

## EXAMEN DE SYNTHESE D'INFORMATIQUE

### Documents et calculatrice non autorisés

#### Questions de cours (4 pts)

- 1) A quoi sert le registre d'état et de combien de bits est- il généralement constitué ?
- 2) Quel est le rôle du bloc logique de commandes ?
- 3) Quelles sont les principales tâches du système d'exploitation ?
- 4) Donner un exemple de système d'exploitation mono tâche mono utilisateur?
- 5) Représenter par un schéma les principaux composants de la machine de Von Neumann ?
- 6) Quels sont les différents bus système et quel est le rôle de chacun d'entre eux ?
- 7) Que signifie le multi processing ?
- 8) Définir la notion de mémoire virtuelle ?

#### Exercice 1 (4 pts)

- 1) Effectuer les conversions suivantes:

$$(3AE)_{16} = ( )_{10}$$

$$(56,25)_{10} = ( )_2$$

- 2) Convertir le nombre  $(-8,625)_{10}$  en virgule flottante suivant la norme IEEE 754.

- 3) - Comment sont-ils représentés les nombres suivants sur une machine de 8 bits:

$$(+97)_{10} = ( )_{cà2}$$

$$(-34)_{10} = ( )_{cà2}$$

$$(10110101)_2 = ( )_{10}$$

- Effectuer en complément à 2 l'opération  $(+97)_{10} + (-34)_{10}$

- 4) Dans quelle base B est effectuée cette opération (avec une démonstration)

$$(57)_B + (33)_B = (112)_B$$

### **Exercice 2 (4pts)**

- 1) Montrer que les équivalences suivantes sont vraies dans l'algèbre de Boole :

$$x + x \cdot y = x$$

$$x \cdot (x + y) = x$$

$$\bar{x} \cdot (x + y) = \bar{x} \cdot y$$

- 2) Simplifier en utilisant la méthode algébrique les expressions logiques suivantes :

$$F = (\bar{a} + b) \cdot (a + b + d) \cdot \bar{d}$$

$$G = (a + b) \cdot (a + c) + (b + c) \cdot (b + a) + (c + a) \cdot (c + b)$$

$$H = a \cdot b \cdot c + a \cdot \bar{b} \cdot c + a \cdot b \cdot \bar{c}$$

### **Exercice 3 (5pts)**

- I. Soit l'expression Booléenne :

$$E1(a, b, c, d) = \bar{a}\bar{b}\bar{d} + abd + a\bar{b}\bar{c}d + a\bar{b}cd + ab\bar{c}d + \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}cd + a\bar{b}\bar{c}\bar{d}$$

- 1) Tracer la table de vérité correspondante.
- 2) - Donner la 2<sup>ème</sup> forme canonique (produit de somme).  
- Simplifier la fonction en utilisant le tableau de Karnaugh.
- 3) Tracer le logigramme correspondant à la fonction simplifiée en utilisant que :
  - Les portes ET (and), OU (or) et les inverseurs.
  - Les portes NAND (Non- et) et les inverseurs

- II. Simplifier l'expression en utilisant le tableau de Karnaugh

$$E2(a, b, c, d) = \bar{a} + a \cdot b + a \cdot \bar{b} \cdot c + a \cdot \bar{b} \cdot c \cdot d$$

### **Exercice 4 (3pts)**

Donner les différentes étapes d'exécution du programme suivant :

Program puissance

Begin

  Read(x);

  p := x\*x;

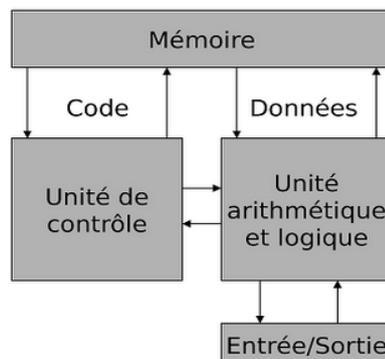
  Write(p) ;

