [2000]_{16} = D600_{71} \Rightarrow$$

$$D600_{71} + [2000]_{16} = F600_{72} \Rightarrow$$

$$F600_{72} + [2000]_{16} = 1600_{73} \Rightarrow$$

$$1600_{73} + [2000]_{16} = 3600_{74} \Rightarrow$$

$$3600_{74} + [2000]_{16} = 5600_{75} \Rightarrow$$

$$5600_{75} + [2000]_{16} = 7600_{76} \Rightarrow$$

$$7600_{76} + [2000]_{16} = 9600_{77} \Rightarrow$$

$$9600_{77} + [2000]_{16} = B600_{78} \Rightarrow$$

$$B600_{78} + [2000]_{16} = D600_{79} \Rightarrow$$

$$D600_{79} + [2000]_{16} = F600_{80} \Rightarrow$$

$$F600_{80} + [2000]_{16} = 1600_{81} \Rightarrow$$

$$1600_{81} + [2000]_{16} = 3600_{82} \Rightarrow$$

$$3600_{82} + [2000]_{16} = 5600_{83} \Rightarrow$$

$$5600_{83} + [2000]_{16} = 7600_{84} \Rightarrow$$

$$7600_{84} + [2000]_{16} = 9600_{85} \Rightarrow$$

$$9600_{85} + [2000]_{16} = B600_{86} \Rightarrow$$

$$B600_{86} + [2000]_{16} = D600_{87} \Rightarrow$$

$$D600_{87} + [2000]_{16} = F600_{88} \Rightarrow$$

$$F600_{88} + [2000]_{16} = 1600_{89} \Rightarrow$$

$$1600_{89} + [2000]_{16} = 3600_{90} \Rightarrow$$

$$3600_{90} + [2000]_{16} = 5600_{91} \Rightarrow$$

$$5600_{91} + [2000]_{16} = 7600_{92} \Rightarrow$$

$$7600_{92} + [2000]_{16} = 9600_{93} \Rightarrow$$

$$9600_{93} + [2000]_{16} = B600_{94} \Rightarrow$$

$$B600_{94} + [2000]_{16} = D600_{95} \Rightarrow$$

$$D600_{95} + [2000]_{16} = F600_{96} \Rightarrow$$

$$F600_{96} + [2000]_{16} = 1600_{97} \Rightarrow$$

$$1600_{97} + [2000]_{16} = 3600_{98} \Rightarrow$$

$$3600_{98} + [2000]_{16} = 5600_{99} \Rightarrow$$

$$5600_{99} + [2000]_{16} = 7600_{100} \Rightarrow$$

$$7600_{100} + [2000]_{16} = 9600_{101} \Rightarrow$$

$$9600_{101} + [2000]_{16} = B600_{102} \Rightarrow$$

$$B600_{102} + [2000]_{16} = D600_{103} \Rightarrow$$

$$D600_{103} + [2000]_{16} = F600_{104} \Rightarrow$$

$$F600_{104} + [2000]_{16} = 1600_{105} \Rightarrow$$

$$1600_{105} + [2000]_{16} = 3600_{106} \Rightarrow$$

$$3600_{106} + [2000]_{16} = 5600_{107} \Rightarrow$$

$$5600_{107} + [2000]_{16} = 7600_{108} \Rightarrow$$

$$7600_{108} + [2000]_{16} = 9600_{109} \Rightarrow$$

$$9600_{109} + [2000]_{16} = B600_{110} \Rightarrow$$

$$B600_{110} + [2000]_{16} = D600_{111} \Rightarrow$$

$$D600_{111} + [2000]_{16} = F600_{112} \Rightarrow$$

$$F600_{112} + [2000]_{16} = 1600_{113} \Rightarrow$$

$$1600_{113} + [2000]_{16} = 3600_{114} \Rightarrow$$

$$3600_{114} + [2000]_{16} = 5600_{115} \Rightarrow$$

$$5600_{115} + [2000]_{16} = 7600_{116} \Rightarrow$$

$$7600_{116} + [2000]_{16} = 9600_{117} \Rightarrow$$

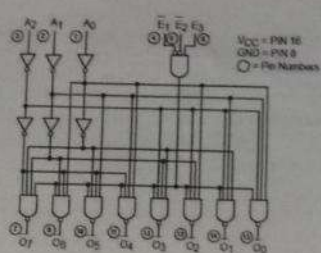
$$9600_{117} + [2000]_{16} = B600_{118} \Rightarrow$$

