Examen Architecture des Ordinateurs
Examen session principale
ING 1 GI FA
2022 - 2023

Modalités :

- Durée : 2h
- Aucun document n'est autorisé, calculatrice autorisée
- Annexe inclue
- Pages 5 et 6 à compléter et à rendre

Exercice 1 : QCM (Pour chaque question, une seule réponse est correcte) (2 pts)

1. Quelles méthodes pouvons-nous utiliser pour simplifier les fonctions booléennes ?

❑ Théorèmes et propriétés de l'Algèbre de Boole ET Tableaux de Karnaugh
❑ Table de vérité ET Tableaux de Karnaugh
❑ Table de vérité ET Portes logiques
❑ Portes logiques ET Théorèmes et propriétés de l'Algèbre de Boole

2. La mémoire peut posséder au maximum :

❑ 2^n - 1 cases mémoires
❑ 2^{n+1} cases mémoires
❑ 2^n cases mémoires
❑ 2^{n-1} cases mémoires

3. L'unité de commande contient :

❑ Décodeur et Accumulateur
❑ Accumulateur et Séquenceur
❑ Compteur Ordinal et l'Unité Arithmétique et Logique
❑ Séquenceur et Compteur Ordinal

4. Soit le nombre représenté en virgule flottante selon la norme IEEE754 simple précision (32 bits). Quelle est la valeur décimale du nombre suivant : 1 1000 0011 10011100000000000000 ?

❑ 0,2575
❑ -25,75
❑ 25,75
❑ -0,2575

Exercice 2 : Circuit Logique (6 pts)

1. On désire effectuer un compteur synchrone modulo 7 à base de bascules JK synchronisées sur front montant. Un compteur modulo 7 est un type de compteur qui compte jusqu'à 6 puis recommence à 0.

a) Compléter la table de transition (**Tableau 1 – page 5**).

	Q2	Q1	Q0	J2	K2	J1	K1	J0	K0
0	0	0	0	0	Φ	0	Φ	1	Φ
1	0	0	1	0	Φ	1	Φ	Φ	1
2	0	1	0	0	Φ	Φ	0	1	Φ
3	0	1	1	1	Φ	Φ	1	Φ	1
4	1	0	0	Φ	0	0	Φ	1	Φ
5	1	0	1	Φ	0	1	Φ	Φ	1
6	1	1	0	Φ	1	Φ	1	0	Φ

D'après la table des transitions d'une bascule JK, on a **J0 = 1** et **K0 = Φ** lors d'une transition de 0 à 1 sur Q0.

b) Remplir les tableaux de Karnaugh correspondant aux fonctions J_i et K_i, puis donner l'équation réduite de chaque fonction.

K0 = 1
J1 = Q0
K2 = Q1

Q2 \ Q1 Q0	00	01	11	10
J0				
0	1	Φ	Φ	1
1	1	Φ	Φ	0

J0 = $\overline{Q1}$ + $\overline{Q2}$

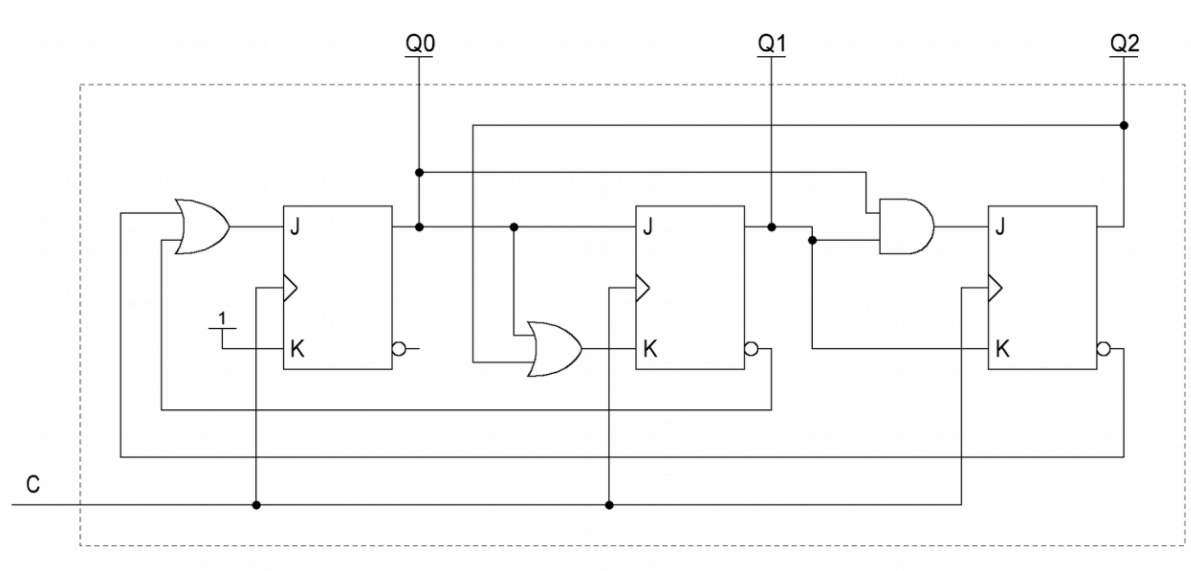
Q2 \ Q1 Q0	00	01	11	10
K1				
0	Φ	Φ	1	0
1	Φ	Φ	Φ	1

