

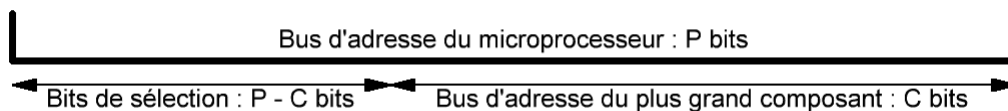
# T.D. 5 – Corrigé

## 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 :  $0010\ 0000\ 0000\ 0000\ 0000\ 0000_2 = 200000_{16}$

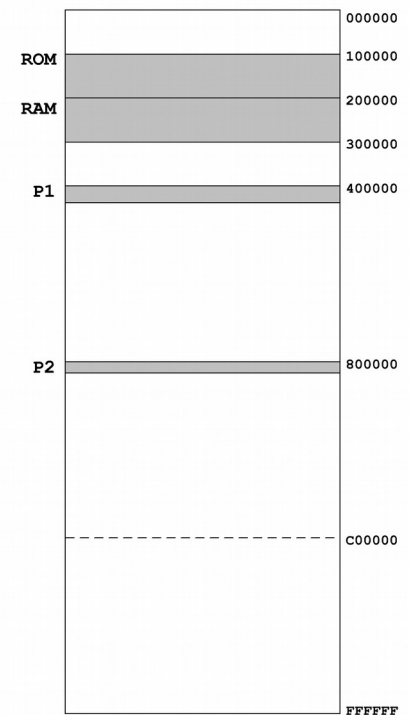
RAM haute :  $0010\ 1111\ 1111\ 1111\ 1111\ 1111_2 = 2FFFFFF_{16}$

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.

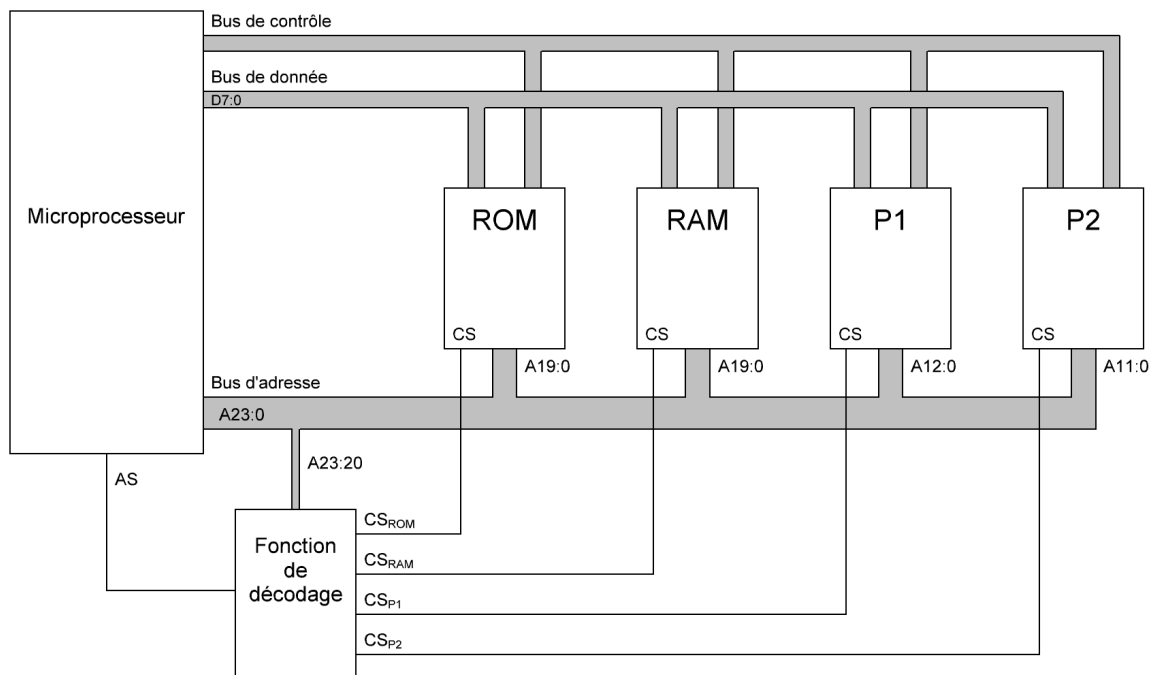
A23	A22	A21	A20	Composant
0	0	0	0	Aucun composant sélectionné
0	0	0	1	ROM seule
0	0	1	0	RAM seule
0	0	1	1	Adresses interdites de $300000_{16}$ à $3FFFFF_{16}$
0	1	0	0	P1 seul
0	1	0	1	Adresses interdites de $500000_{16}$ à $7FFFFF_{16}$
0	1	1	0	
0	1	1	1	
1	0	0	0	P2 seul
1	0	0	1	Adresses interdites de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

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

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.

