

TD N° 2

Exercice 1 :

Soit un microprocesseur ayant un bus d'adresses de 16 bits, pouvant accéder à une donnée de 8 bits.

1. Quelle est la capacité d'adressage totale ?
2. Quelle est la capacité de cette mémoire ?
3. Représenter la structure de cette mémoire (adresse de début et adresse de fin en hexadécimal).
4. Mêmes questions pour un bus de données de 16 bits et un bus d'adresse de 20 bits

Exercice 2 :

Soit une mémoire de 64 Mo organisée en cases de 16 bits.

1. Déterminer le nombre de lignes d'adresses nécessaires.
2. Si l'adresse de début est $(0)_{10}$, déterminer l'adresse de fin en Hexadécimal.

Exercice 3 :

1. Si l'intervalle des adresses d'une mémoire va de 0000H à FFFE_H.
Combien cette mémoire a de cases ?
2. Si une mémoire possède 5120 emplacements en mémoire.
Donner l'intervalle (début et fin) de ces adresses exprimées en hexadécimal.
3. Si l'intervalle des adresses d'une mémoire va de 0531H à F20DH.
Combien cette mémoire a de cases ?

Exercice 4 :

1. Donner les adresses physiques des mémoires telles que les adresses logiques sont comme suit :
3500 : AB00 1CD0 : 1111 0022 : FFFF
2. Proposer une (des) adresse(s) segment : offset pour les mémoires d'adresse physique :
10000 FFFFF 00000

Exercice 5 :

Soit un processeur disposant de 3 registres 8 bits, un bus de données 8 bits, un bus d'adresse de 4 bits. Le jeu d'instruction du processeur est composé de 4 Instructions : Mov (Affectation), Add (Addition), OR (Ou logique), ET (Logique). Ces Instructions utilisent deux opérandes 8 bits (Reg. ou mémoire), exemple :

MOV Reg1, [adr] ; Lis l'emplacement mémoire adr et stocke le résultat dans Reg1

ADD [adr], Reg2 ; Ajoute le contenu de l'adresse à Reg2 et stocke résultat dans adr.

Or Reg1, Reg2 ; Ou logique entre Reg1 et Reg2, résultat dans Reg1

On suppose que le premier opérande est l'opérande destination. On suppose aussi qu'on ne peut avoir deux opérandes mémoire dans une même instruction.

1. Combien de bits sont nécessaires pour coder les opérations
 2. Combien de bits sont nécessaires pour coder les registres
 3. Les combinaisons possibles des opérandes sont appelées mode d'adressage, combien de modes d'adressage peut-t-on avoir, donnez leurs codes machine.
 4. Si les adresses sont codées par leur valeur et que le code chaque instruction est composée de 4 champs : **opcode modeAdressage. registre mémoire**, donnez le code machine de chaque instruction (**avec adr = 3**).
 5. Si la première instruction est stockée à l'adresse 0, donnez le contenu des 3 premières adresses.
 6. En reprenant les mêmes instructions de l'exemple précédents, et si on fait les hypothèses simplificatrices suivantes :
 - Le microprocesseur met le même temps t_i pour décoder et exécuter chaque instruction ;
 - Un accès à la mémoire nécessite t_m ;
 - Les codes des instructions sont au départ en mémoire.
- a. Donnez le temps d'exécution de chaque instruction
 - b. Calculer le temps d'exécution de ce programme en fonction de t_i et t_m . On donne : Fréquence processeur = 100 Mhz, temps d'accès mémoire = 100 ns.