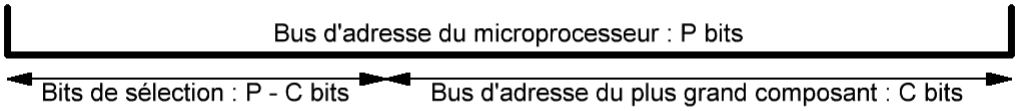
T.D. 6 – Corrigé
Décodage d'adresse

Exercice 1

Soit P, le nombre de bits d'adresse du microprocesseur, et C, le nombre de bits d'adresse du plus grand composant connecté au microprocesseur.

1. Quel est, en fonction de P et de C, le nombre total de composants que l'on peut connecter au microprocesseur avec un décodage linéaire ?



   Le nombre de bits de sélection disponibles est $P - C$.

   En décodage linéaire, on associe un bit de sélection par composant. **On peut donc connecter $P - C$ composants au microprocesseur**.

2. Même question avec un décodage par zone.

   En décodage par zone, on associe une combinaison de bits de sélection à un périphérique. **On peut donc connecter 2^{P-C} périphériques au microprocesseur**.

Exercice 2

On dispose d'une mémoire morte (ROM) de 8 Mib, d'une mémoire vive (RAM) de 8 Mib et de deux périphériques (P1 et P2) adressables respectivement sur 8 Kio et 4 Kio. On désire les rendre accessibles à un microprocesseur via les bus d'adresse (24 bits), de donnée (8 bits) et de commande. Les mémoires et les périphériques sont compatibles en largeur avec le microprocesseur. La ROM sera située dans les adresses les plus faibles, viendront ensuite la RAM, P1 et P2.

1. Donnez la taille du bus d'adresse de chaque mémoire et de chaque périphérique.

   Afin de déterminer la taille des bus d'adresse, il faut commencer par déterminer la **profondeur** des différents composants. Leur largeur étant l'octet, c'est **le nombre d'octets** de chaque composant qu'il faut déterminer. Concernant P1 et P2, leur capacité est exprimée en octet. C'est donc leur profondeur qui nous est directement donnée.

   - **ROM** : 8 Mib = (8 Mi / 8) octets = 1 Mio = 2^{20} octets → **20 fils d'adresse**
   - **RAM** : 8 Mib = (8 Mi / 8) octets = 1 Mio = 2^{20} octets → **20 fils d'adresse**
   - **P1** : 8 Kio = 2^{13} octets → **13 fils d'adresse**
   - **P2** : 4 Kio = 2^{12} octets → **12 fils d'adresse**

Dans un premier temps, c'est le mode linéaire qui sera utilisé.

2. Quels bits d'adresse vont servir au décodage et à quel composant seront-ils associés ?

   Le décodage de type linéaire associe un composant à un bit de sélection du microprocesseur. Les bits de sélection sont les bits de poids fort. Nous avons ici quatre composants à relier au microprocesseur, ce seront donc les quatre bits de poids fort du bus d'adresse du microprocesseur (*A23* à *A20*) qui serviront au décodage.

   La ROM étant située dans les adresses les plus faibles, il faut lui associer le bit *A20*. Viennent ensuite la RAM et les deux périphériques que l'on associe respectivement aux bits *A21*, *A22* et *A23*.

Bit de sélection	Composant associé
A20	ROM
A21	RAM
A22	P1
A23	P2

← La ROM doit être située dans les adresses les plus faibles.

3. En tenant compte du signal *AS* (*Address Strobe*) que fournit le microprocesseur et qui indique si la valeur sur son bus d'adresse est valide, donnez la fonction de décodage ; c'est-à-dire les expressions du *CS* de chaque composant relié au microprocesseur.

   Tant que la valeur présente sur le bus d'adresse n'est pas valide ($AS = 0$), aucun composant ne doit être activé.

