Architecture des ordinateurs
Examen session principale
ING 1 GI
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. **Bus :**

a. Il y a quatre types de bus : données, adresse, commande et contrôle.
b. Un bus est un groupe de lignes électriques qui relie le CPU aux autres composantes.
c. Chaque ligne électrique d'un bus peut transférer deux bits d'information à la fois.
d. Le bus de données permet le transfert des données dans un seul sens.

2. **Adressage :**

a. Adressage registre est un adressage entre adresse mémoire et valeur implicite.
b. Adressage direct est un adressage entre deux registres.
c. Adressage immédiat est un adressage entre deux adresses mémoire.
d. Adressage indirect est un adressage entre un registre et une adresse mémoire.

3. **Soit le nombre -27 en base décimale. Comment écrire ce nombre en binaire complément à 2 sur 8 bits ?**

a. 1110 0101
b. 0001 1011
c. 1110 0100
d. 0001 1100

4. **L'unité de commande ne contient pas :**

a. Décodeur
b. Séquenceur
c. Accumulateur
d. Registre d'instruction

Exercice 2 : Circuit Logique (5,5 pts)

On désire effectuer la synthèse d'un compteur synchrone modulo 8 à base de bascule D. Un compteur modulo 8 est un type de compteur qui compte jusqu'à 7 puis recommence à 0.
On propose d'utiliser des bascules D à front montant.

1. Justifier le choix de **trois** bascules D.
2. Compléter la table de transition (***Tableau 1***).
3. Remplir les tableaux de Karnaugh correspondant à ces fonctions D_i, puis donner l'équation réduite de chaque fonction.

4. Compléter le schéma de câblage de ce compteur (***schéma 1***).
5. Compléter le logigramme correspondant.

Exercice 3 : (5,5 pts)

Nous disposons d'un ordinateur disposant de 512 Mo de mémoire vive dont la taille du mot mémoire est de 32 bits.

A.

1. Déterminer l'adresse la plus haute (la plus grande) de cette mémoire.
2. Convertir l'adresse trouvée en octal.

B. Un tableau de réels, selon la norme IEEE-754 double précision, est stocké dans la mémoire de cette machine à partir de l'adresse **0AE**.

3. Calculer l'adresse en décimal du 8ème élément de ce tableau.
4. Combien d'octets précédent l'adresse de début de ce tableau.

C. Nous désirons effectuer une addition sur deux éléments du tableau de réels.

6. Donner l'instruction en utilisant le mode d'adressage immédiat qui permet de modifier le contenu du registre **ESI**. **ESI** contiendra l'indice du cinquième élément du tableau.
7. Donner l'instruction en utilisant le mode d'adressage indexé pour accéder au cinquième élément du tableau, sachant que le registre de base contenant l'adresse de début du tableau est **EBX.**

Exercice 4 : jeux d'instruction (3 pts)

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

Soit le code en assembleur suivant :

```
POP AX
MOV BX, 000A H
MUL AX
Boucle : ADD AX, 0005 H
SUB BX, 0008 H
CMP BX, 2
JNE Boucle
PUSH BX
PUSH AX
```

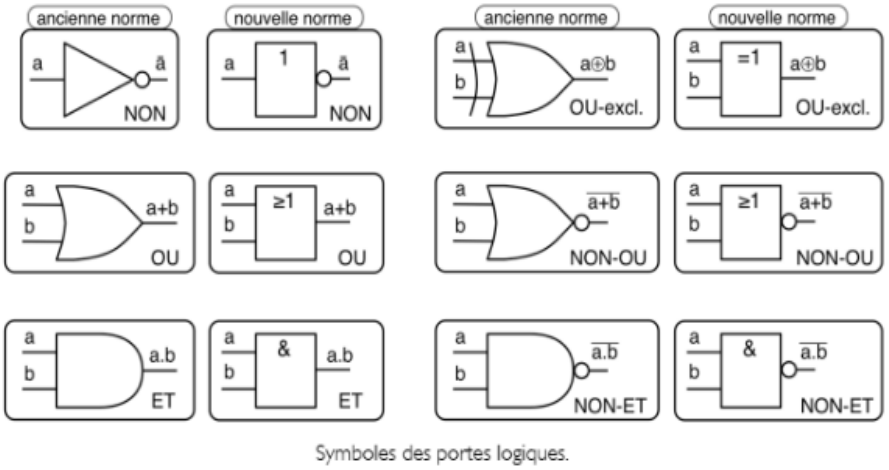
Compléter le tableau correspondant aux contenus des différents registres (***Tableau 2***) sachant que chaque ligne représente une étape d'exécution du code précédent.

Exercice 5 : Assembleur NASM (4 pts)

Écrire un programme assembleur bien commenté qui calcule la somme des carrés des entiers de 1 à 10 et affiche le résultat.

Annexe portes logiques et assembleur x86

Portes logiques



Symboles des portes logiques.