$$B600_{118} + [2000]_{16} = D600_{119} \Rightarrow$$

$$D600_{119} + [2000]_{16} = F600_{120} \Rightarrow$$

$$F600$$

Décodeur démultiplexeur 3 vers 8 SN74LS138 :

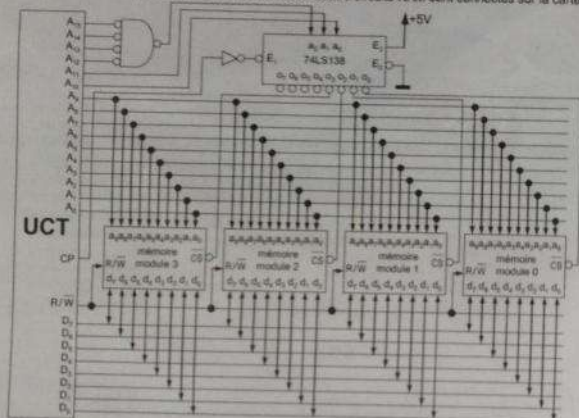


TRUTH TABLE													
INPUTS						OUTPUTS							
E1	E2	E3	A0	A1	A2	O0	O1	O2	O3	O4	O5	O6	O7
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	L	L	L	L	L	H	H	H	H	H	H	H
L	L	H	L	L	L	H	L	H	H	H	H	H	H
L	L	L	H	L	L	H	H	L	H	H	H	H	H
L	L	H	H	L	L	H	H	H	L	H	H	H	H
L	L	L	H	H	L	H	H	H	H	L	H	H	H
L	L	H	H	H	L	H	H	H	H	H	L	H	H
L	L	L	H	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

H = HIGH Voltage Level
L = LOW Voltage Level
X = Don't Care

5.4 Exercice 4

Considérons le schéma suivant, la carte UCT étant la carte unité centrale d'un système électronique composé d'un microprocesseur 8 bits de données. Quatre circuits RAM sont connectés sur la carte.



5.4.1 Déterminer la capacité de chaque RAM.

@ sur 10 bits donc 2^{10} mots de 8 bits soit 1ko et 8kbits. @RAM de \$0000 à \$0000₁₆ + \$400₁₆ - 1 = \$03FF
Supposons que l'UCT lance une opération de lecture des données situées à l'adresse 06A3 en base 16.

5.4.2 Déterminer s'il y a lieu le module de mémoire vive qui est lu.

@ μP = 06A3 soit

$$A_{15} A_{14} A_{13} A_{12} = 0_{16} = 0000_2 \Rightarrow A_{15} A_{14} A_{13} A_{12} = A_{15} + A_{14} + A_{13} + A_{12} = 0 = A_3 = 0$$

$$A_{11} A_{10} A_9 = 0_{16} = 0110_2 \Rightarrow A_{11} = 0, A_{10} = 1, A_9 = 0, A_8 = 1, A_7 = 0, A_6 = 1, A_5 = 0$$

Pour le décodeur : CBA = $A_2 A_1 A_0 = 001_2 = 1_{16} \Rightarrow O_7 = \overline{CS}_7 = 0 \Rightarrow$ sélection du module 1.