   - $CS_{ROM} = AS.A20$
   - $CS_{RAM} = AS.A21$
   - $CS_{P1} = AS.A22$
   - $CS_{P2} = AS.A23$

4. Donnez la représentation de l'espace mémoire avec toutes les adresses remarquables.

   Il faut déterminer l'adresse la plus basse et l'adresse la plus haute qu'occupe chaque mémoire et chaque périphérique. Il faut pour cela déterminer les bits de poids fort (sélection) et de poids faible (adressage du composant) du bus d'adresse du microprocesseur pour chaque composant. S'il existe des bits inutilisés, qui ne servent ni à l'adressage, ni à la sélection, ils seront positionnés à 0.

   Par exemple pour la ROM :
   - *A20* est à 1 pour la sélectionner.
   - *A21*, *A22* et *A23* sont à 0 pour désactiver les autres composants.
   - Pour son adresse basse, on positionne ses 20 bits d'adresse à 0.
   - Pour son adresse haute, on positionne ses 20 bits d'adresse à 1.

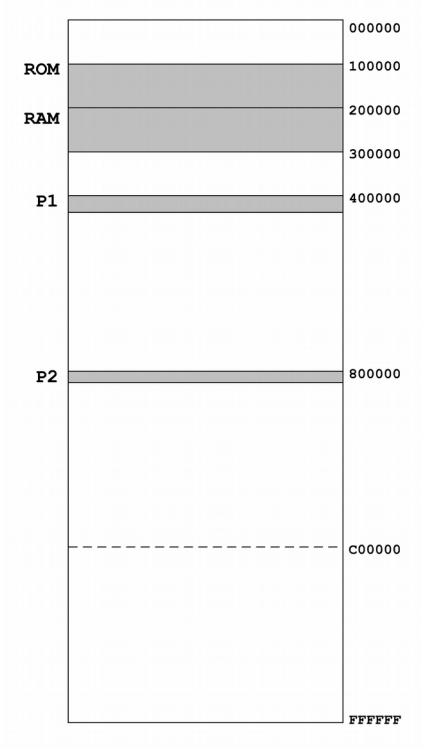
Ce qui donne pour tous les composants :

ROM basse : **0001 0000 0000 0000 0000 0000**$_2$ = 100000$_{16}$
ROM haute : **0001 1111 1111 1111 1111 1111**$_2$ = 1FFFFF$_{16}$

RAM basse : **0010 0000 0000 0000 0000 0000**$_2$ = 200000$_{16}$
RAM haute : **0010 1111 1111 1111 1111 1111**$_2$ = 2FFFFF$_{16}$

P1 basse : **0100** 0000 0000 **0000 0000 0000**$_2$ = 400000$_{16}$
P1 haute : **0100** 0000 0001 **1111 1111 1111**$_2$ = 401FFF$_{16}$

P2 basse : **1000** 0000 0000 **0000 0000 0000**$_2$ = 800000$_{16}$
P2 haute : **1000** 0000 0000 **1111 1111 1111**$_2$ = 800FFF$_{16}$



5. Quel est le principal défaut de ce type de décodage ?

   Le principal défaut de ce type de décodage tient au fait que plusieurs composants peuvent être activés en même temps. Cela peut entraîner un conflit d'accès sur le bus de donnée et causer des dommages.

6. Donnez les adresses interdites.

   Les adresses interdites sont celles qui créent des conflits. C'est-à-dire celles qui activent au moins deux composants en même temps. Ce sont donc toutes les adresses qui comportent au moins deux bits de sélection à 1.

