

Architecture des ordinateurs

TD 1:

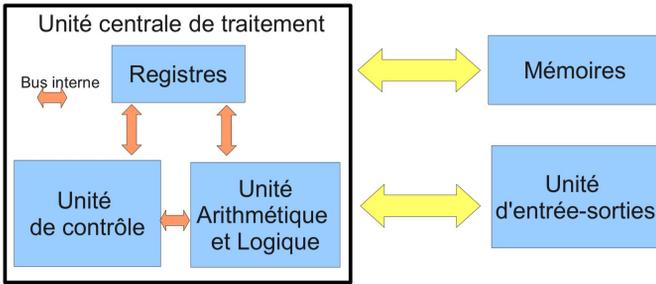
Exercices 4, 5 et 6:

Archi Ord

Exercice 4 : questions de cours

1 - Unité de Contrôle, unité arithmétique et logique, registres
unité de gestion de la mémoire, Bus, cache, horloge

2 -



Elle diffère de celle de Harvard car celle de Von Neuman stocke dans une même mémoire les données et instructions.
Contrairement à celle de Harvard où il y a séparation physique des mémoires contenant les données et les programmes.

3 - Bus : Ensemble de supports de l'information (fils par ex).

types de bus :

- Bus de données
- Bus de Commande
- Bus d'adresse.

4 - registre > mémoire cache > mémoire centrale > mémoire de masse

Composants relatifs

5 - capacité de stockage ?, états binaires 0 ou 1, hexa: 0-9, A-F.
adresse d'une φ mémoire : id de la φ .

Exercice 5 : définitions

- 1) E 2) F. 3) H 4) a 5) d 6) c
7) i 8) J 9) g.

Exercice 6 : stockage de l'information en mémoire

1)

2009h	30
200Ah	A7
200Bh	98
200Ch	47

← la valeur stockée
en 200Bh
en Big Endian

2009h	47
200Ah	98
200Bh	A7
200Ch	30

← en mode little Endian

en Big Endian

200D	CA
200E	FE

en little Endian

200D	FE
200E	CA

- 2) 0x99EE0 : 96 ← début
0x99F2F : 00 ← Fin

la taille en octets = $16 \times 5 : \frac{160}{2} = 80.$

b)

	B.E	L.E
3A	9F0C	31
ED	9F0D	C2
F5	9F0E	01
44	9F0F	8F
8F	9F10	44
01	9F11	FF
C2	9F12	ED
31	9F13	3A

3AED }
 FF44 }
 8F01 }
 C231 }
 Lectures
 au decimal

31C2 }
 018F }
 44FF }
 ED3A }
 Lectures
 a decimal.

c)

c. Si on considère que l'extrait de mémoire ci-dessus peut être adressé en utilisant le mode big-endian ou little-endian et peut être adressée en mode 8, 16 ou 32 bits, donnez les informations manquantes :

- adresse 0x9F14 en mode **big-endian**, au format 8 bits par mot => 0x9B
- adresse 0x9F17 en mode **little-endian**, au format 16 bits par mot => 0xC153
- adresse 0x9F1F en mode **little-endian**, au format 16 bits par mot => **CPFE**
- adresse 0x9F2E en mode **L.E**, au format 16 bits par mot => 0x0047
- adresse 0x9F0E en mode **big-endian**, au format 16 bits par mot => **FF44**
- adresse 0x9F0C en mode **L.E**, au format 32 bits par mot => 0x44F5ED3A
- adresse 0x9F0E en mode **little-endian**, au format 32 bits par mot => **FAA32F94**
- adresse 0x9EF6 en mode **B.E**, au format 16 bits par mot => 0x764C
- adresse 0x9EED en mode **B.E**, au format 32 bits par mot => 0x0A4538E8

TD n°1

Consigne : vous devrez réaliser les exercices marqués du symbole ■ chez vous avant la séance de TD. Les bonnes réponses de ces exercices seront données durant la séance, sans explication détaillée. Seuls les exercices marqués du symbole ☆ seront corrigés de manière détaillée.

