

Ing1 - Examen de Rattrapage d'Architecture des Ordinateurs

CORRIGÉ

Durée : 2H

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

Nom Prénom :

Groupe :

Exercice 1 :

3 points

On considère une machine de Von Neumann qui possède une mémoire constituée de mots de 16 bits et adressée sur 32 bits.

1. Quelle est la taille maximale de la mémoire en mots ?

0,25 Réponse :

2. Quelle est la taille maximale de la mémoire en octets (utiliser l'unité appropriée : octet, Ko, Mo, To, Po) ?

0,25 Réponse : $2^{32} \times 2 \text{ octets} = 2^3 \times 2^{30} \text{ octets}$

3. Quelle est la plage d'adresses dont vous disposez ?

0,25 Réponse (en décimal) :

0,25 Réponse (en binaire) :

0,25 Réponse (en hexadécimal) :

4. Quelle est la taille du registre MAR ?

0,25 Réponse : (adresse)

points de suspension
acceptés si nb de
chiffres précisés

5. Quelle est la taille du registre MDR ?

0,25 Réponse : (donnée)

6. On considère les 3 instructions suivantes qui chargent une donnée de 1 mot (16 bits) dans le registre R1. On suppose que la taille des instructions est fixe sur 3 mots (48 bits). Les instructions sont en mémoire principale. Donner le nombre d'accès à la mémoire principale nécessaire(s) au traitement de chaque instruction. Justifier votre réponse.

- LOAD R1, 127 ; adressage immédiat

0,25 Réponse :

- LOAD R1, (127) ; adressage direct

0,5 Réponse :

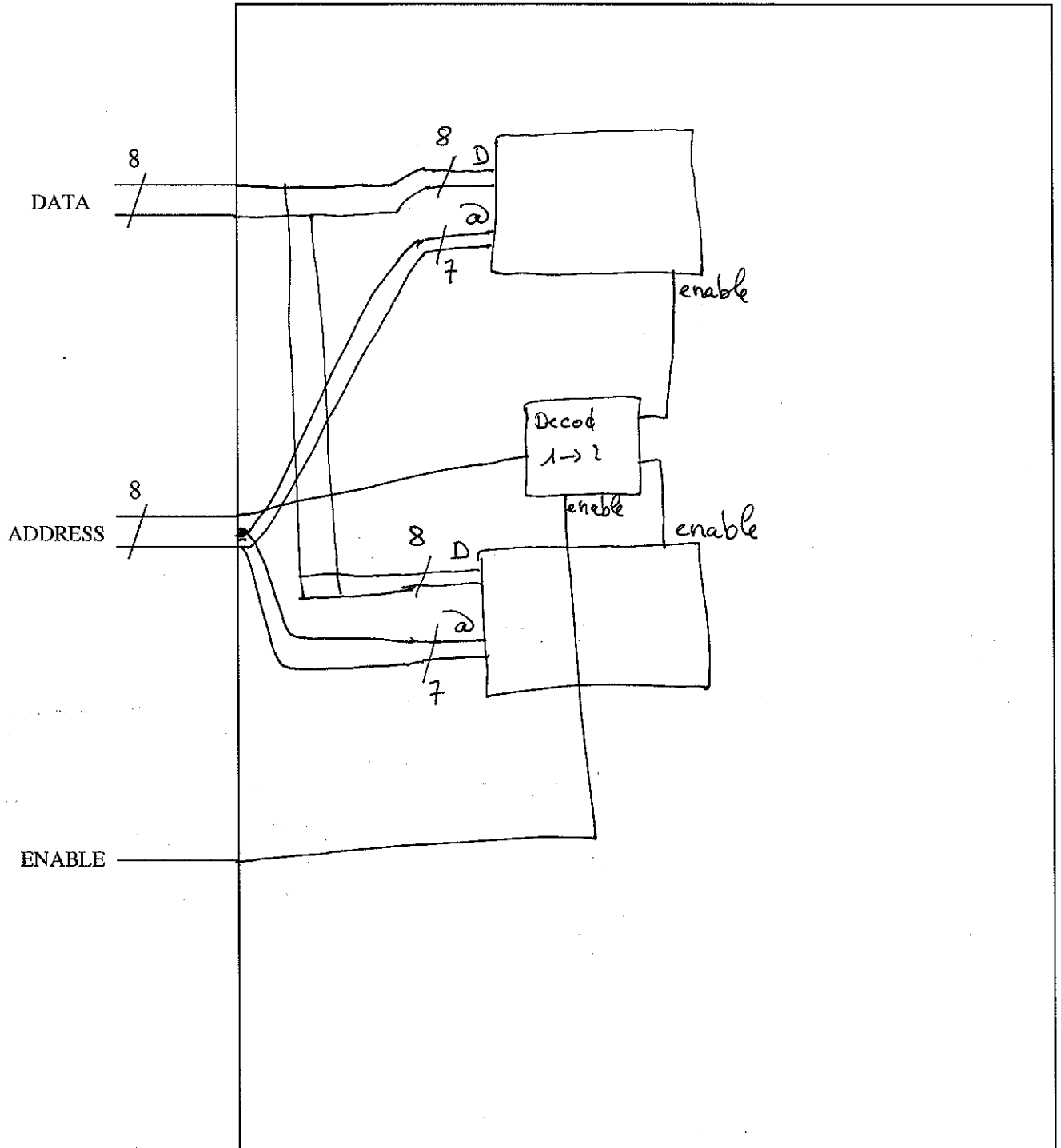
- LOAD R1, ((127)) ; adressage indirect par mémoire