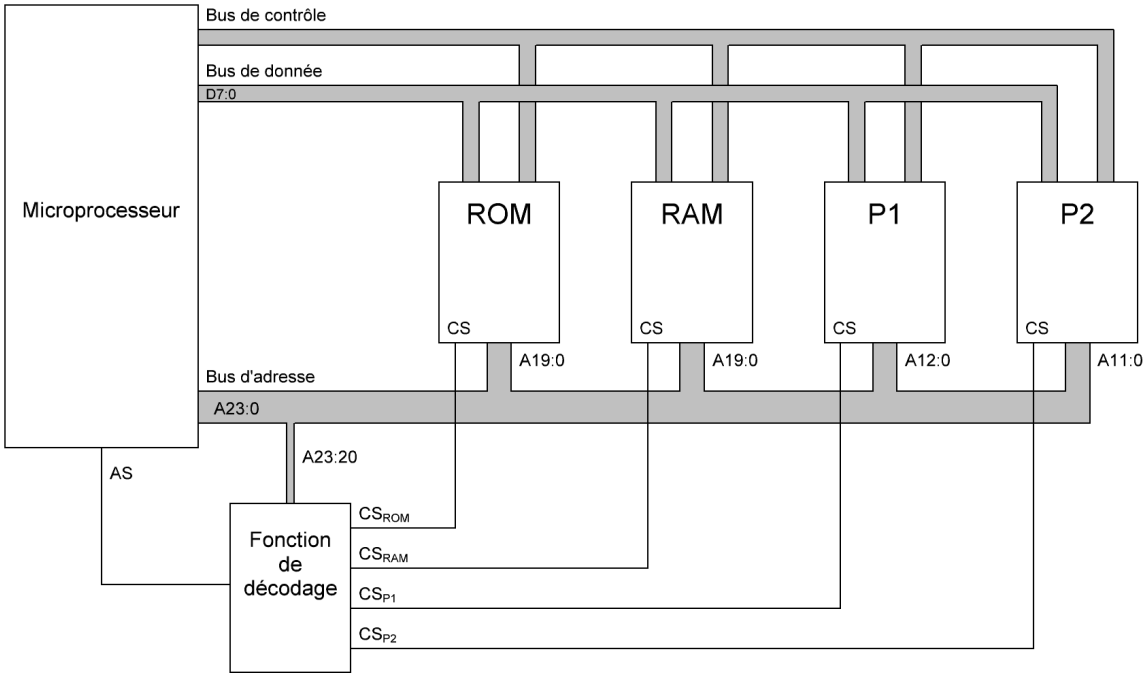
A23	A22	A21	A20	Composant
0	0	0	0	Aucun composant sélectionné
0	0	0	1	ROM seule
0	0	1	0	RAM seule
0	**0**	**1**	**1**	**Adresses interdites de 300000_{16} à $3FFFFF_{16}$**
0	1	0	0	P1 seul
0	**1**	**0**	**1**	**Adresses interdites de 500000_{16} à $7FFFFF_{16}$**
0	**1**	**1**	**0**	
0	**1**	**1**	**1**	
1	0	0	0	P2 seul
1	**0**	**0**	**1**	**Adresses interdites de 900000_{16} à $FFFFFF_{16}$**
1	**0**	**1**	**0**	
1	**0**	**1**	**1**	
1	**1**	**0**	**0**	
1	**1**	**0**	**1**	
1	**1**	**1**	**0**	
1	**1**	**1**	**1**	

7. Proposez une solution simple afin de supprimer les adresses interdites.

   Il faut modifier la fonction de décodage. L'entrée *CS* d'un composant doit être activée si son bit de sélection est à 1 et si les bits de sélection des autres composants sont à 0. Ainsi, aucun composant ne pourra être activé en même temps qu'un autre.

   - $CS_{ROM} = AS.\overline{A23}.\overline{A22}.\overline{A21}.A20$
   - $CS_{RAM} = AS.\overline{A23}.\overline{A22}.A21.\overline{A20}$
   - $CS_{P1} = AS.\overline{A23}.A22.\overline{A21}.\overline{A20}$
   - $CS_{P2} = AS.A23.\overline{A22}.\overline{A21}.\overline{A20}$

8. Donnez le schéma de câblage.



Pour la suite, on désire ajouter un périphérique P3 adressable sur 2 Kio (11 fils d'adresse).

9. Est-ce toujours possible en mode linéaire et pourquoi ?

   Le décodage de type linéaire associe un composant à un fil d'adresse du microprocesseur. Or ici, nous avons 5 composants, donc **5 fils du bus d'adresse sont nécessaires pour la sélection**. Les plus grands composants connectés au microprocesseur sont la ROM et la RAM avec 20 fils d'adresse. Sur les 24 fils du bus d'adresse du microprocesseur, il reste donc **4 fils disponibles** pour le décodage, **ce qui rend impossible l'utilisation du mode linéaire**.