K1 = Q0 + Q2

Q2 \ *Q1 Q0*	00	01	11	10
J2				
0	0	0	1	0
1	Φ	Φ	Φ	Φ

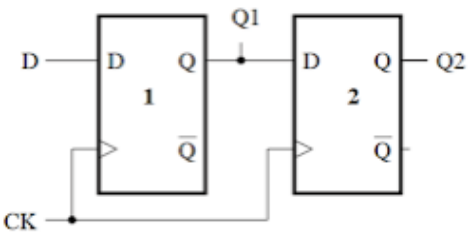
J2 = Q0.Q1

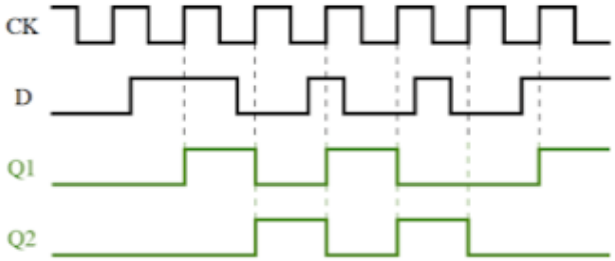
c) Compléter le schéma de câblage de ce compteur (**Schéma 1 - page 5**).



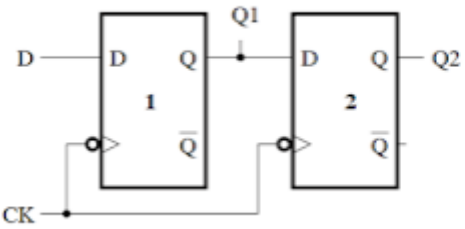
2. Soit les circuits logiques ci-dessous utilisant deux bascules D. Compléter le chronogramme de chaque circuit (**Schéma 2 (a) et Schéma 3 (b) – pages 5 et 6**).

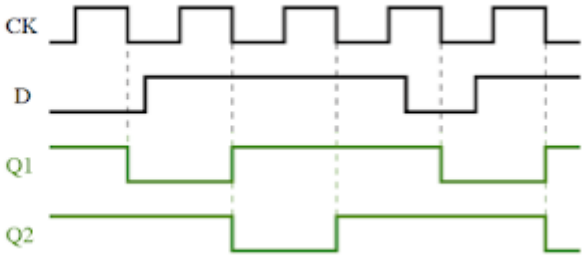
a)





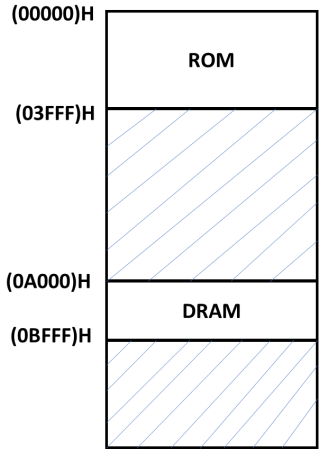
b)





Exercice 3 : Mémoire (5 pts)

Un microprocesseur a un bus d'adresse de 20 bits et un bus de données de 8 bits. La figure ci-dessous illustre la mémoire formée par une ROM et une DRAM, ainsi que les adresses de chaque partie de la mémoire.



1. Quel est l'espace adressable du processeur ?
2. Quelle est la capacité totale de cette mémoire ?
3. Quels sont les adresses de début et fin de cette mémoire ?
4. Quel est le nombre de bits permettant d'adresser la ROM ?
5. Quelle est la capacité de la ROM ?
6. Quelle est la capacité de la DRAM ?