Exercice 1 : rappels sur le codage de l'information ■

1) Convertir en décimal les nombres suivants exprimés en binaire naturel :

128 = (a) 10000000_2 (b) 01011010_2 (c) 11111000_2 (d) 0001010000010110_2 *4226*

2) Convertir en binaire naturel les nombres décimaux suivants :

(a) 23_d *10111* (b) 372_d *101110100* (c) 2875_d *101100101011*

3) Convertir en hexadécimal les nombres suivants exprimés en binaire naturel :

(a) 1011_2 *B* (b) 1100_2 *C* (c) 01101010_2 *6A* (d) 1010101110010011_2 *AB90*

4) Écrire les nombres hexadécimaux suivants sous forme binaire et préciser sur combien d'octets ils sont représentés :

(a) $F6_h$ *11110110* (b) $1A3C_h$ *11011000111100* (c) $9BD5_h$ (d) $4E67C1FF_h$ *32 octets*

5) Écrire le nombre décimal 118_d en code :

(a) binaire naturel *011110110* (b) hexadécimal *76* (c) ASCII *V*

Note : le code ASCII du caractère 'a' vaut 97_d

Exercice 2 : rappels sur l'arithmétique binaire ■

On considère une machine numérique travaillant sur des mots binaires de 8 bits et comportant une UAL (Unité Arithmétique et Logique) de 8 bits.

1) Donnez la représentation binaire des nombres décimaux signés suivants selon cette machine :

(a) -13_d *11110011* (b) $+44_d$ *00101100* (c) -82_d *10101110*

2) Donnez le résultat en binaire, et son interprétation signée, de l'addition des nombres décimaux signés suivants et donnez la valeur des bits de retenue (bit $C = C_n \oplus C_0$, avec respectivement C_n et C_0 la retenue sortante du dernier étage de l'UAL et la retenue entrante du premier étage) et de débordement (bit $V = C_n \oplus C_{n-1}$, avec C_n et C_{n-1} la retenue sortante des deux derniers étages de l'UAL) du registre d'état de l'UAL :

(a) $+7_d + (-5_d)$ (b) $+3_d + (-8_d)$ (c) $-4_d + (-5_d)$

Exercice 3 : premier programme assembleur 8086 avec TASM ☆

*on converti en binaire
Dnc + 2d
et on trouve C et V*

- Écrivez un programme assembleur 8086 en respectant la syntaxe supportée par Turbo Assembler (TASM) permettant d'additionner les deux valeurs constantes (exprimées ici en hexadécimal) 27_h et 21_h et de stocker le résultat dans le registre accumulateur AL du microprocesseur. Le programme se terminera par un appel à l'instruction `int 21h` qui permettra de mettre fin "proprement" au programme.
Quelle instruction utilise-t-on pour l'addition ? Quel est le résultat de l'addition ? Quelles sont les valeurs des bits C (retenue) et V (débordement) du registre d'état de l'UAL ?
On remplace maintenant dans le programme précédent la valeur constante 27_h par $e9_h$. Quel résultat obtient-on ? Quelles sont les valeurs des bits C et V ? Que s'est-il passé ?
- Écrivez maintenant un programme permettant de faire la soustraction, dans le registre AL, de la valeur 27_h par la valeur constante 21_h . Quelle instruction utilise-t-on pour la soustraction ? Quel est le résultat de cette soustraction ? Quelles sont les valeurs des bits C (retenue) et V (débordement) du registre d'état de l'UAL ?
On remplace maintenant dans le programme précédent la valeur constante 27_h par $e9_h$. Quel résultat obtient-on ? Quelles sont les valeurs des bits C et V ? Expliquez pourquoi on obtient ces valeurs.

Le bus interne.
 Registre (stockage temporaire)
 L'unité de contrôle

Exercice 4 : questions de cours

Bus :
 De Données
 De Commande
 D'adresses

- Quels sont les éléments constituant l'unité centrale de traitement d'un ordinateur ? **UAL (opérations)**
- Donnez le schéma général d'une architecture de Von Neumann. En quoi cette architecture diffère-t-elle de celle de Harvard ? **L'archi de Von Stoche dans une m. mémoire le donne et instruction / Harvard**
- Dans l'architecture d'un ordinateur, qu'est-ce qu'un bus, et à quoi cela sert-il ? Quels sont les différents types de bus dans l'architecture d'un ordinateur ? **E de suppt de l'inf (file par bus)**
- Soit les catégories de mémoire suivantes : Ordonnez les types de mémoire suivants en fonction de leur rapidité d'accès par l'unité centrale de traitement, du plus rapide au moins rapide : **mémoire centrale, registres, mémoire de masse, mémoire cache**. Parmi ces mémoires, lesquelles sont volatiles ? **3 1 4**
- Quelle est la capacité de stockage d'une cellule mémoire ? Combien d'états différents peut prendre une cellule mémoire ? Quels sont ces différents états en binaire, en hexadécimal ? Qu'est-ce que l'adresse d'une cellule mémoire ? **0 ou 1 ou 1**