On utilise maintenant le mode zone (toujours avec P3).
On travaillera de préférence avec le moins de zones possible.

10. Quels bits d'adresse vont servir au décodage et à quelles combinaisons seront associés les différents composants ?

    Pour connecter 5 composants en utilisant le moins de zones possible, il faut utiliser 3 bits de sélection afin de découper l'espace mémoire en 8 zones. Ce seront les 3 bits de poids fort : *A23*, *A22* et *A21*.

A23	A22	A21	Composant associé
0	0	0	ROM
0	0	1	RAM
0	1	0	P1
0	1	1	P2
1	0	0	P3

← La ROM doit être située dans les adresses les plus faibles.

11. Donnez la nouvelle fonction de décodage.

- $CS_{ROM} = AS.\overline{A23}.\overline{A22}.\overline{A21}$
- $CS_{RAM} = AS.\overline{A23}.\overline{A22}.A21$
- $CS_{P1} = AS.\overline{A23}.A22.\overline{A21}$
- $CS_{P2} = AS.\overline{A23}.A22.A21$
- $CS_{P3} = AS.A23.\overline{A22}.\overline{A21}$

12. Donnez la nouvelle représentation de l'espace mémoire avec toutes les adresses remarquables.

Il faut déterminer l'adresse la plus basse et l'adresse la plus haute qu'occupent chaque mémoire et chaque périphérique. Il faut pour cela déterminer les bits de poids fort (sélection) et de poids faible (adressage du composant) du bus d'adresse du microprocesseur pour chaque composant. S'il existe des bits inutilisés, qui ne servent ni à l'adressage, ni à la sélection, ils seront positionnés à 0.

Par exemple pour la ROM :
- *A21*, *A22* et *A23* sont à 0 pour la sélectionner (*cf.* tableau de la question 10).
- *A20* = 0 car il est inutilisé.
- Pour son adresse basse : on positionne ses 20 bits d'adresse à 0.
- Pour son adresse haute : on positionne ses 20 bits d'adresse à 1.

Ce qui nous donne pour tous les composants :

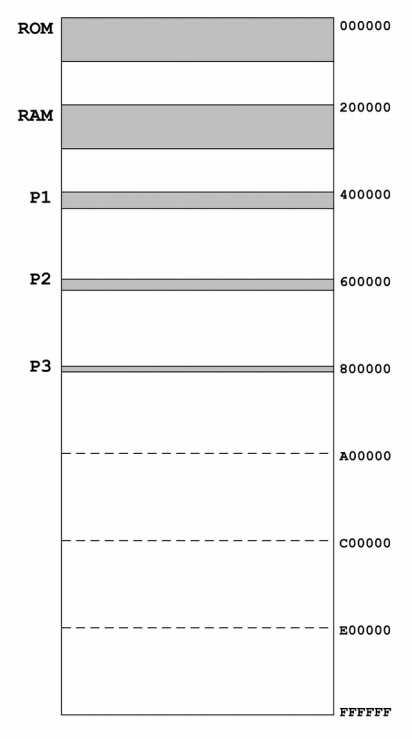
ROM basse : **0000 0000 0000 0000 0000 0000**$_2$ = 000000_{16}
ROM haute : **0000 1111 1111 1111 1111 1111**$_2$ = $0FFFFF_{16}$

RAM basse : **0010 0000 0000 0000 0000 0000**$_2$ = 200000_{16}
RAM haute : **0010 1111 1111 1111 1111 1111**$_2$ = $2FFFFF_{16}$

P1 basse : **0100** 0000 0000 **0000 0000 0000**$_2$ = 400000_{16}
P1 haute : **0100** 0000 0001 **1111 1111 1111**$_2$ = $401FFF_{16}$

P2 basse : **0110** 0000 0000 **0000 0000 0000**$_2$ = 600000_{16}
P2 haute : **0110** 0000 0000 **1111 1111 1111**$_2$ = $600FFF_{16}$