End

## CORRECTION DE L'EXAMEN DE SYNTHÈSE

### RÉPONSES DE COURS (4pts)

- 1) - Le **registre d'état** sauvegarde l'état de l'UAL après exécution d'opération  
- Ce registre est généralement sur 8bits
- 2) **Le bloc logique de commandes** (le séquenceur) sert à de synchroniser l'exécution des instructions au rythme d'une horloge. Il est ainsi chargé de l'envoi des signaux de commande.
- 3) **Les tâches** du SE
  - ✓ *Un utilisateur exécute des tâches variées sur l'ordinateur:*
    - \* Il peut demander à exécuter un programme
    - \* Il peut lire des données enregistrées sur un support tel qu'un CD, disque, clé USB
    - \* Il peut manipuler d'une façon particulière des fichiers (copier,...)
    - \* Il peut demander à imprimer des résultats
    - \* Il peut partager l'ordinateur avec d'autres utilisateurs, qui peuvent eux-mêmes demander simultanément des programmes
  - ✓ *Plusieurs autres tâches sont fournies à l'utilisateur:*
    - \* Contrôler l'exécution des programmes
    - \* Gérer l'utilisation de la mémoire
    - \* Gérer les opérations d'E/S
    - \* Gérer les fichiers
    - \* Partager les ressources de l'ordinateur entre les utilisateurs
    - \* Protéger l'ordinateur et ses utilisateurs
    - \* Détecter les erreurs lors de l'utilisation ...
- 4) **Exemple** de SE mono tâche mono utilisateur : MSDOS
- 5) **Composants** de la machine de Von Neumann



6) **Les bus**

- ***Bus d'Adresse*** : Transporte les adresses mémoire auxquelles le processeur souhaite accéder pour lire ou écrire une donnée.
- ***Bus de données*** : Véhicule les instructions en provenance ou à destination du processeur.
- ***Bus de commande*** : Transporte les ordres en provenance de l'unité de commande et à destination de l'ensemble des composants matériels.

7) Le **multi-processing** est une technique consistant à faire fonctionner plusieurs processeurs en parallèle afin d'obtenir une puissance de calcul plus importante que celle obtenue avec un processeur haut de gamme ou bien afin d'augmenter la disponibilité du système (en cas de panne d'un processeur).

8) La **mémoire virtuelle** est une mémoire artificielle qui repose sur l'utilisation d'une mémoire de masse. C'est une technique avancé et sophistiqué dans la gestion de la mémoire combinant une mémoire centrale et une mémoire auxiliaire donnant au programmeur l'illusion d'un espace mémoire immense.

**CORRECTION EXERCICE 1 (4pts)**

1) Les conversions

$$(3AE)_{16} = (942)_{10}$$

$$(56,25)_{10} = (00111000,010)_2$$

2) La conversion de  $(-8,625)_{10}$  en virgule flottante suivant la norme IEEE 754

- $(8,625)_{10} = (1000,101)_2$
- $1000,101 \times 2^0 \Leftrightarrow 1,0001010 \times 2^3$  (pseudo normalisation)
- Exposant biaisé sur  $\Rightarrow 3 + 127 = (130)_{10} \Rightarrow (10000010)_2$

Signe (1bit)	Exposant biaisé (8bits)	Pseudo mantisse (23bits)
<b>1</b>	<b>1000 0010</b>	<b>0 00101 0000 0000 0000 0000</b>

3) Représentation sur une machine de 8bits:

- $(+97)_{10} = (01100001)_{cà2}$
- $(-34)_{10} = (11011110)_{cà2}$
- $(10110101)_2 = (-53)_{10}$
- $(+97)_{10} + (-34)_{10} = (01100001)_{cà2} + (11011110)_{cà2} = (10011111)_{cà2} = (+63)_{10}$

4) Déduction de la Base B (avec DEMONSTRATION)

$$(57)_B + (33)_B = (112)_B$$

$$\Rightarrow 7.B^0 + 5.B^1 + 3.B^0 + 3.B^1 = 2.B^0 + 1.B^1 + 1.B^2$$

$$\Rightarrow B^2 - 7.B^1 - 8.B^0 \text{ (Equation du second degré)}$$

Delta=81  $\Rightarrow$  2solutions: B= -1 (Rejetée) ou **B=8** (Acceptée)  
 Donc **B=8**

**CORRECTION EXERCICE 2 (4pts)**

1) **Démonstration** des équivalences

$$x + x \cdot y = x(1+y) = x$$

$$x \cdot (x + y) = x \cdot x + x \cdot y = x + x \cdot y = x$$

$$\bar{x} \cdot (x + y) = \bar{x} \cdot x + \bar{x} \cdot y = \bar{x} \cdot y$$

2) Simplification algébrique

a)  $F = (\bar{a} + b) \cdot (a + b + d) \cdot \bar{d}$

$$F = (\bar{a} + b) \cdot (a + b + d) \cdot \bar{d}$$

$$F = (\bar{a} + b) \cdot (a \cdot \bar{d} + b \cdot \bar{d} + d \cdot \bar{d})$$