Exercice 5 : définitions

Associez chaque mot de la liste suivante avec une des définitions proposées ci-après :

- a) Horloge b) Unité Centrale de Traitement c) Circuit intégré d) RAM e) Registre
 f) Transistor g) Système d'exploitation h) Carte Mère. i) Bus j) Microprocesseur

- Petite mémoire contenue dans le microprocesseur destiné à stocker de manière très temporaire un certain nombre d'informations comme les résultats intermédiaires d'un calcul ou l'adresse de la prochaine instruction à exécuter. **registre**
- Sorte de commutateur électronique servant à contrôler le passage du courant électrique. **transistor**
- Principale carte de circuits imprimés dans un ordinateur. **carte mere**
- Cristal de quartz vibrant à une fréquence déterminée, produisant des signaux périodiques qui servent à synchroniser les tâches d'un microprocesseur. **horloge**
- Type de mémoire volatile, lisible et réinscriptible, dont chaque cellule est directement accessible. **RAM**
- Ensemble de circuits gravés sur une plaque de silicium. **circuit integes**
- Ensemble des lignes transportant les signaux qui permettent au microprocesseur de communiquer avec ses mémoires et ses périphériques. **BUS**
- Unité principale de traitement d'un ordinateur, généralement contenue dans un circuit intégré unique. **microprocesseur**
- Programme nécessaire à la gestion des ressources matérielles et logiciels d'un ordinateur. **Système d'exploitation**

Exercice 6 : stockage de l'information en mémoire

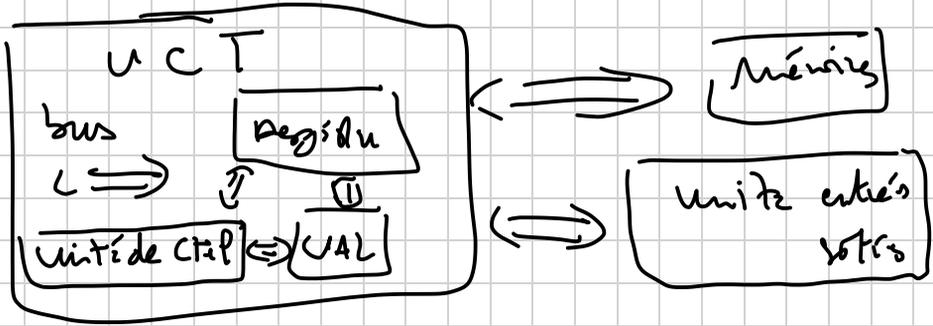
- La valeur hexadécimale 0x30A79847 est stockée dans la mémoire centrale d'un ordinateur à l'adresse 2009h, et la valeur 0xCAFE à l'adresse 200Dh. Quelle est la valeur stockée dans la mémoire à l'adresse 200Bh si le mode de stockage est le système big-endian ? Si le mode de stockage est le système little-endian ?
- Soit le contenu suivant, d'un extrait de la mémoire centrale d'un ordinateur :

Addr	:	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x9EE0	:	96	7B	5C	43	87	66	FB	57	44	05	D3	53	02	E8	38	45
0x9EF0	:	0A	97	0C	7D	AC	9A	76	4C	B7	88	8D	F6	D3	24	59	6A
0x9F00	:	9F	B6	AD	27	1D	A8	7F	61	AD	52	B5	B0	3A	ED	F5	44
0x9F10	:	85	01	C2	31	9B	39	7D	53	C1	0A	49	94	2F	A3	FE	CF
0x9F20	:	5A	AC	F6	77	55	75	D8	02	C8	8D	B2	02	7B	A8	47	00