P3 basse : **1000** 0000 0000 **0000 0000 0000**$_2$ = 800000_{16}
P3 haute : **1000** 0000 0000 **0111 1111 1111**$_2$ = $8007FF_{16}$



13. Quelle est la redondance des différents composants ?

La redondance est le nombre de combinaisons que l'on peut réaliser avec les fils inutilisés du bus d'adresse du microprocesseur. Les fils inutilisés sont ceux qui n'appartiennent ni aux fils de sélection, ni aux fils du bus d'adresse des composants.

Fils inutilisés = 24 fils en tout – 3 fils de sélection – fils d'adresse du composant

- **ROM** : 24 – 3 – 20 = 1 fil inutilisé → $\mathbf{2^1 = 2}$
- **RAM** : 24 – 3 – 20 = 1 fils inutilisé → $\mathbf{2^1 = 2}$
- **P1** : 24 – 3 – 13 = 8 fils inutilisés → $\mathbf{2^8 = 256}$
- **P2** : 24 – 3 – 12 = 9 fils inutilisés → $\mathbf{2^9 = 512}$
- **P3** : 24 – 3 – 11 = 10 fils inutilisés → $\mathbf{2^{10} = 1\ 024}$

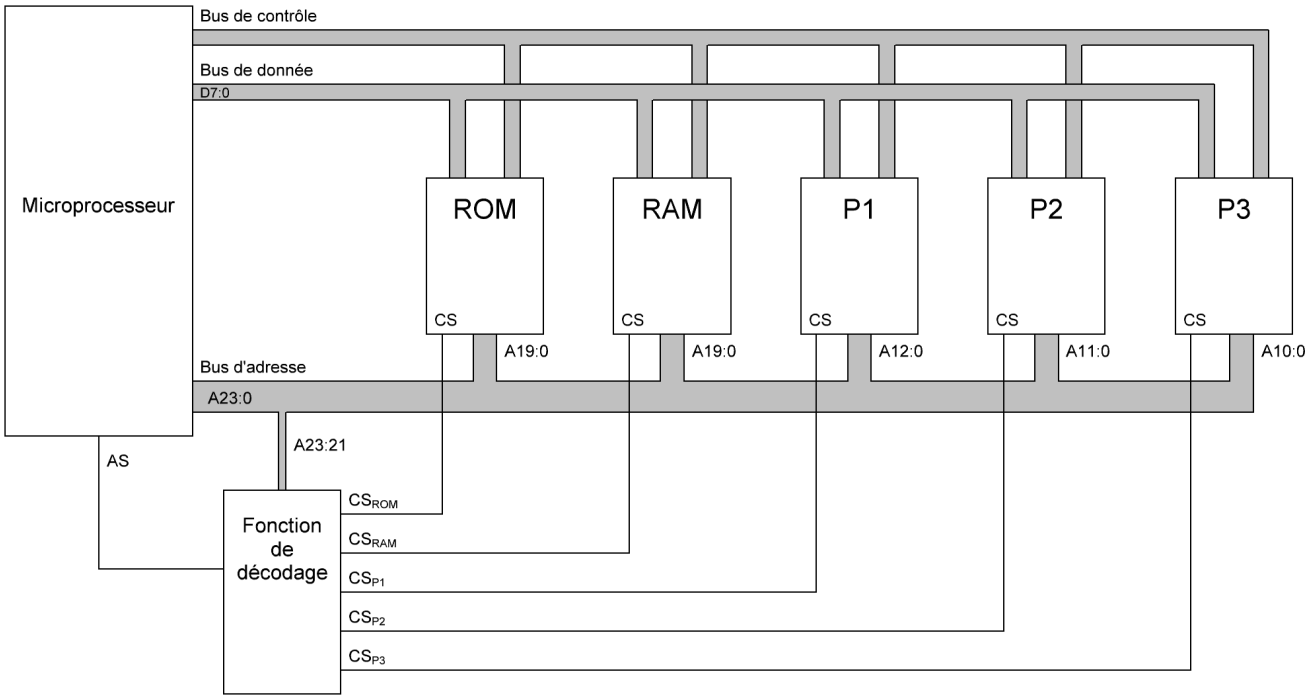
14. Donnez deux adresses différentes par composant pour sélectionner leur adresse $1F2_{16}$.