$$A_7 A_6 A_5 A_4 = 10_{16} = 1010_2$$

$$A_3 A_2 A_1 A_0 = 3_{16} = 0011_2$$

$$@M_1 = a_9 a_8 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0 = 1010100011_2 = 2A3_{16} = 2 \cdot 16^2 + 10 \cdot 16^1 + 3 \cdot 16^0 = 2 \cdot 256 + 10 \cdot 16 + 3 = 679_{10}$$

5.4.3 Déterminer les adresses pour chacun des modules.

$$\text{Module0 @ début : } \$0000_{16} = 0000_{16}$$

$$\text{Module0 @ fin : } \$03FF_{16} = 1023_{16}$$

$$\text{Module1 @ début : } \$0400_{16} = 1024_{16}$$

$$\text{Module1 @ fin : } \$07FF_{16} = 2047_{16}$$

$$\text{Module2 @ début : } \$0800_{16} = 2048_{16}$$

$$\text{Module2 @ fin : } \$0BFF_{16} = 3071_{16}$$

$$\text{Module3 @ début : } \$0C00_{16} = 3072_{16}$$

$$\text{Module3 @ fin : } \$0FFF_{16} = 4095_{16}$$

6 TD6

6.1 Exercice 1

Nous souhaitons fabriquer un plan mémoire à partir de mémoire contenant 2 ko par boîtier.

6.1.1 Indiquer combien de boîtiers sont nécessaires pour obtenir une mémoire de 2 kilo mots de 16 bits et réaliser le schéma de connexion de ces boîtiers (vous pouvez utiliser, si cela vous semble utile, des circuits logiques supplémentaires).

2k mots de 16 bits \Rightarrow 2 boîtiers 2k x 8bits, le bus de données est relié pour faire 2 x 8 bits = 16 bits.

6.1.2 Indiquer combien de boîtiers sont nécessaires pour obtenir une mémoire de 8 Kilo octets et réaliser le schéma de connexion de ces boîtiers (vous pouvez utiliser, si cela vous semble utile, des circuits logiques supplémentaires).

8k octets \Rightarrow 4 boîtiers x 2ko $\Rightarrow 2^{11}-1 \Rightarrow$ 11 bits d'adresse

8k = $8 \times 2^{10} \Rightarrow 2^{11}-1 \Rightarrow$ 11 bits d'adresse

[A12 \rightarrow A11] servent à sélectionner un des 4 boîtiers,

[A0 \rightarrow A10] sont connectés à tous les boîtiers.

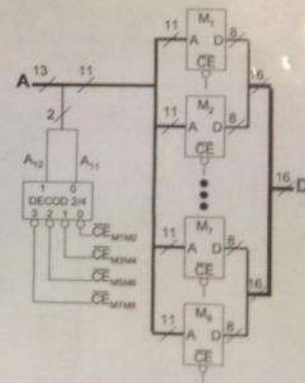
6.1.3 Indiquer combien de boîtiers sont nécessaires pour obtenir une mémoire de 8 Kilo mots de 16 bits et réaliser le schéma de connexion de ces boîtiers (vous pouvez utiliser, si cela vous semble utile, des circuits logiques supplémentaires).

8k mots de 16 bits $\Rightarrow 4 \times 2$ boîtiers (mélange des questions 1 et 2)

6.2 Exercice 2

Nous souhaitons réaliser un plan mémoire adressable par un microprocesseur sur 16 bits comprenant :

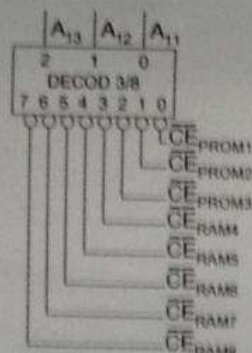
- 10 Kilo de RAM entre \$D800 et \$FFFF,
- 6 Kilo de PROM entre \$4000 et \$57FF.



Ces mémoires sont matériellement réalisées à l'aide de boîtiers de 2 Kilo Octets.

