

Solution TD2: Architecture des ordinateurs 2020/2021

Question 1

- Par taille : Registres < L1 < L2 < RAM < CD < DD.
- Par vitesse : Registres > L1 > L2 > RAM > DD > CD.

Question 2 Supposons qu'un livre contienne **500 pages** de **80 lignes** de **100 caractères** chacune (ponctuations et blancs compris). Combien de caractères composent ce livre ? Sachant qu'un octet représente un caractère, combien de livres faut-il pour avoir l'équivalent de **30 Go** ? Enfin l'épaisseur de ce livre est de **2cm**, quelle hauteur de livres obtient-on avec **30 Go** ?

- **Nombre de caractères composant le livre = $500 \times 80 \times 100 = 4.000 .000$ caractères = 4 Mo**
- **Nombre de livres pour 30 Go = $30 \text{ Go} / 4 \text{ Mo} = (30 \times 1000) / 4 = 7500$ Livres**
- **Hauteur de livres avec 30 Go = $7500 \times 2 \text{ cm} = 15000 \text{ cm} = 150 \text{ mètre}$**

Question 3

- $(00101001)_2 = (41)_{10} + (11001010)_2 = (202)_{10} = (11110011)_2 = (243)_{10}$
- $(10101011)_2 (171)_{10} + (11001010)_2(202)_{10} = (101110101)_2(303)_{10}$ le résultat produit un overflow.
- $(11111111)_2(255)_{10} + 111111112(255)_{10} = (111111110)_2(510)_{10}$ le résultat produit un overflow.

Question 4

- Supposons sans perte de généralité que $n \geq m$
- Prenons $X_{\max} = 2^n - 1$ et $Y_{\max} = 2^m - 1$
- Or $x + y \leq X_{\max} + Y_{\max} = 2^n + 2^m - 2 = 2^n (1 + 2^{m-n} - 2^{1-n}) < 2^{n+1}$, il faut donc
- $n + 1 = \max(n;m) + 1$ bits.
- $X \cdot Y = (2^n - 1) \cdot (2^m - 1) = 2^{n+m} - 2^n - 2^m + 1 < 2^{n+m} + 1$
- $(n + m)$ bits pour la multiplication

Question 5 On dispose d'une mémoire RAM dont le boîtier comporte un bus d'adresse de 16 entrées et un bus de données de 8 entrées.

1. Quelle est la taille des registres RM et RA ?
2. Quel est le nombre de cases adressables dans cette mémoire ?
3. Quel est le nombre de cases constituant un mot mémoire si la taille d'une case est de 4 bits?
4. Quelle est la taille de cette mémoire en octets pour chacun des cas suivants :
 - a. La taille d'une case mémoire est de 1 bit,
 - b. La taille d'une case mémoire est de 2 bits,
 - c. La taille d'une case mémoire est de 4 bits,
5. Pour des cases mémoire de 2 bits.
 - a. Quelle est la taille maximum de cette mémoire en octets ?

b. Quelle est l'adresse en décimal et en hexadécimal du **15^{ème} mot mémoire** ?
(la numérotation commence de zéro)

c. Calculer l'adresse en décimal puis en hexadécimal du **9^{ème} élément** d'un tableau dont l'adresse du premier élément est **(34)₁₀ = (0022)₁₆**, et dont tous les éléments sont composés de **6 bits** ;

d. Calculer le nombre d'éléments de ce tableau sachant que l'adresse de son dernier élément est **(91)₁₀ = (005B)₁₆** ;

e. Quelle est la taille de ce tableau en octets ?

f. combien de tableaux on peut charger dans cette mémoire si la capacité de chaque tableau est de **10 élément**, chaque élément est de **4 bits**, la case

✓ Solution Question 5

➤ **1. La taille du Registre Mémoire (RM) est 8 bits et le RA 16 bits**

➤ **2. Nombre de case adressable est 2^{16}**

➤ **3. Pour une case mémoire de 4 bits le mot mémoire sera de 8 bits**

➤ **4. 1^{er} Cas** une case mémoire de 1 bit la taille en octet de cette mémoire est de :
 $2^{16} / 8 \text{ octet} = 8192 = 8 \text{ Ko}$

2^{ème} Cas une case mémoire de 2 bit la taille en octet de cette mémoire est de :
 $2^{16} \times 2 / 8 \text{ octet} = 16384 = 16 \text{ Ko}$

3^{ème} Cas une case mémoire de 4 bit la taille en octet de cette mémoire est de :
 $2^{16} \times 4 / 8 \text{ octet} = 32768 = 32 \text{ Ko}$

➤ **5. a. adresse du 15^{ème} mot mémoire $(15-1) \times 2 = (28)_{10} = (001C)_{16}$**

Numéro case	Adr (Décimale)	Adr (Héxa)	Case mémoire (2bits)	
1	0	0		
2	1	1		
3	2	2		
4	3	3		
5	4	4		
6	5	5		
7	6	6		
8	7	7		
9	8	8		
10	9	9		
11	10	A		
12	11	B		
13	12	C		
14	13	D		
15	14	E		
	15	F		
	16			
	17			
	18			
	19			
	20			
	21			
	22			
	23			
	24			

25			
26			
27			
28			
29			

b . Calculer l'adresse en décimal puis en hexadécimal du 9^{ème} élément du tableau est

Décimale : on a chaque élément a 6 bits et une case mémoire a 2 bits donc Chaque élément du tableau a

