

BANQUE D'EPREUVES DUT-BTS

- SESSION 2008 -

**EPREUVE
D'INFORMATIQUE**

CODE EPREUVE : BE-INFO

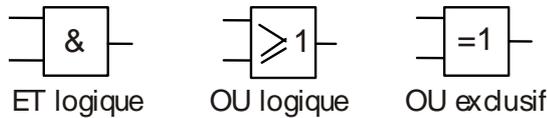
CALCULATRICE INTERDITE

L'épreuve comporte 19 questions regroupées suivant les thèmes : logique, électronique numérique, structure des machines, programmation, numération, systèmes d'exploitation, graphes, réseaux.

DUREE : 2H30

Logique

La signification des symboles est la suivante :



‘ \oplus ’ représente le OU exclusif,
 ‘+’ représente le OU logique,
 ‘ \cdot ’ représente le ET logique.

Question 1

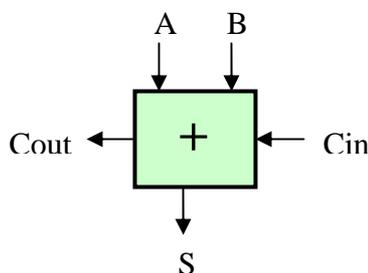
On veut construire un comparateur de deux nombre A et B codés en binaire naturel sur 2 bits ($A=A_12^1+ A_02^0$, $B=B_12^1+ B_02^0$). Le résultat y sera exprimé en binaire sur 3 bits : y_0 , y_1 et y_2 . y_0 vaut 1 si et seulement si $A < B$, y_1 vaut 1 si et seulement si $A = B$ et y_2 vaut 1 si et seulement si $A > B$.

- (A) Quelque soit les nombres A et B, on a l’expression $y_0 \oplus y_1 \oplus y_2 = 0$,
- (B) Quelque soit les nombres A et B, on a l’expression $y_0 \oplus y_1 \oplus y_2 = 1$,
- (C) L’expression simplifiée de y_1 est : $y_1 = (A_1 \oplus B_1) \cdot (A_0 \oplus B_0)$,
- (D) L’expression simplifiée de y_1 est : $y_1 = \overline{A_1 \oplus B_1} \cdot \overline{A_0 \oplus B_0}$,
- (E) Les expressions de y_0 , y_1 et y_2 vérifient les deux égalités : $y_0 \cdot y_1 \cdot y_2 = 0$, $y_0 + y_1 + y_2 = 1$

Electronique numérique

Question 2

Un additionneur complet (cellule *FullAdder*) réalise l’addition de deux bits A et B, plus une retenue entrante Cin, en produisant un bit de résultat S et un bit de retenue Cout.



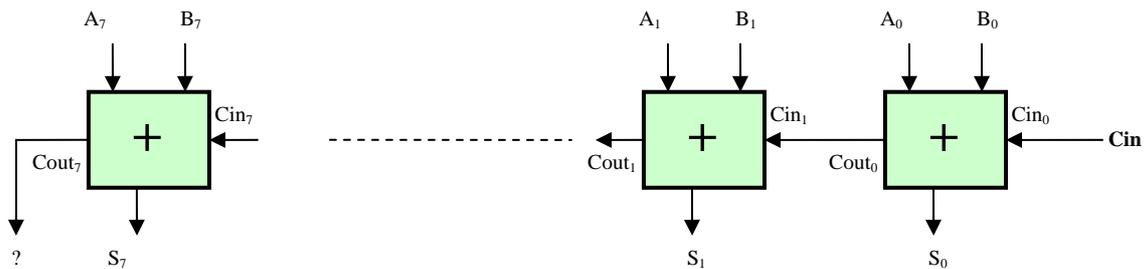
A	B	Cin	Cout	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

- (A) En forçant la retenue Cin à ‘1’, il est possible d’utiliser cet additionneur pour calculer le produit de 2 bits
- (B) L’équation simplifiée du bit de résultat est : $S = A \oplus B \oplus Cin$
- (C) L’équation simplifiée du bit de retenue est : $Cout = A \cdot B + A \cdot Cin + B \cdot Cin$

- (D) L'équation simplifiée du bit de retenue est : $Cout = A \cdot B + (A \oplus B) \cdot Cin$
 (E) L'équation simplifiée du bit de résultat est : $S = A + B + Cin$

Question 3

En cascadeant la cellule précédente, on réalise l'addition de deux mots **A** et **B** sur 8 bits interprétés en complément à 2, plus une retenue entrante **Cin** sur 1 bit (figure ci-dessous). Le résultat **S**, interprété en complément à 2, est sur 8 bits (on ne tient pas compte de la retenue finale **Cout₇**).