6.2.1 Déterminer les espaces d'adresses de chaque RAM et chaque PROM.

De \$4000 à \$47FF : $[A_{15}A_{14}A_{13}A_{12}A_{11}] = [0100\ 0]_2$	→	PROM1
De \$4800 à \$4FFF : $[A_{15}A_{14}A_{13}A_{12}A_{11}] = [0100\ 1]_2$	→	PROM2
De \$5000 à \$57FF : $[A_{15}A_{14}A_{13}A_{12}A_{11}] = [0101\ 0]_2$	→	PROM3
De \$D800 à \$DFFF : $[A_{15}A_{14}A_{13}A_{12}A_{11}] = [1101\ 1]_2$	→	RAM4
De \$E000 à \$E7FF : $[A_{15}A_{14}A_{13}A_{12}A_{11}] = [1110\ 0]_2$	→	RAM5
De \$E800 à \$EFFF : $[A_{15}A_{14}A_{13}A_{12}A_{11}] = [1110\ 1]_2$	→	RAM6
De \$F000 à \$F7FF : $[A_{15}A_{14}A_{13}A_{12}A_{11}] = [1111\ 0]_2$	→	RAM7
De \$F800 à \$FFFF : $[A_{15}A_{14}A_{13}A_{12}A_{11}] = [1111\ 1]_2$	→	RAM8



6.2.2 Proposer un schéma réalisant ce décodage en utilisant trois décodeurs à 3 entrées. Un décodeur permettra de sélectionner soit la RAM soit la PROM.

Les bits A_{15} , A_{14} et A_{13} des adresses binaires imposées pour accéder aux 8 boîtiers codent les nombres 0 à 7. Nous pouvons ainsi utiliser un décodeur 3 (binaire) vers 8 (unaire complémenté). Ces sorties peuvent piloter les entrées de validation complémentées des 8 boîtiers mémoire.

6.3 Exercice 3

La figure suivante représente le schéma d'un système à microprocesseur. Le décodeur découpe l'espace adressable par le microprocesseur en huit blocs.

6.3.1 Calculer l'étendue de chaque bloc et en déduire les adresses basses et hautes correspondantes.

Adressage de la PROM : A_{10} à $A_{11} \Rightarrow \$000$ à $\$FFF$

Adressage du contrôleur : $A_{16}A_{15} \Rightarrow \0 à $\$4$

Adressage de la RAM : A_{10} à $A_{11} \Rightarrow \$000$ à $\$7FF \Rightarrow 000\ 0000\ 0000$ à $111\ 1111\ 1111$

6.3.2 Quelles sont les adresses basses et hautes (vue du microprocesseur) des différents circuits utilisés ? Sont-elles uniques ?

Vu du Processeur :

La PROM a deux champs d'adresses \$E000 à \$EFFF ou \$F000 à \$FFFF selon la valeur de A_{12} qui n'est pas précisée ($111X\ 0000\ 0000\ 0000$ à $111X\ 1111\ 1111\ 1111$)

La RAM : A_{12} et A_{11} non précisées, $000X\ XXXX\ 0000\ 0000$ et $000X\ XXXX\ 1111\ 1111\ 1111$, 4 Champs adresses selon les valeurs de A_{11} et A_{12} : \$0000 à \$07FF, \$0800 à \$0FFF, \$1000 à \$17FF, \$1800 à \$1FFF

Le contrôleur $100X\ XXXX\ XXXX\ XXXX$ à $100X\ XXXX\ XXXX\ XXXX$, \$8000 à \$8FFF.

6.4 Exercice 4

Soit une mémoire centrale de 2 M octets où chaque octet est adressable séparément.

6.4.1 Calculer l'adresse, en octal du sixième élément d'un tableau dont l'adresse du premier élément est 778 et dont tous les éléments sont composés de 16 bits.

6.4.2 Calculer, en décimal, le nombre d'Octets précédant l'adresse 778

6.4.3 Calculer la taille de cette mémoire en l'exprimant en mots de 16 bits et en mots de 32 bits.