Pour sélectionner l'adresse $1F2_{16}$ d'un composant, il doit être sélectionné et son bus d'adresse doit contenir l'adresse $1F2_{16}$. Plusieurs combinaisons d'adresses, présentes sur le bus d'adresse du microprocesseur, permettent d'obtenir ce résultat. Il suffit de modifier les bits inutilisés.

Si l'on positionne le bit inutilisé *A20* à 0, puis à 1 pour la même sélection et la même adresse d'un composant, on obtient le tableau suivant :

	A20 = 0	**A20 = 1**
RAM	$0001F2_{16}$	$1001F2_{16}$
ROM	$2001F2_{16}$	$3001F2_{16}$
P1	$4001F2_{16}$	$5001F2_{16}$
P2	$6001F2_{16}$	$7001F2_{16}$
P3	$8001F2_{16}$	$9001F2_{16}$

15. Modifiez le schéma de câblage.



On désire maintenant remplacer la RAM de 1 Mio par une RAM de 4 Mio.

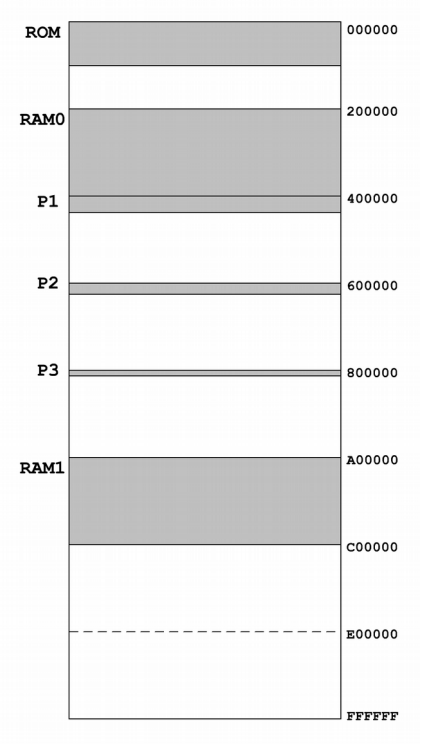
16. Est-ce toujours possible en mode zone et pourquoi ?

Le décodage de type zone associe un composant à une zone. Une zone correspond à une combinaison de fils d'adresse du microprocesseur. Or ici, nous avons 5 composants, donc au moins **3 fils du bus d'adresse sont nécessaires pour la sélection (8 zones)**. Le plus grand composant connecté au microprocesseur est maintenant la RAM avec 22 fils d'adresse (4 Mio). Sur les 24 fils du bus d'adresse du microprocesseur, il reste donc **2 fils disponibles** pour le décodage, **ce qui rend impossible l'utilisation du mode zone**.

Pour pouvoir brancher cette nouvelle RAM de 4 Mio, on propose d'utiliser une méthode dérivée du décodage par zone. Tout en gardant la configuration précédente de 8 zones de 2 Mio chacune, on utilisera deux zones pour la RAM. La première zone utilisée contiendra un premier morceau de la RAM que l'on nommera RAM0 et occupera la même zone que la RAM précédente. La seconde zone utilisée contiendra un second morceau de la RAM que l'on nommera RAM1 et occupera la zone située après P3.