Les valeurs de A, B et S proposées dans toute cette question sont données soit en binaire (indice 2) soit en décimal (indice 10).

- (A) Si $Cin=0$, $A=01001011_2$ et $B=11000110_2$ alors $S=12_{10}$
 (B) Si $Cin=0$, $A=25_{10}$ et $B=-35_{10}$ alors $S=10001010_2$
 (C) Si $Cin=0$, $A=-25_{10}$ et $B=-35_{10}$ alors $S=10111100_2$
 (D) Pour réaliser la soustraction de deux nombres $N1$ et $N2$ ($S=N1 - N2$), il suffit de forcer l'entrée Cin à 1, $A = N1$ et $B = \overline{N2}$ (où $\overline{N2}$ représente le mot obtenu en inversant tous les bits de $N2$)
 (E) Pour réaliser la soustraction de deux nombres $N1$ et $N2$ ($S=N1 - N2$), il suffit de forcer l'entrée Cin à 0, $A = N1$ et $B = \overline{N2}$ (où $\overline{N2}$ représente le mot obtenu en inversant tous les bits de $N2$)

Question 4

Soit le tableau de Karnaugh de la fonction logique $S2$.

		BA			
		00	01	11	10
C	0	0	1	0	0
	1	1	1	0	1

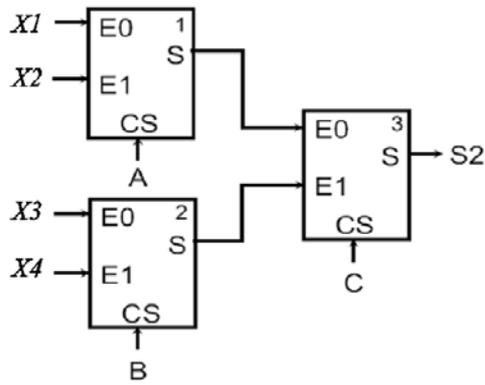
- (A) On peut écrire la fonction $S2$ comme suit :

$$S2 = C\overline{A} + \overline{B}A,$$

 (B) La fonction réalisée est celle d'un multiplexeur 2 vers 1, C étant l'entrée de sélection,
 (C) On peut simplifier l'expression de la fonction $S2$ en :

$$S2 = C\overline{A} + C\overline{B} + \overline{B}A$$

On cherche à réaliser $S2$ à l'aide du schéma de la figure suivante en positionnant des valeurs sur les entrées $E0$ et $E1$ des multiplexeurs 1 et 2. Les éléments 1 à 3 sont des multiplexeurs à 2 entrées. Pour chacun d'entre eux, la sortie S vaut $E0$ lorsque $CS=0$ et $E1$ lorsque $CS=1$



(D) La fonction $S2$ peut être réalisée en forçant les entrée $X1$ à $X4$ comme suit :

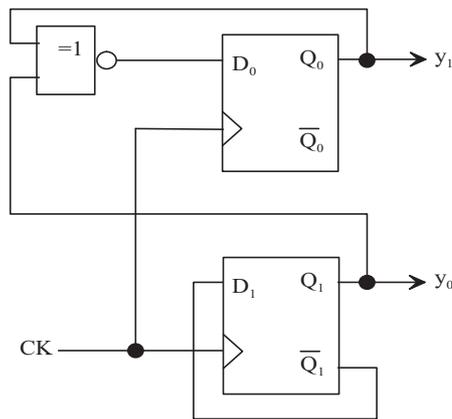
$$X1 = 0, X2 = \bar{A}, X3 = 1 \text{ et } X4 = B$$

(E) La fonction $S2$ peut être réalisée en forçant les entrée $X1$ à $X4$ comme suit :

$$X1 = 0, X2 = \bar{B}, X3 = 1 \text{ et } X4 = \bar{A}$$

Question 5

Considérons le circuit séquentiel ci-dessous. Les sorties sont données dans l'ordre y_1y_0 .