- A quelle adresse débute cet extrait de la mémoire ? A quelle adresse se termine-t-il ? Quelle est la taille en octets de cet extrait de mémoire ?
- Représentez l'extrait de de cette mémoire allant des adresses 0x9F0C à 0x9F13 sous la forme d'un tableau à une dimension. En supposant que les données contenues dans cet extrait constituent un tableau de nombres entiers non signés codés sur 16 bits, donnez le contenu en décimal de ce tableau selon que les données sont rangées selon la convention big-endian ou little-endian.
- Si on considère que l'extrait de mémoire ci-dessus peut être adressé en utilisant le mode big-endian ou little-endian et peut être adressée en mode 8, 16 ou 32 bits, donnez les informations manquantes :
 - adresse 0x9F14 en mode **big-endian**, au format _____ bits par mot => **0x9B**
 - adresse 0x9F17 en mode **little-endian**, au format _____ bits par mot => **0xC153**
 - adresse 0x9F1F en mode **little-endian**, au format **16 bits** par mot => _____
 - adresse 0x9F2E en mode _____, au format **16 bits** par mot => **0x0047**
 - adresse 0x9F0E en mode **big-endian**, au format **16 bits** par mot => _____
 - adresse 0x9F0C en mode _____, au format **32 bits** par mot => **0x44F5ED3A**
 - adresse 0x9F0E en mode **little-endian**, au format **32 bits** par mot => _____
 - adresse 0x9EF6 en mode _____, au format _____ bits par mot => **0x764C**
 - adresse 0x9EED en mode _____, au format _____ bits par mot => **0x0A4538E8**

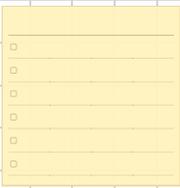
TD 01

Structure système en mémoire:



1) $0x30A79847$ ou se termine par h.

↓
Valeur
en hexadécimal



↓
Codé sur 32 bits
4 octets

4 case mémoires du comp
consécutives
Après 2 méthodes de
stockage

↓
Little-Endian Big-Endian
47 puis 98 puis A7 puis 30 puis 30 puis A7 puis 98 puis 47

47	0000h
98	0001h
A7	0002h
30	0003h

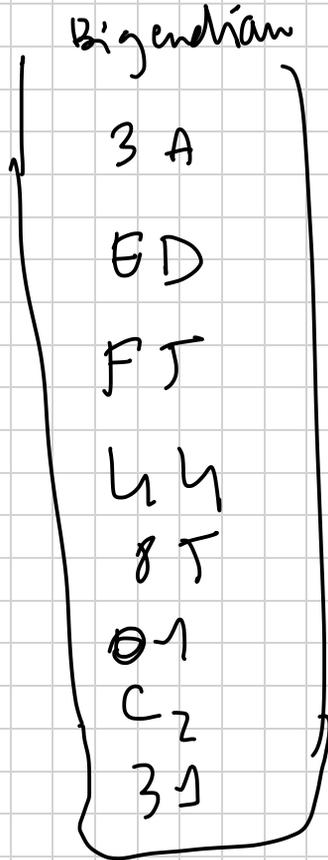
1) Big-Endian | little-Endian

30	47	2009h
A7	98	200Ah
98	A7	200Bh
47	30	200Ch

	<u>Big-Endian</u>	<u>little-Endian</u>	
CA		FE	200D
F		CA	200E

2) a) 3 E E 0 0
 ä
 9 F 2 0 F

b)



8 1 0 0 0
 9 1 0 0 1
 A 1 0 1 0
 B 1 0 1 1
 C 1 1 0 0
 D 1 1 0 1
 E 1 1 1 0
 F 1 1 1 1

Orig.küh

Lil.küh

3 A E D : 1 5 0 8 T

E D 3 A : 6 0 7 3 0

F T 4 4 : 6 2 7 8 8

4 4 F T : 1 7 6 7 3

8 5 0 1 : 3 4 0 4 1

0 1 8 T : 3 8 9

2 3 1 : 4 9 7 1 3

3 1 C 2 : 1 2 7 3 8

- c)
- 8 bits
 - 16 bits
 - CEFÉ h
 - little-Endian
 - FT 44 h
 - little-Endian
 - FSED 3A B0 h
 - 89 24 D3FC
 - Big endian 16 bits 76 4C h
 - Big-Endian 32 bits.



