

# Ing1 – Examen d'Architecture des Ordinateurs - Corrigé

## Exercice 1 – Espace adressable

1.a) Quel est l'espace adressable maximal de cette machine (en mots) ?  $2^{24}$  mots soit 16 M mots

1.b) Quelle est la plage d'adresses (en hexa) ? 0x000000 à 0xffffffff

1.c) ROM : 1Mmots=1.5Mio / RAM\_1 et RAM\_2 : 4Mmots=6Mio / RAM\_3 : 2Mmots=3Mio soit 15Mio de RAM

1.d) Donner les équations des signaux d'activation des 4 boîtiers mémoires et des 2 interfaces.

$$\begin{aligned}
 CS_{RO1} &= A_{23}A_{22} A_{21}\overline{A_{20}} \text{ (lu en sortie } y_{14} \text{ de DEC}_1\text{)} \\
 CS_{RA1} &= \overline{A_{23}} \\
 CS_{RA2} &= A_{23}\overline{A_{22}} \\
 CS_{RA3} &= A_{23}A_{22} \overline{A_{21}} \\
 CS_{SE} &= A_{23}A_{22} A_{21}A_{20} \overline{A_{19}}\overline{A_{18}} \overline{A_{17}}\overline{A_{16}} \overline{A_{15}}\overline{A_{14}} \overline{A_{13}}\overline{A_{12}} \overline{A_{11}}\overline{A_{10}} \overline{A_9}\overline{A_8} \overline{A_7}\overline{A_6} \overline{A_5}\overline{A_4} \text{ (comb. des 5 décodeurs)} \\
 CS_{PA} &= A_{23}A_{22} A_{21}A_{20} \overline{A_{19}}\overline{A_{18}} \overline{A_{17}}\overline{A_{16}} \overline{A_{15}}\overline{A_{14}} \overline{A_{13}}\overline{A_{12}} \overline{A_{11}}\overline{A_{10}} \overline{A_9}\overline{A_8} \overline{A_7}\overline{A_6} \overline{A_5}\overline{A_4} \text{ (comb. des 5 décodeurs)}
 \end{aligned}$$

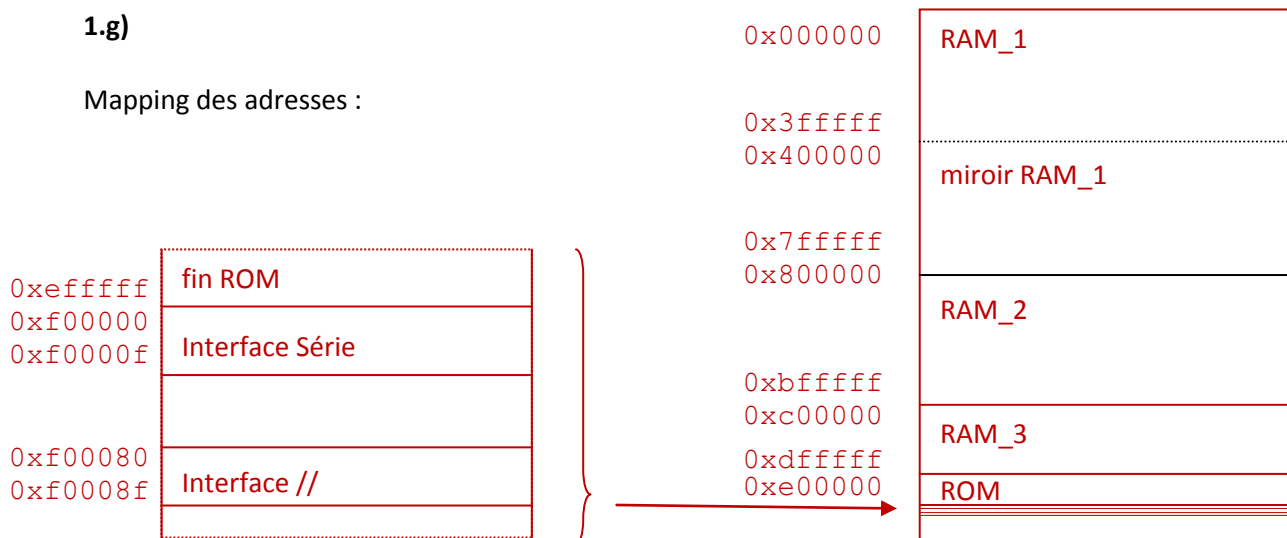
1.e)

|                     | Adresse minimale (hexa) | Adresse maximale (hexa)                  |
|---------------------|-------------------------|------------------------------------------|
| ROM                 | 0xe00000                | 0xffffffff                               |
| RAM 1               | 0x000000                | 0x3fffff<br>(0x40000-0x7fffff en miroir) |
| RAM 2               | 0x800000                | 0xbfffff                                 |
| RAM 3               | 0xc00000                | 0xdfffff                                 |
| Interface série     | 0xf00000                | 0xf0000f                                 |
| Interface parallèle | 0xf00080                | 0xf0008f                                 |