(A) La séquence réalisée est : $00 \rightarrow 01 \rightarrow 10 \rightarrow 11 \rightarrow 00 \dots$

(B) La séquence réalisée est : $00 \rightarrow 11 \rightarrow 10 \rightarrow 01 \rightarrow 00 \dots$

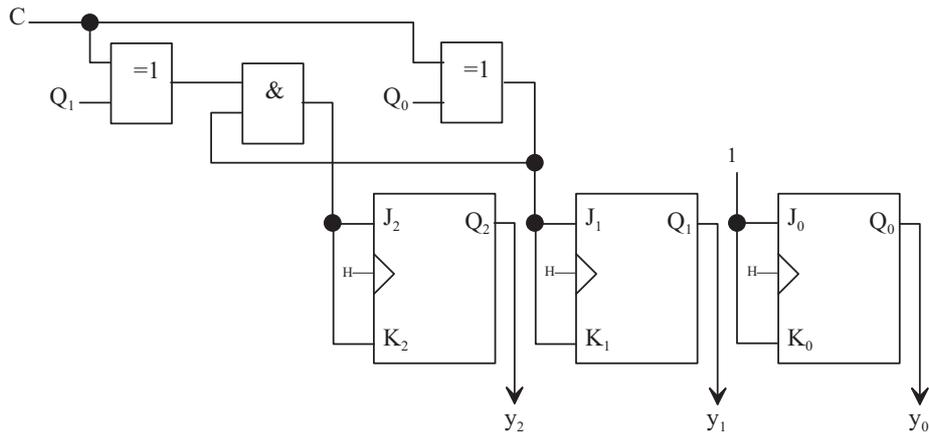
(C) La séquence réalisée est : $00 \rightarrow 10 \rightarrow 11 \rightarrow 01 \rightarrow 00 \dots$

(D) En remplaçant le OU exclusif inverseur par un OU exclusif non inverseur on obtient la séquence suivante : $00 \rightarrow 01 \rightarrow 10 \rightarrow 11 \rightarrow 00 \dots$

(E) En remplaçant les deux bascules D par deux bascules JK (avec les entrées J et K reliées, $J=K$) la séquence réalisée est : $11 \rightarrow 01 \rightarrow 11 \dots$

Question 6

Considérons le circuit séquentiel ci-dessous. Les sorties sont données dans l'ordre $y_2y_1y_0$.



- (A) Si $C=0$ le circuit réalise la séquence : $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 0 \dots$
 (B) Si $C=0$ le circuit réalise la séquence : $0 \rightarrow 7 \rightarrow 6 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0 \dots$
 (C) Si $C=1$ le circuit réalise la séquence : $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 0 \dots$
 (D) Si $C=1$ le circuit réalise la séquence : $0 \rightarrow 7 \rightarrow 6 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0 \dots$
 (E) Le circuit est un compteur/décompteur asynchrone

Numération

Question 7

Dans toute la question, les nombres sont considérés comme étant codés sur **8 bits** et interprété en complément à deux. Si $(a_{n-1}, a_{n-2}, \dots, a_0)_2$ dénote un nombre en base 2 et interprété en complément à deux, sa valeur décimale est donnée par :

$$A = -a_{n-1} \cdot 2^{n-1} + \sum_{i=0}^{n-2} a_i \cdot 2^i$$

Dans toute la question, l'indice associé à chaque nombre indique sa base de numération.