$$F = (\bar{a} + b) \cdot (a \cdot \bar{d} + b \cdot \bar{d})$$

$$F = \bar{a} \cdot b \cdot \bar{d} + a \cdot b \cdot \bar{d} + b \cdot \bar{d}$$

$$F = b \cdot \bar{d} \cdot (\bar{a} + a + 1)$$

$$F = b \cdot \bar{d}$$

b)  $G = (a + b) \cdot (a + c) + (b + c) \cdot (b + a) + (c + a) \cdot (c + b)$

$$G = (a + b) \cdot (a + c) + (b + c) \cdot (b + a) + (c + a) \cdot (c + b)$$

$$G = a \cdot a + a \cdot c + b \cdot a + b \cdot c + b \cdot b + b \cdot a + b \cdot c + c \cdot a + c \cdot c + c \cdot b + a \cdot c + a \cdot b$$

$$G = a + a \cdot c + a \cdot b + b \cdot c + b + c$$

$$G = a(1 + c + b) + b(c + 1) + c$$

$$G = a + b + c$$

c)  $H = a \cdot b \cdot c + a \cdot \bar{b} \cdot c + a \cdot b \cdot \bar{c}$

$$H = a \cdot b \cdot c + a \cdot \bar{b} \cdot c + a \cdot b \cdot \bar{c}$$

$$H = a \cdot (b \cdot c + \bar{b} \cdot c + b \cdot \bar{c})$$

$$H = a \cdot [b \cdot (c + \bar{c}) + \bar{b} \cdot c]$$

$$H = a \cdot (b + \bar{b} \cdot c)$$

$$H = a \cdot [(b + \bar{b}) \cdot (b + c)]$$

$$H = a \cdot (b + c)$$

**CORRECTION EXERCICE3 (5pts)**

**I. PARTIE 1**

$$E1(A, B, C, D) = \bar{a}\bar{b}\bar{d} + abd + a\bar{b}\bar{c}d + a\bar{b}cd + ab\bar{c}d + \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}cd + a\bar{b}\bar{c}\bar{d}$$

1) Table de Vérité -TV-

a	b	c	d	E(a,b,c,d)
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

2)

- 2<sup>ème</sup> forme canonique

$$E_1(A,B,C,D) = (a + b + c + \bar{d})(a + b + \bar{c} + d)(a + \bar{b} + c + \bar{d})(a + \bar{b} + \bar{c} + \bar{d})(\bar{a} + b + c + d)(\bar{a} + b + c + d)$$

- **Tableau de Karnaugh**

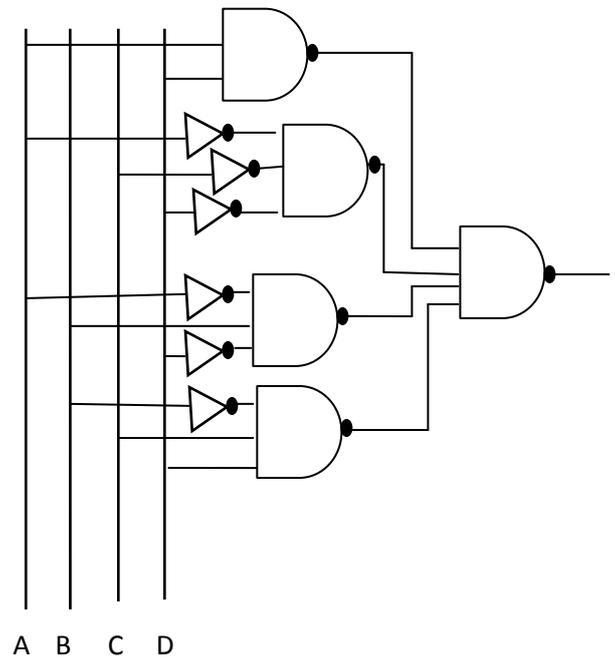
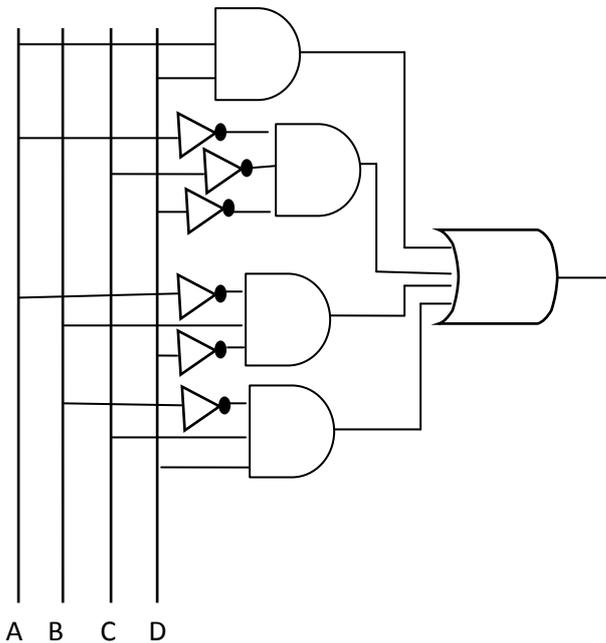
c d \ a b	00	01	11	10
00	1	0	1	0
01	1	0	0	1
11	0	1	1	0
10	1	1	1	0