1.f) A quoi sert le fil d'adresse A22 non connecté à la RAM\_1 ? mettre une RAM jusqu'à 8M/12Mio

1.g)

Mapping des adresses :



## Exercice 2 – Entiers Relatifs

2.a) Quel est l'intervalle des nombres codables dans ce système ? -8 à +7

2.b) Donner le code binaire des entiers relatifs suivants : +2 (0010), -2(1110), +6(0110), -6 (1010)

3.c) Calculer les 3 additions -2 + 2, -2 + (-6), 2 + 6 en binaire complément à 2 (donner le détail des calculs).

$$\begin{array}{r}
 \boxed{11}10 \\
 1110 \\
 +0010 \\
 \hline
 0000
 \end{array}
 \quad
 \begin{array}{r}
 \boxed{11}10 \\
 1110 \\
 +1010 \\
 \hline
 1000
 \end{array}
 \quad
 \begin{array}{r}
 \boxed{01}10 \\
 0010 \\
 +0110 \\
 \hline
 1000
 \end{array}$$

OK : -2 + 2 = 0

OK : -2 + (-6) = -8

FAUX (2 dernières retenues différentes)

Autre justification pour le 1<sup>er</sup>, OK car 2 nombres de signes différents

### Exercice 3 - Assembleur

$$\text{Soit } v2 = v_{2_{15}}2^{15} + \dots + v_{2_8}2^8 + v_{2_7}2^7 + \dots + v_{2_0}2^0$$
$$v1 * v2 = \underbrace{(v_{2_{15}} \cdot 2^7 + \dots + v_{2_8} \cdot 2^0)}_{v2_h} 2^8 \cdot v1 + \underbrace{(v_{2_7} \cdot 2^7 + \dots + v_{2_0} \cdot 2^0)}_{v2_b} \cdot v1$$

Il nous faut donc calculer  $v2_b$  fois  $v1$ , ainsi que  $v2_h$  fois  $2^8 = 256$  fois  $v1$ .

1<sup>ère</sup> étape : calculer 256 fois  $v1$  en mémorisant au passage  $v2_b \cdot v1$

2<sup>ème</sup> étape : ajouter  $v2_h$  fois 256. $v1$

NB : on peut aussi utiliser le décalage (SRA) et utiliser le reste pour additionner ou pas  $2^k \cdot v1$ ,  $k$  étant l'étape de calcul

-- chargement de l'adresse du début des données en mémoire

LD IX, 0x0100    -- (IX) : partie basse de  $v1$  / (IX+1) : partie haute de  $v1$   
                  -- (IX+2) : partie basse de  $v2$  / (IX+3) : partie haute de  $v2$   
                  -- (IX+4) : partie basse de produit / (IX+5) : partie haute de produit

LD B, (IX+2)    // compteur de boucle B <-  $v2_b$

-- initialiser la somme (DE) à 0

LD D, 0        // partie basse

LD E, 0        // partie haute

--  $v2_b$  fois additionner  $v1$  à DE

boucle :    LD A, D        // addition partie basse :  $D \leftarrow D + v1_b$

ADD A, (IX)

LD D, A

LD A, E        // addition partie haute :  $E \leftarrow E + v1_h$

ADC A, (IX+1)

LD E, A

DJNZ boucle

LD (IX+4), D    // sauvegarde  $v2_b \cdot v1$  de DE vers la mémoire (variable produit)

LD (IX+5), E

--  $(256 - v2_b)$  fois additionner  $v1$  à DE pour obtenir  $2^8 \cdot v1$

LD A, 256       // on calcule le  $(256 - v2_b) \rightarrow B$  (compteur de boucle)

SUB (IX+2)

LD B, A

boucle2 :    LD A, D        // addition partie basse :  $D \leftarrow D + v1_b$

ADD A, (IX)

LD D, A

LD A, E        // addition partie haute :  $E \leftarrow E + v1_h$

ADC A, (IX+1)

LD E, A

DJNZ boucle2

-- DE contient maintenant  $2^8 \cdot v1$         // FIN 1<sup>ère</sup> Partie

LD B, (IX+3)    // compteur de boucle B <-  $v2_h$

-- boucle n°3 :  $v2_h$  fois additionner  $2^8 \cdot v1$  à la variable produit en mémoire

boucle3 :    LD A, (IX+4)    // ajout des parties basses de produit (mémoire) et  $2^8 \cdot v1$

ADD A, D

LD (IX+4), A

LD A, (IX+5)    // ajout des parties hautes de produit (mémoire) et  $2^8 \cdot v1$

ADC A, E

LD (IX+5), A

DJNZ boucle3

-- le résultat est bon en mémoire : produit =  $v1 * v2$

## Exercice 4 – VHDL et Chronogramme

Dessiner le chronogramme de simulation sur 10 cycles d'horloge pour les signaux clock, reset, a1, b1, s1, a2, b2, s2.