- (A) Le complément à deux du nombre $(10110011)_2$ est $(+77)_{10}$
 (B) La valeur décimale du nombre $(10110100)_2$ est $(-77)_{10}$
 (C) Le complément à deux de $(-25)_{10}$ est $(10011001)_2$
 (D) L'opération $(10011100)_2 + (11100101)_2$ provoque un dépassement
 (E) L'addition suivante $(10011100)_2 + (11100101)_2$ donne le résultat $(10000001)_2$ et ne provoque pas de dépassement

Programmation

Question 8

Soit la fonction suivante écrite en langage C :

```
int fonction(int n)
{
    if (n == 1) return(1);
    else return(n*n + fonction(n-1));
    printf("n=%d", n); /* imprimer la valeur de n en décimal */
}
```

- (A) Les valeurs renvoyées par cette fonction sont toujours positives,

- (B) Lorsque cette fonction est appelée avec $n=4$, la valeur renvoyée vaut 29,
- (C) Cette fonction permet de calculer la factorielle d'un entier naturel,
- (D) Lorsque cette fonction est appelée avec $n=5$, la valeur renvoyée vaut 55,
- (E) Lorsque cette fonction est appelée avec $n>1$, la valeur de n est imprimée avant la terminaison de la fonction.

Question 9

Soit la fonction suivante écrite en langage C :

```
void fonction(int n, int b)
{
    char table[] = "0123456789abcdef";
    if (n >= b)
        fonction(n/b, b);
    printf("%c", table[n%b]);
}
```

L'opérateur % calcule le reste de la division entière (modulo).

- (A) Cette fonction permet d'imprimer la représentation binaire d'un entier n ,
- (B) Cette fonction ne peut fonctionner correctement que si n est positif,
- (C) L'appel suivant `fonction(64, 1)` conduit à une boucle infinie
- (D) L'appel suivant `fonction(64, 8)` imprime 100,
- (E) L'appel suivant `fonction(480, 16)` imprime 50.

Question 10

On considère la fonction suivante dont l'algorithme est écrit en pseudo-langage :

```
PROCEDURE suite(n : ENTIER POSITIF)
DEBUT
    VARIABLES : u1, u2, u : ENTIER
    u1 ← 0, u2 ← 1, u ← 1
    SI (n EGAL A 1) ALORS u ← 1
    SINON
        POUR i VARIANT DE 1 A n FAIRE
            u ← u1 + u2
            u1 ← u2
            u2 ← u
            AFFICHE(u)
        FIN POUR
    FIN DE SI
    AFFICHE(u)
FIN
```

- (A) Pour $n=5$ l'appel de la procédure `suite` conduit à l'affichage : 1, 2, 4, 5, 5
- (B) Cette procédure calcule une suite dont les 4 premières valeurs sont : 1,2,3,5
- (C) Cette procédure peut être utilisée pour vérifier si un nombre est un nombre premier,
- (D) Cette procédure calcule le terme général de la suite $S(n) = S(n-1)+S(n-2)$
- (E) Cette procédure donne le même résultat que la fonction suivante écrite en C :

```
int fibo(int n)
{
    if(n < 2)
        return n;
    return fibo(n-1) + fibo(n-2);
}
```

Question 11

Soit la fonction suivante écrite en langage C :

```
int fonc(int tab[],int lg)
{
    int i,mini;
    int ln, z;

    i=0;
    while(i<(lg-1))
    {
        if (tab[i]>tab[i+1])
        {
            mini=tab[i+1];
            tab[i+1]=tab[i];
            tab[i]=mini;
            if (i==0) i++;
            else i--;
        }
        else i++;
    }
    ln = lg/2;
    z = lg%2; /* z ← le reste de la division par 2 */
    return (1/2 * (tab[ln-1+z] +tab[ln]));
}
```

L'appel de la fonction se fait avec le tableau $t=\{10, 30, 28, 64, 22, 16, 12\}$

- (A) L'appel $fonc(t, 6)$ modifie le contenu du tableau t comme suit :
 $t=\{10, 12, 16, 22, 28, 30, 64\}$
- (B) Cette fonction retourne la valeur 25 si elle est appelée comme suit : $fonc(t, 6)$
- (C) Cette fonction retourne la valeur 28 si elle est appelée comme suit : $fonc(t, 5)$
- (D) Cette fonction retourne la valeur 28 si elle est appelée comme suit : $fonc(t, 3)$
- (E) L'appel $fonc(t, 4)$ modifie le contenu du tableau t comme suit :
 $t=\{10, 28, 30, 64, 22, 16, 12\}$