17. Donnez la nouvelle représentation de l'espace mémoire.

ROM : 000000_{16} - $0FFFFF_{16}$
RAM0 : 200000_{16} - $3FFFFF_{16}$
P1 : 400000_{16} - $401FFF_{16}$
P2 : 600000_{16} - $600FFF_{16}$
P3 : 800000_{16} - $8007FF_{16}$
RAM1 : $A00000_{16}$ - $BFFFFF_{16}$



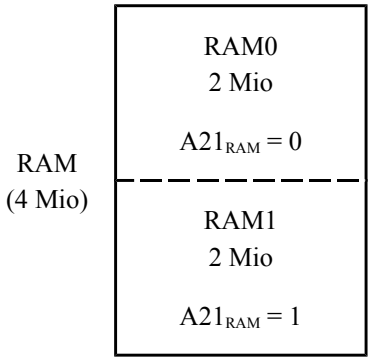
18. Sachant que le *CS* de la RAM doit être activé pour deux zones, donnez sa nouvelle expression.

$CS_{RAM} = AS.(\overline{A23}.\overline{A22}.A21 + A23.\overline{A22}.A21)$
$CS_{RAM} = AS.\overline{A22}.A21.(\overline{A23} + A23)$
$CS_{RAM} = AS.\overline{A22}.A21$

19. Si l'on se contente de cette nouvelle fonction de décodage, la sélection de la RAM se fait-elle correctement ?

Si l'on souhaite diviser la RAM de 4 Mio en deux RAM de 2 Mio, c'est le bit *A21* de la RAM ($A21_{RAM}$) qui permettra de distinguer si une adresse se trouve dans sa zone de poids faible (RAM0) ou dans sa zone de poids fort (RAM1).

Pour la suite, on notera :
- *A21* le bit d'adresse 21 du microprocesseur
- $A21_{RAM}$ le bit d'adresse 21 de la RAM.



En mode zone, les bits de poids faible du bus d'adresse du microprocesseur sont reliés à l'intégralité du bus d'adresse d'un composant. Dans le cas de la RAM nous avons donc les bits *A21:0* du microprocesseur qui sont reliés aux bits *A21:0* de la RAM et donc plus particulièrement : $A21_{RAM} = A21$ du microprocesseur.

Dans l'expression de l'entrée CS_{RAM}, on constate que la RAM est active uniquement lorsque $A21 = 1$. Or, si $A21 = 1$ et que $A21_{RAM} = A21$, c'est donc toujours la RAM1 qui est sélectionnée et jamais la RAM0.

Si l'on se contente de cette fonction de décodage, la sélection de la RAM ne se fait donc pas correctement.

20. Proposez une solution pour résoudre ce problème.

Pour résoudre ce problème, il faut positionner la valeur de $A21_{RAM}$ en fonction de la zone sélectionnée, c'est-à-dire en fonction des bits de sélection. Lorsque la RAM n'est pas active, la valeur sur son bus d'adresse est ignorée. On peut résumer tout cela dans le tableau ci-dessous :

A23	**A22**	**A21**	**Composant activé**	**$A21_{RAM}$**	
0	0	0	ROM	Φ	
0	0	1	RAM	**0**	→ RAM0
0	1	0	P1	Φ	
0	1	1	P2	Φ	
1	0	0	P3	Φ	
1	0	1	RAM	**1**	→ RAM1

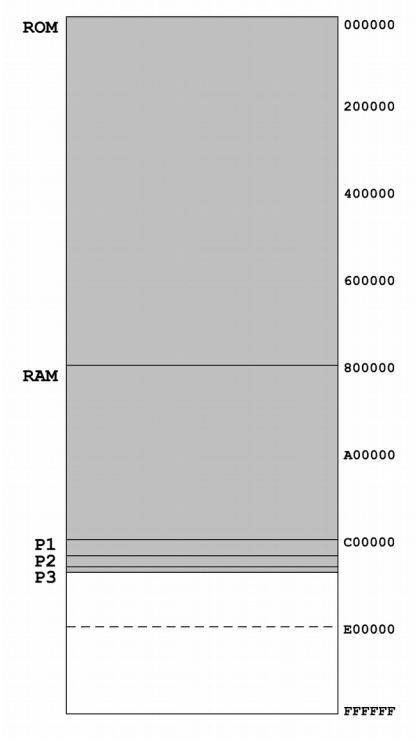
À partir de ce tableau, on déduit de manière évidente que la broche $A21_{RAM}$ doit être reliée à la broche *A23* du microprocesseur et non plus à la broche *A21* comme précédemment avec le mode zone.