TD 2: Orga et gestion de la mémoire par le 8086.

- 1) Les mémoires: RAM: mémoire modifiable
ROM: mémoire non modifiable

1) $2^8 = 256$.

bornes des valeurs codées: 00000000 à 1111 1111

/ / / / / / / / : 00h à FFh.

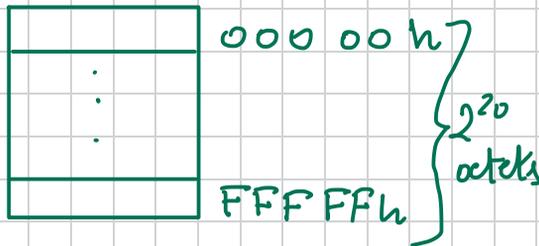
/ / / / / / / / : 0 à 255.

2) a) $2^4 = 16$ octets b) $2^8 = 256$ octets c) $2^{16} = 65536$ octets

d) $2^{20} = 1 \text{ millions} = 64 \text{ Kioctets} = 1 \text{ Mébioctet}$

e) $2^{22} = 4 \text{ gioctets}$

3) 00000h à FFFFFh.



2) des paragraphes:

1) 00000h à 0000Fh : le premier

{ 00000h
00001h
⋮
0000Eh
0000Fh

00010h à 0001Fh :

le 2ème

2) 7200h à 72c5Fh

3) FFFF0h à FFFFFh.

4) $16^4 = 2^{16} = 65536$) blocs de 16 octets

3) la segmentation de la mémoire:

1) ——— Paragraphe α Paragraphe β .

2) 00010h... 0001Fh, 00020h, 00021h
Dernière Schéma

00022h à 00023h

3) 00010h à 00023h

$$\text{adresse} = \underbrace{\text{segment} \times 16}_{\substack{16 \text{ bits} \\ \text{de poids} \\ \text{fort}}} + \underbrace{\text{offset}}_{\substack{16 \text{ bits} \\ \text{de poids} \\ \text{faible}}}$$

$2^{12} = 4096$) qui donne une très adresse physique

$$\text{Adresse} = 16 \text{ bit Segment} + \text{offset}$$

F476h (1111 0100 0101, 0110)

segment 0079h [0000 0000 0101 1001]
offset
F476h: 0079h.

F4760h
+ 0079h

F4769h

2B3E4F

2B3E4h - segment * 16 + offset

55550h
+ 00004h

55554h

4	4	7	2	0	h
0	4	7	0	A	h
<hr/>					
4	E	2	A		

Exercice 2:

1. est correct, mode d'adressage immédiat
2. incorrect
3. est correct, mode d'adressage immédiat
4. est correct, mode d'adressage immédiat
5. est correct, mode d'adressage par registre
6. est correct, mode d'adressage indirect
7. est correct, mode d'adressage indirect
8. est correct, mode d'adressage indirect.

Exercice 1:

Assume CS: code, DS: Data

Data segment

A \leftarrow 0F3h

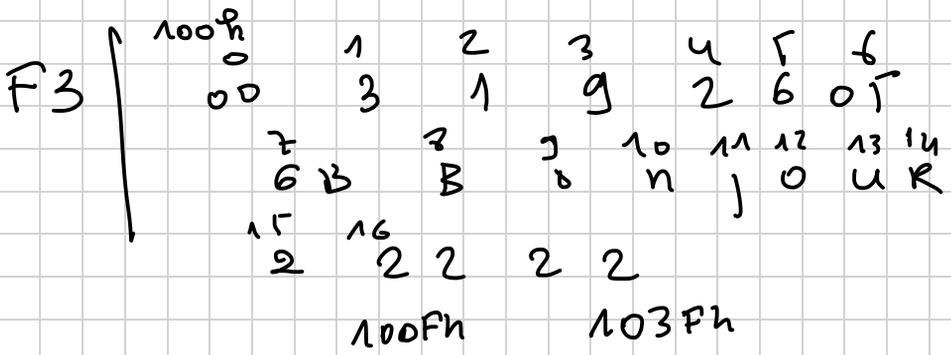
VEC1 db 6, 2, 9, 1, 3

B du 6B0Fh

Texte DB "Bonjour"

VEC2 DB 1 dup(2)

DATA ENDS



0104FH



MOV AX, CS: [BX].