0,5 Réponse :

Exercice 2 : 3 points

On souhaite construire une mémoire constituée de mots de 8 bits et adressée sur 8 bits. Pour la fabriquer, on dispose de boîtiers mémoires plus petits avec des adresses sur 7 bits et des mots de 8 bits.

Compléter le schéma suivant de montage en câblant uniquement les broches d'adressage, de données et d'autorisation.



- 0,5 2 boîtiers internes
- 0,5 enable boîtier principal sur decodeur
- 0,5 data boîtier — sur data boîtiers internes
- 1 adresse (7) sur decodeur sur enable boîtier interne
- 0,5 adresse (6..0) sur adresse boîtier interne

Exercice 3 : 4 points

1. Donner le code en complément à 2 sur 6 bits des entiers relatifs suivants :

$5_{(10)}$:

0	0	0	1	0	1
---	---	---	---	---	---

$-3_{(10)}$:

1	1	1	1	0	1
---	---	---	---	---	---

$-16_{(10)}$:

1	1	0	0	0	0
---	---	---	---	---	---

$-18_{(10)}$:

1	0	1	1	1	0
---	---	---	---	---	---

2 pt
(0,5 chacun)

2. Quel est l'intervalle des nombres codables en complément à 2 sur 6 bits ?

0,25 pt Réponse :

-	32	(2^5))	à	31	(2^5-1))
---	----	---	---------	---	---	----	---	-----------	---

3. Quelle est la condition de dépassement d'une addition de deux nombres codés en complément à 2 ?

0,25 pt Réponse :

2	dernières	retenues	(si	les	pes	font)	ont	différentes
---	-----------	----------	---	----	-----	-----	-------	-----	-------------

4. Calculer les additions suivantes en complément à 2 sur 6 bits. Vous préciserez si le résultat est correct ou s'il y a dépassement de capacité.

$5_{(10)} + (-3)_{(10)}$: $\uparrow =$

1	1	1	0	1		
0	0	0	1	0	1	
+						
1	1	1	1	1	0	1

1	0	0	0	0	1	0

$(-16)_{(10)} + (-18)_{(10)}$: \neq

1	0	0	0	0	0
1	1	0	0	0	0
+					
1	0	1	1	1	0

1	0	1	1	1	0

1,5 pt
(0,75 chacune)

Interprétation du résultat : $2_{(10)}$

Sur 6 bits, résultat : $000\ 010 \equiv 2_{(10)}$ correct vérif : 2 dernières retenues égales cqd

Interprétation du résultat :

2 dernières retenues $\neq \Rightarrow$ dépassement de capacité. Normal -34 non codable sur 6 bits en C2

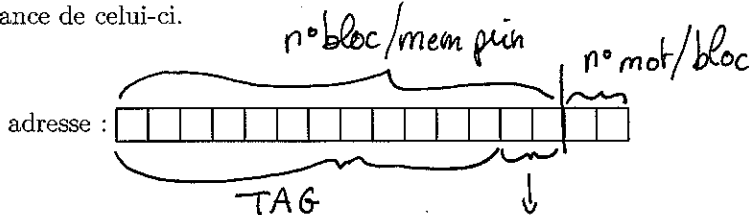
Exercice 4 : 3,5 pts

On adjoint à une mémoire principale (adressée sur 16 bits) une mémoire cache associative par ensemble. Les deux mémoires sont constituées de blocs de 4 mots. La mémoire cache possède 4 ensembles de 8 blocs.