1) Espace adressable du processeur.
20 bits d'adresse --> 2^20 adresses possibles --> espace adressable : 2^20 mots mémoires

2) Capacité totale de la mémoire
TMO = 1 octet
NA = 2^20
TME = NA * TMO = 2^20 * 1 = 2^20 octets = 1Mo

3) Les adresses de début et fin de cette mémoire
Adresse de début : $(00000)_{16}$ ou 0x00000
Adresse de fin : $(FFFFF)_{16}$ ou 0xFFFFF

4) Nombre de bits permettant d'adresser la ROM :
Plage d'adresses : 0x00000 à 0x03FFF --> $(03FFF)_{16} - (00000)_{16} = (3FFF)_{16}$
$= (11\ 1111\ 1111\ 1111)_2$
= 2^14 - 1
14 bits permettent d'adresser la ROM

5) Capacité de la ROM
Comme les adresses des mots mémoire de la ROM varient de 0 à 2^14 - 1, la ROM comprend 2^14 mots mémoire
Capacité ROM = Nombre de mots mémoire * taille mot mémoire
= 2^14 * 1 = 2^14 bits = 2^4 * 2^10 = 16 Ko

6) Quelle est la capacité de la DRAM
Nombre de mots mémoire de la DRAM :
$(0BFFF)_{16} - (0A000)_{16} + 1 = (1FFF)_{16} + 1 = (1\ 1111\ 1111\ 1111)_2 + 1 = 2^13$
Capacité DRAM = Nombre mots mémoire * taille mot mémoire
= 2^13 * 1 = 2^3 * 2^10 = 8 Ko

Exercice 4 : Jeux d'instruction (3 pts)

Soit l'extrait de programme ASSEMBLEUR INTEL 8086 suivant avec les valeurs initiales : AX = 0001_H, BX = 0000_H, le Flag z = 0 et l'état de pile suivant : SP = $FFFC_H$, $FFFE_H$ = 0001, $FFFC_H$ = 0002 et $FFFA_H$ = 0000.

Soit le code en assembleur suivant :
POP AX
MOV BX, $000B_H$
Boucle : ADD AX, 0002_H
SUB BX, 0009_H
CMP BX, 2
JNE Boucle
PUSH BX
PUSH AX

Compléter le tableau correspondant aux contenus des différents registres (**Tableau 2- page 6**) sachant que chaque ligne représente une étape d'exécution du code précédent.

Instruction	AX	BX	Flag z	SP	Stack :
Etat initial	0001	0000	0	FFFC	00 01 00 02 00 00
POP AX	0002	0000	0	FFFE	00 01 00 00 00 00
MOV BX, $000B_H$	0002	000B	0	FFFE	00 01 00 00 00 00
Boucle : ADD AX, 0002_H	0004	000B	0	FFFE	00 01 00 00 00 00
SUB BX, 0009_H	0004	0002	0	FFFE	00 01 00 00 00 00
CMP BX, 2	0004	0002	1	FFFE	00 01 00 00 00 00
JNE Boucle	0004	0002	1	FFFE	00 01 00 00 00 00
PUSH BX	0004	0002	1	FFFC	00 01 00 02 00 00
PUSH AX	0004	0002	1	FFFA	00 01 00 02 00 04

Exercice 5 : Assembleur NASM (4 pts)

Ecrire un programme qui affiche une chaîne de caractères saisie à partir du clavier.

```
section .data
    message db "Entrez une chaîne de caractères : ", 0
    buffer db 100 ; taille du tampon de saisie
    output db "Vous avez saisi : ", 0

section .text
    global _start

_start:
    ; Afficher le message demandant à l'utilisateur de saisir une chaîne de caractères
    mov eax, 4
    mov ebx, 1
    mov ecx, message
    mov edx, 29
    int 0x80
```

```
; Lire la saisie de l'utilisateur
mov eax, 3
mov ebx, 0
mov ecx, buffer
mov edx, 100
int 0x80

; Afficher la chaîne de caractères saisie
mov eax, 4
mov ebx, 1
mov ecx, output
mov edx, 17
int 0x80

; Afficher la saisie de l'utilisateur
mov eax, 4
mov ebx, 1
mov ecx, buffer
mov edx, 100
int 0x80

; Terminer le programme
mov eax, 1
xor ebx, ebx
int 0x80
```