Les expressions des différents *CS* restent inchangées.

On désire maintenant remplacer la ROM de 1 Mio par une ROM de 8 Mio. Pour pouvoir brancher cette nouvelle ROM, on propose d'utiliser une méthode consistant à ne pas laisser d'espace vide entre les différentes adresses de chaque composant. La ROM occupera le bas de l'espace mémoire, viendront ensuite la RAM (4 Mio), P1, P2 et P3.

21. Donnez la nouvelle représentation de l'espace mémoire.

ROM : 000000_{16} - $7FFFFF_{16}$
RAM : 800000_{16} - $BFFFFF_{16}$
P1 : $C00000_{16}$ - $C01FFF_{16}$
P2 : $C02000_{16}$ - $C02FFF_{16}$
P3 : $C03000_{16}$ - $C037FF_{16}$



22. Donnez la nouvelle fonction de décodage.

Il faut déterminer les bits d'adresse suffisamment significatifs qui permettront de sélectionner le bon composant en fonction de l'adresse présente sur le bus d'adresse. Nous allons pour cela réaliser une conversion hexadécimale-binaire des adresses déterminées à la question précédente en représentant par un caractère *phi* les bits d'adresse de chaque composant.

ROM	:	**0**	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	(23 fils d'adresse)
RAM	:	**1**	**0**	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	(22 fils d'adresse)
P1	:	**1**	**1**	0	0	0	0	0	0	0	0	0	0	**0**	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	(13 fils d'adresse)
P2	:	**1**	**1**	0	0	0	0	0	0	0	0	0	0	**1**	**0**	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	(12 fils d'adresse)
P3	:	**1**	**1**	0	0	0	0	0	0	0	0	0	0	**1**	**1**	0	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	Φ	(11 fils d'adresse)
		A 23	**A 22**											**A 13**	**A 12**													

Il suffit de remplacer les caractères *phi* par des 0 pour obtenir l'adresse la plus basse d'un composant et de les remplacer par des 1 pour obtenir l'adresse la plus haute.

Le premier bit significatif que l'on peut remarquer est le bit *A23*. En effet, si *A23* = 0 alors c'est la ROM qui doit être sélectionnée et si *A23* = 1 c'est à l'un des quatre autres composants d'être sélectionné. Le bit *A23* nous sert donc à différencier la sélection de la ROM des autres composants. On peut commencer à écrire la fonction de décodage sous la forme suivante :

- $CS_{ROM} = AS.\overline{A23}$
- $CS_{RAM} = AS.A23$
- $CS_{P1} = AS.A23$
- $CS_{P2} = AS.A23$
- $CS_{P3} = AS.A23$

Le prochain bit significatif est le bit *A22* qui, de la même manière que précédemment, nous permet de différencier la sélection de la RAM des autres composants. Si *A22* = 0, c'est la RAM qui doit être sélectionnée, sinon c'est soit P1, P2 ou P3. On peut maintenant écrire la fonction de décodage sous la forme suivante :

- $CS_{ROM} = AS.\overline{A23}$
- $CS_{RAM} = AS.A23.\overline{A22}$
- $CS_{P1} = AS.A23.A22$
- $CS_{P2} = AS.A23.A22$
- $CS_{P3} = AS.A23.A22$

En raisonnant exactement de la même façon avec les autres composants, on remarque que les prochains bits significatifs sont *A13* et *A12*, ce qui nous permet d'écrire la fonction de décodage complète suivante :

- $CS_{ROM} = AS.\overline{A23}$
- $CS_{RAM} = AS.A23.\overline{A22}$
- $CS_{P1} = AS.A23.A22.\overline{A13}$
- $CS_{P2} = AS.A23.A22.A13.\overline{A12}$
- $CS_{P3} = AS.A23.A22.A13.A12$

On peut remarquer que les numéros des fils d'adresse présents dans la fonction de décodage correspondent aux nombres de fils de chaque composant.