Assembleur

- **Empiler : push**

Syntaxe

push source

- ❏ copie le contenu de source au sommet de la pile
- ❏ commence par décrémenter esp de 4 puis effectue la copie
- ❏ source: adresse, constante ou registre

- **Dépiler: pop**

Syntaxe

pop destination

- ❏ copie les 4 octets qui se trouvent au sommet de la pile dans destination
- ❏ commence par effectuer la copie puis incrémente esp de 4
- ❏ destination: un registre ou une adresse

- **Comparaison: cmp**

Syntaxe

cmp destination, source

- ❏ Effectue l'opération destination - source
- ❏ destination: registre ou adresse
- ❏ source: constante, registre ou adresse
- ❏ Les valeurs des flags ZF , SF et CF sont éventuellement modifiées.

- **Incrémentation: Inc**

Syntaxe

Inc destination

❏ incrémente destination

- **Décrementation: Dec**

Syntaxe

Dec destination

❏ décremente destination

Instruction de saut inconditionnel : JMP

Syntaxe

jmp adr

❏ Va à l'adresse/ étiquette adr

Instructions de saut conditionnel

	Instruction	Description	Indicateurs
Valeurs non signées	JB	Jump below	CF = 1
	JBE	Jump below or equal	CF = 1 et ZF=1
	JA	Jump above	CF =0 et ZF=0
	JAE	Jump above or equal	CF=0
	JE/JZ	Jump if equal /Jump if zero	ZF=1
	JNE/JNZ	Jump if not equal/ Jump if not zero	ZF=0
Valeurs signées	JL	Jump less	SF=NOT OF
	JLE	Jump less or equal	ZF=1 ou SF=NOT OF
	JG	Jump greater than	ZF=0 et SF=OF
	JGE	Jump greater or equal	SF=OF

Rappel: ZF (zero flag), SF (sign flag), CF (carry flag), OF (overflow flag)

Pages à compléter et à rendre

Etat présent (Q_n)			Etat futur ($D_n = Q_{n+1}$)		
Q_2	Q_1	Q_0	D_2	D_1	D_0
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Tableau 1 -Table de transition d'un compteur modulo 8

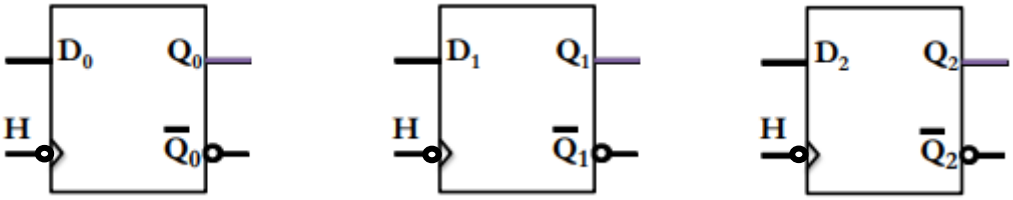


Schéma 1 (utilisez des couleurs si nécessaires)

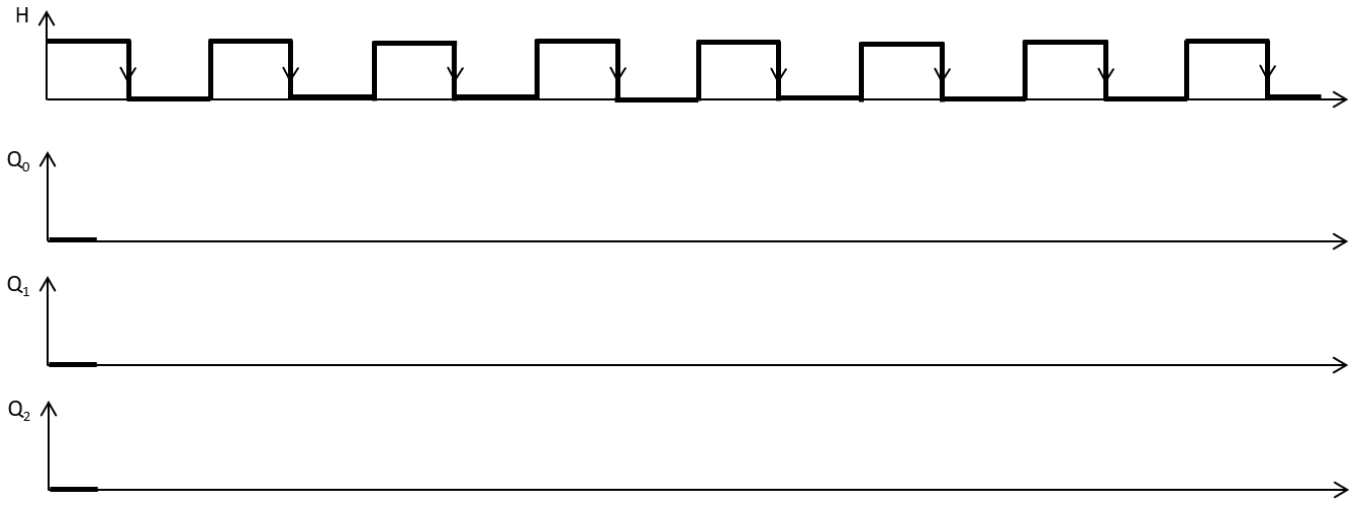


Schéma 2

	AX	BX	Flag z	SP	Stack: FFFF, FFFE, FFFD, FFFC, FFFB, FFFA
Etat initial					

Tableau 2