CS: 4917h

BX: 0054h

49170
 0054

 491C4

mov cx,
bucle:

loop bucle \equiv Dec cx
CMP cx, 0
JNE bucle.

TD n°2

Organisation et gestion de la mémoire par le 8086

1) Les mémoires

Les mémoires, qu'elles soient modifiables (RAM) ou non modifiables (ROM), sont divisées en cellules contenant une information codée sur 8 bits (1 octet). Le microprocesseur sélectionne une cellule donnée grâce au bus d'adresses, à partir de l'adresse-mémoire de la cellule, appelée aussi simplement adresse. Il accède au contenu de la cellule grâce à un bus de données.

1°) Combien d'états différents peut prendre une cellule mémoire ? Quels sont ces différents états en binaire, en hexadécimal ?

2^8
16²

2°) Quelle est la taille mémoire adressable par un microprocesseur avec un bus d'adresses de

- a) 4 bits ? b) 8 bits ? c) 16 bits ? d) 20 bits ? e) 32 bits ?

2^n

65536 1 Mo 4 Go

3°) Le 8086 possède un bus d'adresses de 20 bits et un bus de données de 16 bits. Quelles sont les différentes adresses possibles en hexadécimal ?

00000h à FFFFFFFh

2) Les paragraphes

Les adresses mémoire dont le dernier chiffre hexadécimal est 0h divisent la mémoire en paragraphes. Un paragraphe est donc un ensemble de 16 adresses consécutives dont la première est un multiple de 16. On le repère par les 4 chiffres hexadécimaux de poids fort.

1 para

1°) Quel paragraphe est constitué par les adresses de 00000h à 0000Fh ? Quel paragraphe est constitué par les adresses de 00010h à 0001Fh ?

deuxième

2°) Quelles adresses constituent le paragraphe n° 72C9h ?

7290h à 729Fh

3°) Sur quelle plage d'adresses s'étend le dernier paragraphe ?

FFFFFFh à FFFFFFFh

4°) Combien de paragraphes le 8086 comporte-t-il ?

65536 paragraphes

3) La segmentation de la mémoire

Les registres du 8086 sont des registres 16 bits. Pour gérer les adresses mémoire de son bus d'adresses de 20 bits, les ingénieurs de chez Intel ont mis au point un procédé d'adressage particulier : le mode d'adressage réel, utilisant des segments de mémoire.

Un segment est une zone mémoire dont le début coïncide avec un début de paragraphe et dont la taille est comprise entre 16 et 65536 octets (64 Ko). L'adresse de départ d'un segment est donc un multiple de 16. Le 8086 dispose de 4 segments appelés: segment de code, segment de données, segment de pile et segment auxiliaire.

1°) Quelle doit être la taille des registres dans lesquels on souhaiterait stocker :

- a) l'adresse de début du segment ?
- b) la longueur d'un segment ?

2°) Représenter sur un schéma un segment de mémoire de longueur 20 octets débutant au paragraphe 0001h.

3°) Quelles sont les différentes adresses mémoire composant ce segment ?

4) Les registres de segment

Le registre servant à stocker le début du segment est appelé registre de segment. Pour calculer une adresse physique (20 bits), le 8086 combine deux informations (16 bits): la valeur d'un registre de segment, ainsi que la valeur comprise dans un registre d'adresse ou registre pointeur, qui indique le déplacement (ou offset) à l'intérieur du segment.

L1IEEEA – Architecture des ordinateurs – aucun document autorisé

Codez votre numéro (cf. liste) :

0 1 2 3 4 5 6 7 8 9

0 1 2 3 4 5 6 7 8 9

0 1 2 3 4 5 6 7 8 9

Nom et prénom :

.....

Répondez au questionnaire ci-dessous en cochant les cases des bonnes réponses. Les cases blanches ou complètement noircies seront considérées comme non cochées par la correction automatique.

Les questions se réfèrent à la figure ci-dessous quand cela est précisé.

