

ING1 - Architecture des Ordinateurs

Examen - Durée : 2h

Seul document autorisé : mémento Z80 sans annotations. Calculatrice non programmable autorisée.

Exercice 1 :

On considère une machine de Von Neumann qui possède une mémoire constituée de mots de 8 bits et adressée sur 24 bits.

1. Quelle est la taille maximale de la mémoire en mots ? en octets (utiliser l'unité appropriée : octet, Ko, Mo, To, Po) ?
2. Quelle est la plage d'adresses dont vous disposez ? Exprimer votre réponse en décimal, binaire et hexadécimal.
3. Quelle est la taille des registres MAR et MDR ?
4. On considère les 3 instructions suivantes qui chargent une donnée de 8 bits dans le registre R1 :
 - LOAD R1, 127 ; adressage immédiat
 - LOAD R1, (127) ; adressage direct
 - LOAD R1, ((127)) ; adressage indirect

On suppose que la taille des instructions est fixe sur 16 bits. Pour chaque instruction, donner, en le justifiant, le nombre d'accès mémoire nécessaire à son exécution ?

Réponse _____

1. une adresse sur 24 bits permet d'adresser au maximum 2^{24} mots de 1 octet soit $2^{24} \text{ octets} = 2^{14} \text{ Ko} = 2^4 \text{ Mo} = 16 \text{ Mo}$
2. plage d'adresses : 0 à $2^{24} - 1 = 16777215$; $000000000000000000000000_{(2)}$ à $111111111111111111111111_{(2)}$; $000000_{(16)}$ à $FFFFFF_{(16)}$
3. registre d'adresse MAR : 24 bits ; registre de donnée MDR : 8 bits.
4. Nombre d'accès mémoire :
 - LOAD R1, 127 ; 2 pour charger l'instruction
 - LOAD R1, (127) ; 3 : 2 pour charger l'instruction et 1 pour charger la donnée à l'adresse 127
 - LOAD R1, ((127)) ; 6 : 2 pour charger l'instruction, 3 pour charger l'adresse (de la donnée) contenue dans la mémoire à l'adresse 127 (une adresse sur 24 bits tient sur 3 mots mémoire) et 1 pour charger la donnée contenue à cette adresse.



Exercice 2 :

Faites le schéma de la mémoire précédente, en la considérant de taille maximale par rapport aux adresses de 24 bits. Fabriquez votre mémoire à partir de boîtiers mémoires plus petits avec des adresses sur 22 bits et des mots de 8 bits. Câblez uniquement les broches d'adressage, de données et d'autorisation.

Réponse _____

On utilise 4 boîtiers mémoires. Le bus de données est connecté sur tous les boîtiers. Les poids 22 poids faibles de l'adresse (21 à 0) sont connectés sur les broches d'adressage des 4 boîtiers. Les 2 bits de poids forts de l'adresse passe à travers un décodeur qui activera un seul boîtier à la fois. Ce décodeur est activé par le signal d'activation de la mémoire.



Exercice 3 :

Calculer en complément à 2 sur 8 bits les additions suivantes : $122 + (-7)$; $(-111) + (-17)$ sur des entiers relatifs. Vous préciserez si le résultat est correct ou s'il y a dépassement de capacité.

Réponse _____

Codage des nombres en binaire complément à 2 :

$$122_{(10)} = 01111010_{(2)}$$

$$7_{(10)} = 0000111_{(2)} ; C1 : 11111000 ; C2 : 11111001$$

$$111_{(10)} = 01101111_{(2)} ; C1 : 10010000 ; C2 : 10010001$$

$$17_{(10)} = 00010001_{(2)} ; C1 : 11101110 ; C2 : 11101111$$

$$122 + (-7) : \begin{array}{r} 01111010 \\ +11111001 \\ \hline 101110011 \end{array}. \text{ Sur 8 bits, on obtient bien le bon résultat } 01110011_{(2)} = 115_{(10)}.$$

$$(-111) + (-17) : \begin{array}{r} 10010001 \\ +11101111 \\ \hline 110000000 \end{array}. \text{ Sur 8 bits, il nous reste } 10000000 \text{ qui est bien la représentation de } -128, \text{ plus petit nombre représentable sur 8 bits en C2 } (10000000 - 1 = 01111111 ; C1 : 10000000_{(2)} = 128_{(10)}).$$



Exercice 4 :

On suppose que la mémoire de notre machine est organisée en blocs de 16 mots. Pour améliorer les performances, on ajoute une mémoire cache associative par ensemble de 32 Ko.

1. Combien de blocs peut contenir la mémoire cache ?

Réponse _____

Le cache contient 2^{15} octets = $2^{11} * 2^4$ octets = 2^{11} blocs.



2. En considérant, qu'un ensemble peut accueillir plus qu'un seul bloc, combien d'ensembles au maximum peut-on avoir ?

Réponse _____

La taille minimale d'un ensemble est de 2 blocs, ce qui nous donne 2^{10} ensembles au maximum.



3. Prenons l'exemple d'une mémoire cache organisée en ensembles de 8 blocs.
(a) Combien avez-vous d'ensembles ?

Réponse _____

On a $2^{11} = 2^8 * 2^3$ blocs, soit $2^8 = 256$ ensembles.



- (b) À partir de l'adresse (sur 24 bits) d'une donnée en mémoire, expliquer à l'aide du schéma suivant, comment trouver :

- le numéro du mot dans le bloc
- le numéro du bloc dans la mémoire principale
- le numéro de l'ensemble qui va accueillir le bloc de cette donnée

- l'information additionnelle (TAG) que vous devez ajouter à un bloc de donnée pour retrouver la provenance de celui-ci.