1. À partir de l'adresse (sur 16 bits) d'une donnée en mémoire principale, expliquer à l'aide du schéma suivant, comment trouver :

- le numéro du mot dans le bloc : $4/\text{blocs} \Rightarrow 2$ bits
- le numéro du bloc dans la mémoire principale
- le numéro de l'ensemble qui va accueillir le bloc de cette donnée 4 ensembles $\Rightarrow 2$ bits
- l'information additionnelle (TAG) que vous devez ajouter à un bloc de donnée pour retrouver la provenance de celui-ci.

1 pt
(0,25 par cas)



n° ensemble (ou n° impré quel sous ensemble fixe de 2 bits du n° de bloc)

2. 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 donnée par le tableau ci-dessous, montrez l'évolution de l'ensemble représenté en fonction des accès (lecture ou écriture) aux données des blocs de numéros suivants : 260, 504, 224, 260 et 496.

Mettez en évidence les défauts de cache. Dans le tableau suivant, on représente les 8 blocs contenus dans l'ensemble étudié par leur numéro de bloc et leur TAG de LRU sur 8 lignes avec leur évolution à chaque accès mémoire.

num bloc	LRU	num bloc	LRU	num bloc	LRU	num bloc	LRU	num bloc	LRU	num bloc	LRU
12	0	-	1	-	2	-	3	-	-	-	4
504	3	-	4	504	0	-	1	-	2	-	3
496	5	-	6	-	-	-	7	-	-	496	0
48	4	-	5	-	-	-	6	-	-	-	7
128	7	* 260	0	-	1	-	2	260	0	-	1
44	2	-	3	-	4	-	5	-	-	-	6
688	1	-	2	-	3	-	4	-	-	-	5
224	6	-	7	-	-	224	0	-	1	-	2

2,5 pt
(0,5 par accès)

Exercice 5 : * : défaut de cache (1 seul) - : pas de changement

Représenter en norme IEEE754 simple précision le nombre réel suivant exprimé en hexadécimal 80,7FAB0D₍₁₆₎.

Pour rappel une version simplifiée de cette norme (en simple précision) :

3,5 pt

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	$(-1)^s \times 0, M \times 2^e$

Réponse : 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 1

Explications

80,7FAB0D₍₁₆₎ = 1000 0000, 0111 1111 1010 1011 0000 1101₍₂₎
 = $(-1)^0 \times 1,000\ 0000\ 0111\ 1111\ 1010\ 1011\ 0000\ 1101 \times 2^7$ → 1 pt

0,5 pt S = 0
 1 pt E = 7 + 127 = 134 → 1000 0110
 1 pt M = 000 0000 0111 1111 1010 1011 / 0000 1101 (on garde 23 chiffres)

Exercice 6 : 3 pts

Traduire en Z80 le fragment de code ci-dessous :

```
long long_entier1; // un entier 64 bits stocké à l'adresse 100h
long long_entier2; // un entier 64 bits stocké à l'adresse 108h

long_entier1 = long_entier1 + long_entier2;
}
```

On utilisera les opérations d'addition 8 bits. Pour rappel, une valeur sur 64 bits est stockée sur 8 mots de 8 bits de la manière suivante : les 8 bits de poids faible sur le 1er octet (adresse de la valeur), le 2ème quart sur le 2ème octet, le 3ème quart sur le 3ème octet et les 8 bits de poids forts sur le 8ème octet.

jusqu'aux

Réponse :

LD IX, 100h

LD A, (IX)

ADD A, (IX+8)

LD (IX), A

} addition des octets
de poids faibles

LD B, 7

bclé : INCR IX

; compteur de bclé pour les 7 octets suivants
(accepter 3 à cause de l'erreur d'énoncé)
→ on passe à l'octet suivant

LD A, (IX)

ADC A, (IX+8)

LD (IX), A

; addition d'1 octet
de chaque nombre avec la
retenue de l'addit° précédente

~~DNZ~~ bclé

; B = B - 1 et rebouclage
jusqu'à B = 0

barème :

1pt : addition poids faibles

1pt : addition avec retenue autres poids

1pt : boucle