0020

CPU 8086		1	
cs:0000	BBFF48 mov ax,48FF 4	ax 48FF	c=0
cs:0003	8ED8 mov ds,ax 2	bx 0000	z=0
cs:0005	BF1800 mov di,0018 2	cx 0000	s=0
cs:0008	C74502FFFF mov word ptr [di+02],FFFF 4	dx 0000	o=0
cs:000D	833D08 cmp word ptr [di],0008 2	si 0000	p=0
cs:0010	7F19 jg 002BA 4	di 0018	a=0
cs:0012	C7061A000100 mov word ptr [001A],0001 6	bp 0000	i=1
cs:0018	833E1B0000 cmp word ptr [001B],0000 2	sp 0000	d=0
cs:001D	74 je 002B	ds 48FF	
cs:001F	8B0D mov cx,[di]	es 48DF	
cs:0021	B80100 mov ax,0001	ss 48EE	
cs:0024	F7E1 mul cx	cs 4901	
cs:0026	E2 loop 0024	ip 0008	
cs:0028	A31A00 mov [001A],ax		
cs:002B	B8004C mov ax,4C00		
cs:002E	CD21 int 21		

Dump		2=[1111]	
ds:0000	43 43 32 20 41 4F 20 4C CC2 00 L	ss:0002	3A59
ds:0008	31 49 45 45 45 41 20 32 1IEEEA 2	ss:0000	0000
ds:0010	30 32 32 2F 32 30 32 33 022/2023	ss:FFFF	3302
ds:0018	05 00 00 00 00 00 00 00 *	ss:FFFC	4901

-252/4 = -63

Question 1 Donnez l'interprétation décimale signée de la valeur contenue dans le registre AL à l'issue de l'exécution de la suite d'instructions suivantes:

```
mov AX, 1100000100b
div AL
```

- 193 772 autre -63 -252

Question 2 Au total, combien de lectures en mémoire sont effectuées par le programme de la figure ?

- 16 25 28 17 18 20

Question 3 L'instruction suivante est-elle correcte ?

```
add AL, byte ptr [SI+2]
```

- Non Oui

Question 4 Dans quel registre sera stocké le résultat de la multiplication suivante ?

```
mov AX, 114h
mov BX, 0F36h
mul BH
```

- AL BX AX DX et AX BH

Question 5 Si on exécute les trois lignes de code suivantes, dans quel registre pourra-t-on récupérer le quotient de la division ?

```
mov AX, 0E2h
mov BP, -146
idiv BP
```

- DX BL AX AH BP AL

Question 6 Dans le cas du programme de la figure, quelle est l'adresse physique de la case mémoire adressée par l'instruction `mov word ptr [DI+02], 0FFFFh` ?

- 4902Ah FFFFh 48F0Ah D8:001Ah 4900Ah

Question 7 Quel(s) bit(s) du registre d'état est(sont) testé(s) par l'instruction `sub` ?

- ZF aucun SF CF OF tous

Question 8 Quel résultat, exprimé en hexadécimal, obtient-on dans AX à l'issue de l'exécution de la suite d'instructions suivante ?

```
mov AX, 11110010011b
mov CX, 101010101010101b
or CX, AX
```

- F06Ch 5AC6h 0511h 5FD7h 0F93h

Question 9 Dans quel registre sera stocké le résultat de la multiplication effectuée par la suite d'instructions suivante:

```
mov AX, 110000100b
mov BX, AX
mul BX
```

- AL DX et AX BX AX AX et BX

Question 10 Dans le cas du programme de la figure, quelle valeur pourra-t-on lire en hexadécimal à l'adresse logique DS:001Ah à la fin de l'exécution du programme ?

- 0001h 05h 0120h FFh 78h

Question 11 Quel mode d'adressage est utilisé pour désigner le premier opérande dans l'instruction `mov word ptr [DI+02], 0FFFFh` ?

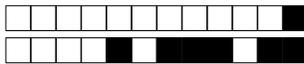
- immédiat registre direct indirect

Question 12 Qu'est-ce qu'une adresse effective ?

- un numéro de segment une adresse physique une valeur de déplacement dans un segment une adresse logique

Question 13 Quel registre est manipulé par l'instruction `loop` ?

