AdO – TD3 – Mémoires

8 bits = 1 octet = 2^3 ⑤
1 mot = 2 octets
= 16 bits

Exercice 1 :

① Mots de 8 bits (1 octet)

2^{16} adresses

Kilobit = 1Kb = 1Ko = 2^{10} octets
mégabit = 1Mb = 1Mo = 2^{20} octets
gigabit = 1Gb = 1Go = 2^{30} octets

Taille totale mémoire en Ko ?

TME = NA * TMO

Rappel : taille mémoire = Nombre d'adresses * taille d'un mot

↳ Capacité

$= \underbrace{2^{16} \times 1}_{\text{octets}} = 2^{10} \times 2^6 = 2^6\ Ko = 64\ Ko$

② Mots de 16 bits (2 octets)

8 bits pour l'adresse

Taille totale mémoire en Octets ?

↳ Nombre d'adresses = 2^8 = 256 octets

↳ $= 2^8 \times 2 = 256 \times 2 = 512$ octets

③ Taille totale mémoire : 32 Mo

Stocke des mots de 32 bits → 4 octets

ⓐ Nombre de bits pour représenter les adresses.

TM = NA * TMO → 32 Mo = NA * 4o

$32\ Mo = 32 \times 2^{20}$

$NA = \frac{32\ Mo}{4o} = \frac{32 \times 2^{20}}{4} = \frac{2^5 \times 2^{20}}{4}$

$= \frac{2^5 \times 2^{20}}{2^2} = 2^3 \times 2^{20} = 2^{23}$

$\log_2(2^{23}) = x$

$2^x = 2^{23}$

$x = 23$

Nous avons besoin de $\log_2(2^{23})$ = 23 bits

(b) adresses minimales

$0b\,\cancel{0}\underbrace{0000\ 0000\ 0000\ 0000\ 0000\ 0000}_{23\ \text{fois}}$ (23 bits) $= 0\times000000$

binaire

$(000000)_{16}$ hexa

adresses maximales

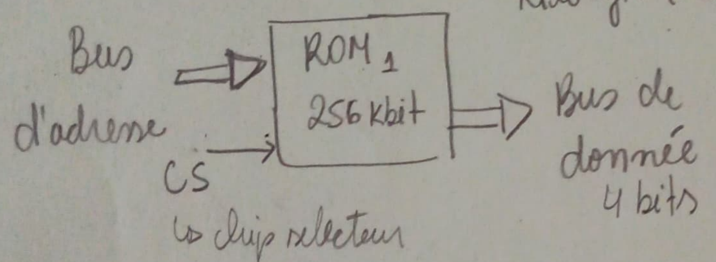
$0b\,\cancel{0}\underbrace{111\ 1111\ 1111\ 1111\ 1111\ 1111}_{23\ \text{fois}}$ (23 bits) $= 0\times7FFFFF$

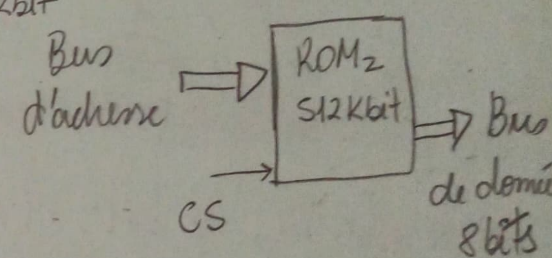
$(7FFFFF)_{16}$

Exercice 2 :

1 kilobit = Kbit ou Kb = 10^3 bit = 1000 bits ou 2^{10} bits

≠ Kilobyte (KB) = 8 Kbit





① Mots de 4 bits avec la ROM 1 ? → 256,000 → 64,000 mots de 4 bits

$2^{16} \times 4 = 2^{16} \times 2^2 = 2^{18}$

2^8 Kilobits = 256 Kb

Profondeur de ROM 1 $= \dfrac{256}{4}$ Kmots de 4 bits = 64K mots de 4 bits.

$2^n = NA = \dfrac{\text{Capacité}}{\text{TMO}}$

$2^{16} \times 8 = 2^{16} \times 2^3 = 2^{19}$

2^9 Kilobits
512 Kb

② Mots de 8 bits avec la ROM 2 ?

Profondeur de ROM 2 $= \dfrac{512}{8}$ Kmot de 8 bits = 64 Kmots de 8 bits.

③ Taille du bus d'adresse de ROM 1 et ROM 2 ?

→ se détermine à partir de la profondeur de ROM

64Kmot $= 2^6 * 2^{10}$ mots $= 2^{16}$ mots

La taille du bus d'adresse de ROM 1 et de ROM 2 est de 16 bits

Exercice 3 :

① bus d'adresse processeur : 16 bits avec alignement à l'octet

Taille de l'espace mémoire maximum ?

R: TME = NA * TMO

$= 2^{16} * 1 = 2^{10} * 2^{6} = 2^{6}$ Ko = 64 Ko

Solutions pour adresser une plus grande zone mémoire ?

R: Augmenter la taille du bus mémoire / bus d'adresse

② Mémoire 1Mo découpée en blocs de 128 Ko

1 mot = 1 octet = 8 bits

ⓐ Nombre de blocs ?

1Mo = 2^{20} octets → TME

128 Ko = 128 * 1Ko = $2^{7} * 2^{10}$ octets = 2^{17} octets → TB

$NB = \frac{TME}{TB} = \frac{1Mo}{128Ko} = \frac{2^{20}}{2^{17}} = 2^{3}$ octets → 8 blocs

→ TME = NA * 1 → $2^{20} = 2^{n} * 1$ → n = 20 bits

ⓑ calculer les adresses de début et de fin de chaque bloc

Le bus d'adresse a 20 lignes → 1Mo = 2^{20} ⇒ 20 lignes

une adresse va être représenté sur 5 chiffres hexadécimale

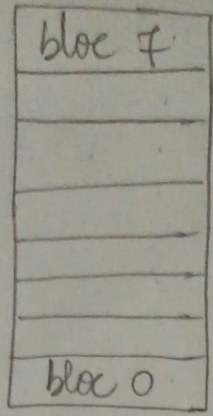
↳ 20/4 = 5, un chiffre hexa est codé sur 4 bits

la plus petite adresse $(00000)_{16}$ = $(0000\ 0000\ 0000\ 0000\ 0000)_2$

la plus grande adresse $(FFFFF)_{16}$

↳ = $(\underbrace{1111\ 1111\ 1111\ 1111\ 1111}_{20\ bits})_2$

8 blocs →



Taille du bloc = 128Ko

adresse début bloc N = adresse de fin bloc(N-1) + 1

Adresse fin de bloc = adresse début + taille du bloc -1

<u>Exemple:</u> Bloc 0

$$\text{Adresse fin de bloc} = (00000)_{16} + 128\text{Ko} - 1$$
$$= 0 + (128 * 1\text{Ko}) - 1$$
$$= (2^7 * 2^{10}) - 1$$
$$(131072)_{10} = \underset{\text{octets}}{2^{17}} - 1 = (20000 - 00001)_{16}$$

$$2^{17} - 1 \rightarrow \text{en binaire} \rightarrow (1\,0\,0000\,0000\,0000\,0000)_2 - 1$$
$$= (0\,1\,1111\,1111\,1111\,1111)_2 = (1FFFF)_{16}$$

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