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

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 :  $0000\ 1111\ 1111\ 1111\ 1111\ 1111_2 = 0FFFF_{16}$

RAM basse :  $0010\ 0000\ 0000\ 0000\ 0000\ 0000_2 = 200000_{16}$

RAM haute :  $0010\ 1111\ 1111\ 1111\ 1111\ 1111_2 = 2FFFF_{16}$

P1 basse :  $0100\ 0000\ 0000\ 0000\ 0000\ 0000_2 = 400000_{16}$

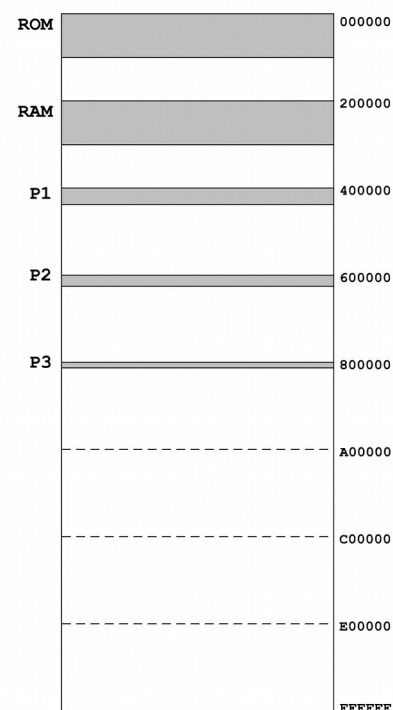
P1 haute :  $0100\ 0000\ 0001\ 1111\ 1111\ 1111_2 = 401FFF_{16}$

P2 basse :  $0110\ 0000\ 0000\ 0000\ 0000\ 0000_2 = 600000_{16}$

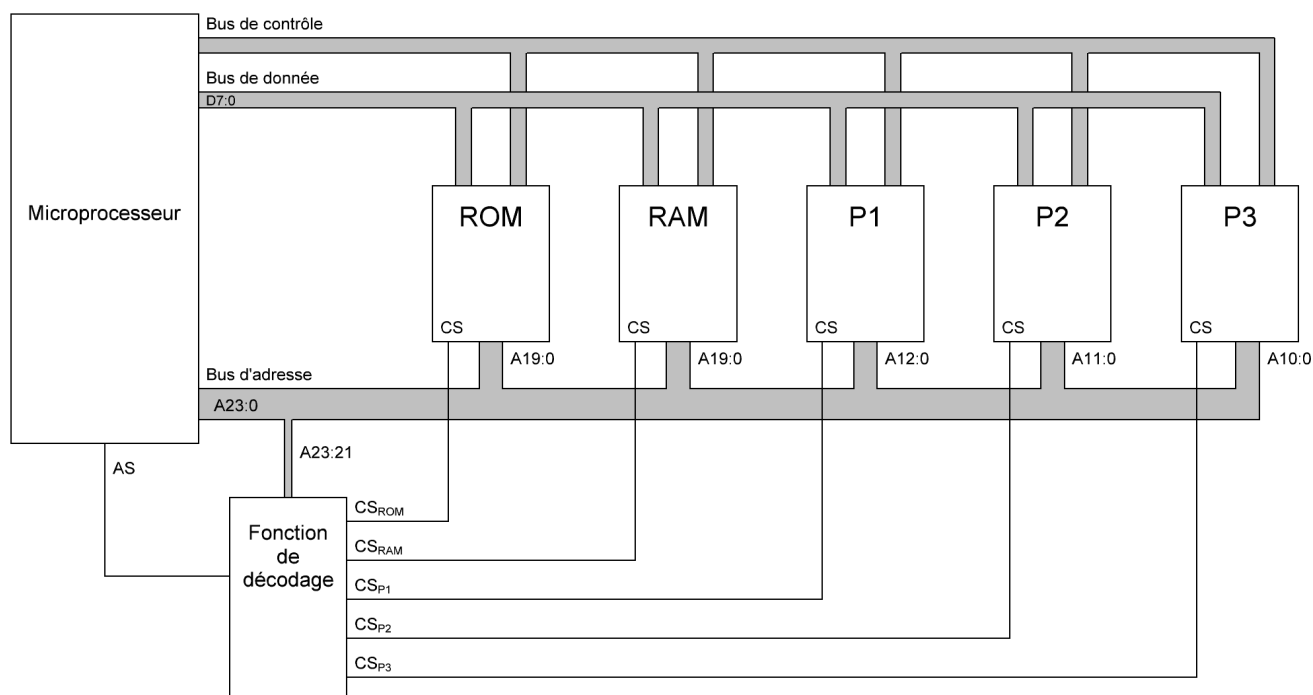
P2 haute :  $0110\ 0000\ 0000\ 1111\ 1111\ 1111_2 = 600FFF_{16}$

P3 basse :  $1000\ 0000\ 0000\ 0000\ 0000\ 0000_2 = 800000_{16}$

P3 haute :  $1000\ 0000\ 0000\ 0111\ 1111\ 1111_2 = 8007FF_{16}$



13. Modifiez le schéma de câblage.



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

14. 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**.