- BX AX BP aucun DX CX



Question 14 Quelle instruction est équivalente à la suite d'instructions suivante ?
 dec CH
 cmp CH,0
 jne label

- jz label loop label aucune jmp label ja label

Question 15 Donnez l'interprétation décimale non signée de la valeur contenue dans le registre AL à l'issue de l'exécution de la suite d'instructions suivantes:
 mov AX, 10110111001b
 shr AX, 5

- 5 45 366 91 183

Question 16 Quel résultat obtient-on, en hexadécimal, dans CX à l'issue de l'exécution de la suite d'instructions suivante ?
 mov CX, 0D3FFh
 xor CL,CH

- D32Ch D300h 00FFh D3D3h

Question 17 Quel mode d'adressage est utilisé pour désigner le premier opérande dans l'instruction mov DS:[001Ah], AX ?

- direct indirect écriture immédiat registre

Question 18 Quel(s) bit(s) drapeau(x) du registre d'état sera(ont) à 1 à l'issue de l'exécution de la suite d'instructions suivantes ?
 mov AL, 12
 cmp AL, -16

- ZF CF aucun SF OF PF

Question 19 Si on exécute les trois lignes de code suivantes, dans quel registre pourra-t-on récupérer le reste de la division ?
 mov AX, 802h
 mov DX, 4
 div AL

- AL AX DX AX et DX AH

Question 20 Quel effet produit l'exécution de la suite d'instructions suivante:
 mov AX, 0FFBFh
 mov CX, 5D47h
 and CX, AX

- met à 0 le bit de rang 3 de AX met à 1 le bit de rang 5 de CX met à 0 le bit de rang 7 de AX met à 0 le bit de rang 3 de CX met à 0 le bit de rang 6 de CX

Question 21 Donnez le résultat obtenu dans le registre AX (en hexadécimal), à l'issue de l'exécution de la suite d'instructions suivante (indice: aidez-vous de la figure) :
 mov AX, 'M'
 add AL, 32

- 004D 109 006D 4D 'm'

Question 22 Dans le cas du programme de la figure, quelle est, en langage machine la valeur de l'opérande de l'instruction loop ?

- 2400 boucle FC 0024 F0

Question 23 Dans le cas du programme de la figure, quelle est, en langage machine la valeur de l'opérande de l'instruction je ?

- 002B 2B00 15 0F 0C

Question 24 L'instruction suivante est-elle correcte ?
 sub BL, byte ptr [BX+SI]

- Non Oui

Question 25 Quel est, en langage machine, le code operation (opcode) d'une instruction de la forme mov [offset], AX ?

- autre 8E 8ED8 A3 B8 A31A

Question 26 Le code suivant est-il correct ?
 mov AX, 0C9h
 div 3

- Non Oui

CPU 8086		-	
cs:0000 B8FF48	mov ax,48FF	ax 48FF	c=0
cs:0003 8ED8	mov ds,ax	bx 0000	z=0
cs:0005 BF1800	mov di,0018	cx 0000	s=0
cs:0008 C74502FFFF	mov word ptr [di+02],FFFF	dx 0000	o=0
cs:000D 833D08	cmp word ptr [di],0008	si 0000	p=0
cs:0010 7F19	jg 002B	di 0018	a=0
cs:0012 C7061A000100	mov word ptr [001A],0001	bp 0000	i=1
cs:0018 833E180000	cmp word ptr [0018],0000	sp 0000	d=0
cs:001D 74	je 002B	ds 48FF	
cs:001F 8B0D	mov cx,[di]	es 48DF	
cs:0021 B80100	mov ax,0001	ss 48EE	
cs:0024 F7E1	mul cx	cs 4901	
cs:0026 E2	loop 0024	ip 0008	
cs:0028 A31A00	mov [001A],ax		
cs:002B B8004C	mov ax,4C00		
cs:002E CD21	int 21		

[]=Dump		2=[1111]	
ds:0000 43 43 32 20 41 4F 20 4C CC2 A0 L	ss:0002 3A59		
ds:0008 31 49 45 45 45 41 20 32 11EEA 2	ss:0008 0000		
ds:0010 30 32 32 2F 32 30 32 33 022/2023	ss:FFFE 3302		
ds:001B 05 00 00 00 00 00 00 00 *	ss:FFFC 4901		