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Réponse _____

- 4 bits de poids faible : numéro du mot dans le bloc
- 20 bits de poids forts : numéro de bloc dans la mémoire
- 8 bits du poids 11 à 4 : numéro d'ensemble, correspondant au modulo 2^8 du numéro d'ensemble
- 12 bits du poids 23 à 12 : TAG de reste d'ensemble



- (c) Chaque ensemble est géré en LRU, chaque bloc étant estampillé par une date sur 3 bits (la valeur 0 désigne le plus récemment utilisé). À partir de la configuration de départ suivante, montrez l'évolution de cet ensemble en fonction des accès aux données d'adresses $012A0C_{(16)}$, $747A00_{(16)}$, $CBAA07_{(16)}$, $ABCA0D_{(16)}$, $075FA0_{(16)}$. Mettez en évidence les défauts de cache.

numéro bloc initial	LRU	accès $012A0C_{(16)}$	accès $747A00_{(16)}$	accès $CBAA07_{(16)}$	accès $ABCA0D_{(16)}$	accès $075FA0_{(16)}$
$012A0_{(16)}$	7					
$FFAA0_{(16)}$	3					
$ABCA0_{(16)}$	5					
$C0DA0_{(16)}$	4					
$DACA0_{(16)}$	0					
$FABA0_{(16)}$	2					
$75FA0_{(16)}$	1					
$C4CA0_{(16)}$	6					

Réponse _____

numéro bloc initial	LRU	accès $012A0C_{(16)}$	accès $747A00_{(16)}$	accès $CBAA07_{(16)}$	accès $ABCA0D_{(16)}$	accès $075FA0_{(16)}$
$012A0_{(16)}$	7	$012A0_{(16)}$ 0	$012A0_{(16)}$ 1	$012A0_{(16)}$ 2	$012A0_{(16)}$ 3	idem
$FFAA0_{(16)}$	3	$FFAA0_{(16)}$ 4	$FFAA0_{(16)}$ 5	$FFAA0_{(16)}$ 6	$FFAA0_{(16)}$ 7	idem
$ABCA0_{(16)}$	5	$ABCA0_{(16)}$ 6	$ABCA0_{(16)}$ 7	$CBAA0_{(16)}$ 0	$CBAA0_{(16)}$ 1	idem
$C0DA0_{(16)}$	4	$C0DA0_{(16)}$ 5	$C0DA0_{(16)}$ 6	$C0DA0_{(16)}$ 7	$ABCA0_{(16)}$ 0	idem
$DACA0_{(16)}$	0	$DACA0_{(16)}$ 1	$DACA0_{(16)}$ 2	$DACA0_{(16)}$ 3	$DACA0_{(16)}$ 4	idem
$FABA0_{(16)}$	2	$FABA0_{(16)}$ 3	$FABA0_{(16)}$ 4	$FABA0_{(16)}$ 5	$FABA0_{(16)}$ 6	idem
$75FA0_{(16)}$	1	$75FA0_{(16)}$ 2	$75FA0_{(16)}$ 3	$75FA0_{(16)}$ 4	$75FA0_{(16)}$ 5	idem
$C4CA0_{(16)}$	6	$C4CA0_{(16)}$ 7	$747A0_{(16)}$ 0	$747A0_{(16)}$ 1	$747A0_{(16)}$ 2	idem

Le dernier accès mémoire ne concerne pas cet ensemble (FA au lieu de A0).



Exercice 5 :

Représenter en norme IEEE754 simple précision le nombre $FAB, C0D_{(16)}$.
 Pour rappel une version simplifiée de cette norme (en simple précision) :

signe 1 bit	Exposant 8 bits $E = e + 127$	mantisse M de 23 bits $m = 1, \underbrace{m_0 m_1 \dots m_{22}}_{M_{(2)}}$	valeur
s	11111111	00000000000000000000000	$(-1)^s \times \infty$
s	E	M	$(-1)^s \times m \times 2^e$
?	00000000	00000000000000000000000	0
?	11111111	M	NaN
?	00000000	M	$(-1)^s \times 0, M \times 2^e$

Exercice 6 :

Traduire en 780 le fragment de code ci-dessous :

```
int *int_t1; // un tableau de 4 entiers 16 bits stocké à l'adresse 100h
int *int_t2; // un tableau de 4 entiers 16 bits stocké à l'adresse 108h
int_i;      // un entier 8 bits

for (int_i = 0; int_i < 4; int_i++) {
int_t2[int_i] = int_t1[int_i] + int_t2[int_i];
}
```

On utilisera les opérations d'addition 8 bits. Pour rappel, une valeur sur 16 bits est stockée de la manière suivante : la partie de poids faible sur le 1er octet (adresse de la valeur) et la partie de poids fort sur le second octet (adresse de la valeur + 1).

Réponse _____

Cette solution utilise le registre IX pour parcourir le 1er tableau et IX+8 pour parcourir le 2nd.

```
LD IX,100h    ; adresse des tableaux
LD B,0       ; compteur de boucle
bcle : LD A,(IX)    ; addition des poids faibles
LD C,A
LD A,(IX+8)
ADD A,C
INC IX       ; incrémente l'adresse
LD A,(IX)   ; addition des poids forts avec retenue
LD C,A
LD A,(IX+8)
ADC A,C

INC IX      ; incrémente l'adresse
INC B      ; incrémentation du compteur et test de fin de boucle
LD A,4
CP B
JR NZ,bcle
```

On peut également utiliser 2 registres d'adresse pour identifier les 2 tableaux.

