

## Série d'exercices 4: Interfaces Entrées/Sortie

### Exercice 1 : Distinction entre E/S mappées et E/S isolées

Expliquez la différence entre l'adressage des E/S mappées en mémoire et des E/S isolées dans le 8085. Donnez les mnémotechniques des instructions utilisées pour chaque mode.

#### Solution :

1. **E/S mappées en mémoire** : Les périphériques sont vus comme des cellules mémoire. On utilise les instructions MOV, LDA, STA, MVI, etc. Toute l'espace mémoire (64 Ko) est partagé.
2. **E/S isolées** : Espace d'adressage séparé (256 ports, adresses 8 bits). Instructions spécifiques :
  - a. IN 8-bit-port (lit un port)
  - b. OUT 8-bit-port (écrit sur un port)Les signaux IO/M = 1 pour E/S, 0 pour mémoire.

### Exercice 2 : Décodage d'adresses pour un port de sortie

On veut adresser un port de sortie 8 bits à l'adresse 0F8H en E/S isolée. Proposez un circuit de décodage simple en utilisant des portes logiques et expliquez la séquence d'écriture.

#### Solution :

- Port 0F8H = 1111 1000 en binaire.
- En E/S isolée, les bits A7..A0 du bus d'adresses sont utilisés.
- Pour décoder 1111 1000 :
  - A7=1, A6=1, A5=1, A4=1, A3=1, A2=0, A1=0, A0=0.
  - Utiliser une porte AND à 8 entrées avec des inverseurs sur A2,A1,A0.
- Signal IO/M=1 (E/S) combiné avec l'AND pour valider le chip select du port.

#### Séquence d'écriture :

assembly

MVI A, 55H ; donnée à envoyer

OUT 0F8H ; écrit sur le port

### Exercice 3 : Interface d'entrée avec temporisation

Un capteur est connecté au bit D0 d'un port d'entrée (port 10H). Écrire un programme 8085 qui attend que ce bit passe à 1, puis incrémente une variable mémoire et recommence.

#### Solution :

assembly

```
WAIT: IN 10H ; lire le port
      ANI 01H ; masquer D0
      JZ WAIT ; boucle si D0=0
      LDA VAR ; variable mémoire
      INR A
      STA VAR
      JMP WAIT
```

**Explication :** ANI 01H met le flag Z à 1 si D0=0. On boucle tant que Z=1.

### Exercice 4 : Interface de sortie avec afficheur 7 segments

Un afficheur 7 segments (anode commune) est connecté au port de sortie 20H.

Table de conversion pour les chiffres hexadécimaux (a=LSB, g=MSB) :

0=3FH, 1=06H, 2=5BH, 3=4FH, 4=66H, 5=6DH, 6=7DH, 7=07H, 8=7FH, 9=6FH, A=77H, B=7CH, C=39H, D=5EH, E=79H, F=71H.

Écrire un programme qui affiche successivement 0 à F en boucle, avec un délai.

#### Solution :

assembly

```
MVI B, 00H ; compteur
TABLE: LXI H, SEGTAB
      MOV A, B
      ADD L ; A = index
      MOV L, A
      MOV A, M ; code 7seg
      OUT 20H
      CALL DELAY
      INR B
      MOV A, B
      CPI 10H
```

JNZ TABLE

JMP 00H ; boucle infinie

SEGTAB: DB 3FH,06H,5BH,4FH,66H,6DH,7DH,07H

DB 7FH,6FH,77H,7CH,39H,5EH,79H,71H

DELAY: LXI D, 0FFFFH ; délai simple

LOOP: DCX D

MOV A, D

ORA E

JNZ LOOP

RET

## Exercice 5 : Lecture d'un commutateur et commande de LED

Un commutateur 8 bits est connecté au port d'entrée 30H. Un ensemble de 8 LED est connecté au port de sortie 31H. Écrire un programme qui lit en permanence le commutateur et allume les LED selon la même valeur lue.

### Solution :

assembly

LOOP: IN 30H ; lire commutateur

OUT 31H ; envoyer aux LED

JMP LOOP

**Remarque :** Aucune conversion nécessaire, les 8 bits passent directement.

## Exercice 6 : Décodage d'adresses pour 2 ports en E/S mappés

### Énoncé :

On utilise l'espace mémoire de 0000H à FFFFH. On veut placer un port d'entrée à l'adresse 8000H et un port de sortie à l'adresse 8001H. Proposez un circuit de décodage simple avec un décodeur 3 vers 8 (ex : 74138) et des portes logiques.

### Solution :

- Adresses 16 bits : 8000H = 1000 0000 0000 0000  
8001H = 1000 0000 0000 0001
- Bits A15..A13 = 100 (binaire) → 4 en décimal.
- Utiliser le décodeur 74138 :
  - A15 → G2A, A14 → G2B, A13 → C.
  - G1 relié à Vcc.
  - La sortie Y4 (active low) sélectionne la zone 8000H-8FFFH.



- Pour distinguer entre entrée/sortie, utiliser A0 :
  - A0=0 → port d'entrée (actif sur RD)
  - A0=1 → port de sortie (actif sur WR)
- Ajouter IO/M=0 (mappée mémoire) en validation supplémentaire.

### Schéma fonctionnel :

text

A15 --- G2A (74138)

A14 --- G2B

A13 --- C

A12-A1 --- (non connectés pour ce décodage)

A0 --- sélection E/S

RD --- AND avec (A0=0) → CS\_entrée

WR --- AND avec (A0=1) → CS\_sortie

### Programme d'exemple :

assembly

**LDA** 8000H ; lit port entrée

**STA** 8001H ; écrit sur port sortie