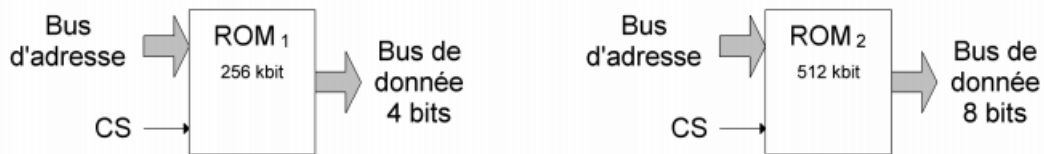


Architecture des Ordinateurs

TD6 : Mémoires

Exercice 2 :

Soit les deux mémoires de type ROM suivantes :



- 1) Combien peut-on former de mots de 4 bits avec la ROM₁ ?
- 2) Combien peut-on former de mots de 8 bits avec la ROM₂ ?
- 3) Quelle est la taille du bus d'adresse des deux types de ROM ?

1. Combien peut-on former de mots de 4 bits avec la ROM₁ ?

Il s'agit ici de déterminer la profondeur de la ROM₁.

$256 \text{ kbit} = 256/4 \text{ kmot de 4 bits} = 64 \text{ kmot de 4 bits.}$

On peut former 64 kmot de 4 bits avec la ROM₁.

2. Combien peut-on former de mots de 8 bits avec la ROM₂ ?

Il s'agit ici de déterminer la profondeur de la ROM₂.

$512 \text{ kbit} = 512/8 \text{ kmot de 8 bits} = 64 \text{ kmot de 8 bits.}$

On peut former 64 kmot de 8 bits (64 ko) avec la ROM₂.

3. Quelle est la taille du bus d'adresse des deux types de ROM ?

La taille du bus d'adresse d'une mémoire se détermine à partir de sa profondeur.

$64 \text{ kmot} = 2^{16} \text{ mots.}$

La taille du bus d'adresse des deux types de ROM est de 16 bits.

Exercice 3 :

La mémoire centrale peut être vue comme un large vecteur (tableau) de mots ou octets. Un mot mémoire stocke une information sur n bits. Chaque mot possède sa propre adresse. La mémoire peut contenir des programmes et les données utilisées par les programmes.

- 1) Sachant que le bus d'adresse du processeur est de 16 bits avec un alignement à l'octet (un mot = 1 octet), quelle est la taille de l'espace mémoire maximum que celui-ci peut adresser ? Quelles solutions existent pour adresser une plus grande zone mémoire ?

$$\text{Taille de la mémoire} = 1 \times 2^{16} = 2^{10} \times 2^6 = 2^6 \text{ Ko} = 64 \text{ Ko}$$

Les solutions : augmenter la taille du bus mémoire/ bus d'adresse

2) Soit une mémoire de 1Mo découpée en blocs de 128 Ko (taille d'un mot = 1 octet).

a. Calculer le nombre de blocs

$$1\text{MO} = 2^{20} \text{ octets}$$

$$128 \text{ KO} = 2^7 \times 2^{10} \text{ octets} = 2^{17} \text{ octets}$$

$$1\text{MO} / 128\text{KO} = 2^{20} / 2^{17} = 2^3 \text{ octets} \rightarrow 8 \text{ blocs}$$

b. Calculer les adresses de début et de fin de chaque bloc

Une adresse mémoire va être représentée sur 5 chiffres hexadécimale. En effet, le bus d'adresse a 20 lignes ($1\text{MO} = 2^{20} \text{ octets} \Rightarrow 20 \text{ lignes}$) et $20/4 = 5$ (car 1 chiffre hexa est codé sur 4 bits)
Donc la plus petite adresse $(00000)_{16}$, la plus grande adresse $(FFFFF)_{16}$

Adresse debut bloc N = adresse de fin bloc N-1 +1

Adresse fin de bloc = adresse debut + taille du bloc – 1

bloc	adresse debut (hexa)	adresse fin (hexa)
0	00000	00000+ 128k -1 00000+ 20000 - 00001 1FFFF
1	1FFFF+00001= 20000	20000+20000-00001 3FFFF
2	40000	5FFFF
3	60000	7FFFF
4	80000	9FFFF
5	A0000	BFFFF
6	C0000	DFFFF
7	E0000	FFFFF

Exemple pour bloc :

$$\text{Adresse fin} = (00000)_{16} + 128\text{ko} - 1 = 2^7 \times 2^{10} \text{ octets} - 1 = 2^{17} - 1 = 20000 - 1$$

$$2^{17} - 1 \rightarrow \text{en binaire} \rightarrow 100000000000000000 - 1 = 01 \ 1111 \ 1111 \ 1111 \ 1111 = 1FFFF$$