$$E_1(a,b,c,d) = ad + \bar{a}\bar{c}\bar{d} + \bar{a}b\bar{d} + \bar{b}cd$$

3) **Logigramme**

-Portes ET, OU et les inverseurs.

- Portes NAND (Non-et) et les inverseurs



**II. PARTIE 2**

$$E2(A, B, C, D) = \bar{A} + A.B + A.\bar{B}.C + A.\bar{B}.C.D$$

**Tableau de Karnaugh**

Remplissage de la table

$AB \backslash CD$	$CD$	$\bar{C}D$	$C\bar{D}$	$\bar{C}\bar{D}$
$AB$	1	1	1	1
$\bar{A}B$	1	1	1	1
$\bar{A}\bar{B}$	1	1	1	1
$A\bar{B}$	1	0	0	1

$$\Rightarrow E2(A, B, C, D) = \bar{A} + B + C$$

**SOLUTION EXERCICE 4 (3pts)**

❖ L'instruction Begin :

- 1- Le PC contient une adresse (par exemple 100), cette adresse est placée dans le BUS d'adresse par l'unité de commande (le bloc logique de commande ou séquenceur).
- 2- Commande de lecture est envoyée a la MC pour lire le contenu de l'adresse (le contenu de l'@100).
- 3- L'instruction *begin* est placée dans le bus de données.
- 4- L'instruction *begin* est placée dans le registre d'instruction le RI qui se trouve dans unité de commande
- 5- L'instruction *begin* est décodée par le décodeur d'instruction se trouvant dans l'unité de commande et déduit que c'est le commencement du programme.
- 6- Le micro-programme se trouvant dans le bloc logique de commande réalisant l'instruction est exécuté
- 7- L'unité de commande positionne le PC pour l'instruction suivante c'est-à-dire le PC contient l'adresse @101.

❖ L'instruction **Read(x)** :

- 1- Le PC contient l'@101
- 2- L'UCC décode l'instruction et identifie que c'est une instruction de lecture (ou d'introduction de données).
- 3- Le RI contient *read x*
- 4- Le bloc logique envoie un ordre au clavier par le bus de commandes
- 5- La donnée **a** est introduite via le clavier par exemple 8.
- 6- La donnée est chargée dans la **MC** et occupe l'emplacement x. en utilisant le bus de données.

❖ L'instruction **p:=x\*x** :

- 1- L'instruction est envoyée vers l'UCC par le bus de données
- 2- L'UCC décode l'instruction et identifie qu'il s'agit d'une opération de multiplication et d'affectation.
- 3- L'instruction est transformée en une suite de commandes (phase de décodage réalisée par le décodeur d'instruction).
- 4- L'UCC envoie l'ordre vers l'UAL se trouvant dans l'unité de traitement
- 5- Le contenu de **x** est envoyé vers l'UAL par le bus de données
- 6- L'UAL effectue la multiplication **x\*x** ce qui donne **8\*8=64**
- 7- L'UAL envoie le résultat vers la MC et occupe l'emplacement **p**.

❖ L'instruction **write(p)**:

- 1- L'instruction est envoyée vers l'UCC.
- 2- L'UCC décode l'instruction et identifie que c'est une instruction d'affichage.
- 3- L'UCC envoie l'ordre de lecture de la variable **p** de la MC.
- 4- Une commande d'E/S est adressée au périphérique concerné (l'écran).
- 5- Le contenu de **p** est transféré pour son affichage.

❖ L'instruction **End** :

- 1- L'instruction est envoyée vers l'UCC.
- 2- L'UCC décode l'instruction et identifie que c'est une instruction indiquant la fin du programme.
- 3- L'UCC ordonne la fin du programme.