

**Examen**  
**14 janvier 2014**

Durée : 2h

Une feuille A4 recto-verso manuscrite autorisée (à rendre avec la copie).  
Le barème est indicatif. La notation tiendra compte de la validité des réponses  
mais aussi de la présentation et de la clarté de la rédaction.

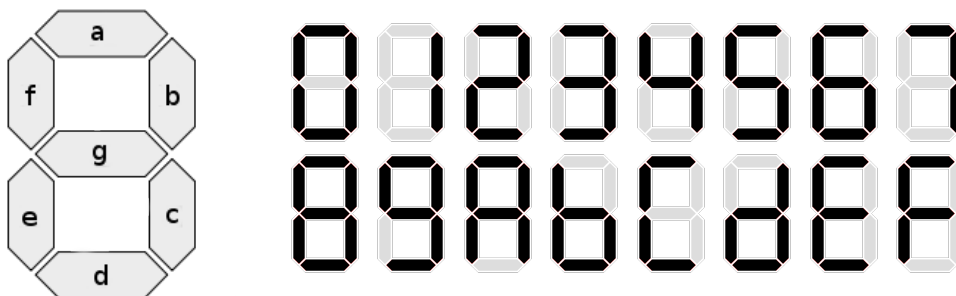
**1 (F1) Questions de cours (3 points)**

- 1.1 Sur quelle architecture repose l'ensemble des ordinateurs modernes? Citer les éléments qui la composent.
- 1.2 Citer l'ensemble des couches de l'architecture en couches vue en cours.
- 1.3 Quelles sont les deux catégories de circuits logiques étudiés? Quelle est leur différence?

**2 (F2) Circuits logiques (9 points)**

**2.1 Réalisation d'un afficheur (3 points)**

On cherche à réaliser un circuit afficheur hexadécimal pour une calculatrice. L'entrée est un nombre  $n$  en binaire sur 4 bits :  $b_0, b_1, b_2$  et  $b_3$ . Les sorties sont appelées  $a, b, c, d, e, f$  et  $g$ . Une sortie est à 1 si le segment correspondant est noir.



- 2.1.1 Écrire les tables de vérité des 7 sorties.
- 2.1.2 En déduire le circuit correspondant.

**2.2 Comparateur série (3 points)**

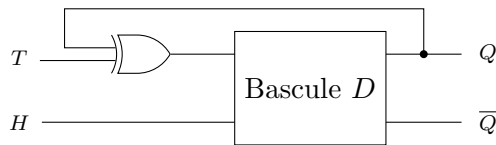
Un comparateur série est un circuit séquentiel synchrone qui permet de comparer deux suites de  $n$  bits  $a = a_0 \dots a_{n-1}$  et  $b = b_0 \dots b_{n-1}$ . Les séquences  $a$  et  $b$  sont traitées de gauche à droite et à chaque

front montant d'horloge, deux bits correspondants sont comparés et la sortie est mise à jour. La sortie est finalement à 1 si les deux suites sont égales et à 0 sinon.

1. Le circuit admet deux entrées  $a$  et  $b$  (en plus de l'entrée correspondant à l'horloge), et une sortie  $R$ . Donner la table d'états de ce circuit. En déduire une expression booléenne exprimant  $R$  en fonction de  $a$ ,  $b$  et  $r$  (valeur de  $R$  à l'étape précédente).
2. En sachant qu'il est possible d'utiliser une bascule  $D$  pour se souvenir de la valeur précédente de la sortie  $R$ , construire un circuit séquentiel synchrone réalisant le comparateur série.

### 2.3 Bascule T synchrone (3 points)

Soit la bascule T suivante :



- Elle est composée d'une bascule  $D$  flip-flop.
- L'entrée H est synchronisée sur l'horloge.

Sachant le comportement d'une bascule  $D$  flip-flop :

**2.3.1** Déterminer la table de vérité de la bascule  $T$  selon les entrées  $T$  et  $Q$ . (On suppose que la bascule est active uniquement lorsque  $H = 1$ , donc on ne traitera pas les cas où  $H = 0$ ).

**2.3.2** En déduire une expression booléenne pour  $Q$ .

## 3 (F1) Jeu d'instructions et langage d'assemblage (8 points)

Soit le jeu d'instructions suivant :

Instruction	Description
MPUSH <i>mot</i>	Place le mot <i>mot</i> au sommet de la pile
VPUSH <i>var</i>	Place la variable locale désignée par <i>var</i> au sommet de la pile
POP <i>var</i>	Supprime le sommet de la pile et le place dans la variable désignée par <i>var</i>
ADD	POP les 2 mots au sommet de la pile et PUSH leur somme dans la pile
INC	Ajoute 1 au sommet de la pile
SUB	POP les 2 mots au sommet de la pile et PUSH leur différence dans la pile
MUL	POP les 2 mots au sommet de la pile et PUSH leur produit dans la pile
NOT	Remplace le sommet de la pile par sa négation
AND	POP les 2 mots au sommet de la pile et PUSH leur ET logique dans la pile
JUMP <i>label</i>	Saute vers l'étiquette <i>label</i>
JE0 <i>label</i>	Saute vers l'étiquette <i>label</i> si le mot au sommet de la pile vaut 0
JGE0 <i>label</i>	Saute vers l'étiquette <i>label</i> si le mot au sommet de la pile est supérieur ou égal à 0.
NOP	Ne fait rien

Un *mot* correspond à une suite de bits sur une taille donnée (dépend du micro-processeur). Plus simplement, cela permet d'indiquer directement une valeur (une constante) sans passer par une variable.

Les opérations de type SUB et DIV sont basées sur l'ordre d'empilement.

**3.1** Traduire les pseudo-codes suivants en instructions :

j=1;	i=10; j=0;
pour (i=0; i != 10; i++)	tant que (i>0) alors
j=j*2	j=j+(i*i);
fin pour	i--;
	fin tantque

(a)

(b)

**3.2** Donner un pseudo-code correspondant aux instructions suivantes :

```
MPUSH 6
POP i
MPUSH 1
POP k
VPUSH k
POP j
JUMP L1
```

```
L1: VPUSH i
    VPUSH k
    SUB
    JGEO L2
    JUMP L3
```

```
L2: VPUSH j
    VPUSH k
    MUL
    POP j
    VPUSH k
    INC
    POP k
    JUMP L1
```

```
L3: VPUSH j
    POP res
```

**3.3** Le programme précédent correspond à une fonction mathématique connue. Laquelle ?

## 4 (F2) BONUS : arithmétique binaire (2 points)

Déterminer le résultat des opérations binaires suivantes, telles que :

- $N1 = 01101110$
- $N2 = 00011101$
- $N3 = 10110011$

**4.1**  $N1 + N2$

**4.3**  $N1 * 2$

**4.5**  $N3 * 4$

**4.2**  $N3 - N1$

**4.4**  $N1 / 2$