Structure des machines

Question 12

- (A) La mémoire cache d'un ordinateur sert à rendre certaines données inaccessibles aux programmeurs,
- (B) La mémoire cache d'un ordinateur joue un rôle de tampon de données vis-à-vis de la mémoire centrale.
- (C) La mémoire cache d'un ordinateur est une extension de la mémoire centrale,
- (D) La mémoire cache d'un ordinateur permet d'accélérer l'exécution des programmes,
- (E) La mémoire cache d'un ordinateur contient à tout moment une copie d'une partie des données de la mémoire centrale,

Question 13

- (A) Durant l'exécution d'une tâche par un microprocesseur, l'ensemble des instructions qui composent le programme est chargé en mémoire centrale,
- (B) Les Unités Arithmétiques et Logiques (UAL) sont conçues pour effectuer des opérations en un seul cycle sur des opérandes se trouvant en mémoire centrale,
- (C) Les opérations effectuées par l'UAL d'un microprocesseur se font sur des opérandes chargés dans des registres,
- (D) Les Unités Arithmétiques et Logiques (UAL) sont conçues pour effectuer des opérations d'addition sur des données de types entier et réel,
- (E) L'UAL est un dispositif combinatoire qui effectue des opérations arithmétiques et logiques d'un microprocesseur.

Systemes d'exploitation

Question 14

- (A) Un système d'exploitation permet de garantir la sécurité et l'intégrité des données des utilisateurs,
- (B) Un système d'exploitation permet d'exploiter efficacement les périphériques,
- (C) Un système d'exploitation multi-tâches permet l'exécution de plusieurs processus en parallèle,
- (D) Un système d'exploitation temps réel traite instantanément les interruptions prioritaires,
- (E) Un système d'exploitation alloue la mémoire centrale entre les différents processus en exécution.

Question 15

- (A) La méthode d'accès aux fichiers par un système d'exploitation est de type séquentiel,
- (B) Lorsqu'un système d'exploitation effectue un accès à un dispositif de sauvegarde de type disque magnétique, c'est pour lire ou écrire un bloc de données correspondant à la taille d'un secteur,
- (C) Un système d'exploitation est un interpréteur de commandes,
- (D) UNIX peut exécuter un programme dont la taille en octets est supérieure à la taille de la mémoire physique disponible,
- (E) Un système d'exploitation temps réel traite tous les événements et garantit un temps maximum de réponse à un événement.

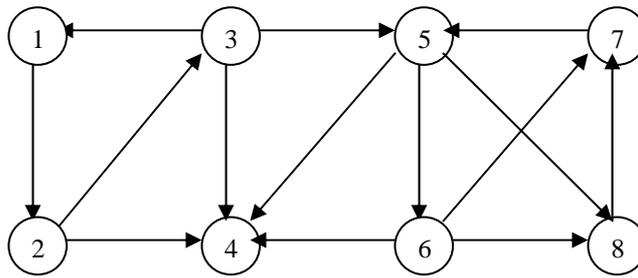
Graphes

Question 16

- (A) Un graphe est toujours planaire.
- (B) Un graphe ne contient pas de sommet isolé.
- (C) On appelle graphe complet un graphe dont tous les sommets sont adjacents.
- (D) Un graphe est connexe.
- (E) La somme des degrés d'un graphe non orienté est égale à deux fois le nombre d'arêtes du graphe.

Question 17

Soit le graphe :



- (A) Les sommets 1, 2 et 3 forment un circuit.
- (B) Le sommet 1 ne possède pas de prédécesseur.
- (C) Tous les sommets de ce graphe possèdent des successeurs.
- (D) Ce graphe est planaire.
- (E) Il existe un seul chemin entre les sommets 1 et 7.

Réseaux

Question 18

Dans le modèle en couche normalisé OSI, les fonctionnalités de la couche réseau sont prises en charge par le protocole IP. Ce protocole assure:

- (A) Le routage des paquets
- (B) Le contrôle et la reprise sur erreurs de transmission
- (C) Le multiplexage des paquets
- (D) La fragmentation et le réassemblage
- (E) La redirection des paquets

Question 19

Un navigateur utilisé pour afficher des pages Web doit pouvoir interpréter nativement du code:

- (A) HTML
- (B) HTTP
- (C) XML
- (D) Javascript
- (E) PHP