6/2 = 3 cases mémoire

Pour l'adresse du 9^{ème} éléments : $34 + ((9-1) \times 3) = (58)_{10} = (003A)_{16}$

c. nombre d'éléments du tableau : adr du dernier élément 91

(91 - 34) / 3 = 19 éléments

d. taille du tableau en octet : $(19 \times 6) = (114)_{\text{bit}} = (114/8)_{\text{octet}} = 14,25 \approx 15$ Octet

Numéro case	Adr (Décimale)	Adr(Héra)	Case mém(2bits)	
1	0	0		
2	1	1		
3	2	2		
4	3	3		
5	4	4		
6	5	5		
7	6	6		
8	7	7		
9	8	8		
10	9	9		
11	10	A		
12	11	B		
13	12	C		
14	13	D		
15	14	E		
16	15	F		
17	16			
18	17			
19	18			
20	19			
21	20			
22	21			
23	22			
24	23			
25	24			
26	25			
27	26			
28	27			
29	28			
30	29			
31	30			
32	31			

33	32				
34	33				
35	34	0022			Adr 1 ^{ère} éle du tab
36	35				
37	36				
38	37	0025			Adr 2 ^{ème} élem
39	38				
40	39				
41	40	0028			Adr 3 ^{ème} élem
42	41				
43	42				
44	43	002B			Adr 4 ^{ème} élem
45	44				
46	45				
47	46	002E			Adr 5 ^{ème} élem
48	47				
49	48				
50	49	0031			Adr 6 ^{ème} élem
51	50				
52	51				
53	52	0034			Adr 7 ^{ème} élem
54	53				
55	54				
56	55	0037			Adr 8 ^{ème} élem
57	56				
58	57				
59	58	003A			Adr 9 ^{ème} élem
60	59				
61	60				
62	61	003D			Adr 10 ^{ème} élem
63	62				
64	63				
65	64	0040			Adr 11 ^{ème} élem
66	65				
67	66				
68	67	0043			Adr 12 ^{ème} élem
69	68				
70	69				
71	70	0046			Adr 13 ^{ème} élem
72	71				
73	72				
74	73	0049			Adr 14 ^{ème} élem
75	74				
76	75				
77	76	004C			Adr 15 ^{ème} élem
78	77				
79	78				
80	79	004F			Adr 16 ^{ème} élem
81	80				
82	81				
83	82	0052			Adr 17 ^{ème} élem
84	83				

85	84			
86	85	0055		Adr 18 ^{ème} elem
87	86			
88	87			
89	88	0058		Adr 19 ^{ème} elem
90	89			
91	90			
92	91	005B		Adr 20 ^{ème} elem
93	92			
94	93			

Question 6 Sachant que la taille du bus d'adresse d'un processeur est de 20 bit, combien de segments peut-il gérer ? Quelle est la taille de ces segments ? Justifier votre réponse.

Le registre de segment est sur 16 bits donc il y a 2^{16} segments, donc 65536 segments.

A chaque segments correspond une adresse d'offset sur 16 bits donc il y a 65536 adresses par segment

Question 7 Le programme suivant réalise une temporisation. Pour ce faire, il décrémente un registre 16 bits, *i.e.* de 0100H (fixé au départ) à 0. Pour chacune des lignes du programme, on donne la durée d'exécution d'une instruction complète en microcycle (μ c). Le processeur travaille à une fréquence de 10 MHz, c'est-à-dire que chacune des opérations élémentaires est effectuée en 1 μ c de 100 ns (rappel 1 ns = 10^9 s). **Modifier la valeur initiale de tempo pour que l'ensemble de la temporisation atteigne une durée 1ms**

```

                ASSUME    CS : Code, DS : Data
Data           SEGMENT
Tempo          DW        0100H
Data           ENDS

Code           SEGMENT
Debut : MOV     AX , Data      10
          MOV     DS , AX       2

          MOV     AX , Tempo    10
Boucle : ADD     AX , -1        4
          JNE     Boucle       16

          MOV     AH , 4CH      4
          INT     21H          52

Code           ENDS
                END         Debut

```

Solution :

$$\begin{aligned}\text{Durée} &= 10 + 2 + 10 + 256(4 + 16) + 4 + 52 \\ &= 5198 \mu\text{c} \\ &= 5198 \times 100 \text{ ns} \\ &= 519\,800 \text{ ns}\end{aligned}$$

Modification de la valeur initiale de tempo pour que l'ensemble de la temporisation atteigne une durée 1ms

$$1\,000\,000 \text{ ns} = [10 + 2 + 10 + \text{Tempo} \times (4 + 16) + 4 + 52] \times 100 \text{ ns}$$

$$10\,000 = 10 + 2 + 10 + \text{Tempo} \times (4 + 16) + 4 + 52$$

$$\text{Tempo} \times (4 + 16) = 10\,000 - 10 - 2 - 10 - 4 - 52$$

$$\text{Tempo} = 9922/20$$

$$\text{Tempo} = (496)_{\text{D\'ecimal}} = 1\text{F0 H}$$

Question 8 Donner la d\'efinition du registre d\'\'etat et citer 4 indicateurs d\'\'etat en pr\'ecisant leur fonction

Le registre d\'\'etat est un registre qui regroupe les indicateurs d\'\'etat du processeur. Citons 4 indicateurs d\'\'etat

- **ZF : indique si le r\'esultat de la derni\'ere op\'eration est nul**
- **CF : indique une \'eventuelle retenue**
- **SF : indique le signe du r\'esultat d\'une op\'eration**
- **OF: indique un \'eventuelle d\'epassement**