

BANQUE D'EPREUVE DUT-BTS

- SESSION 2002 -

**EPREUVE
D'INFORMATIQUE**

CODE EPREUVE : BE-INFO

CALCULATRICE INTERDITE

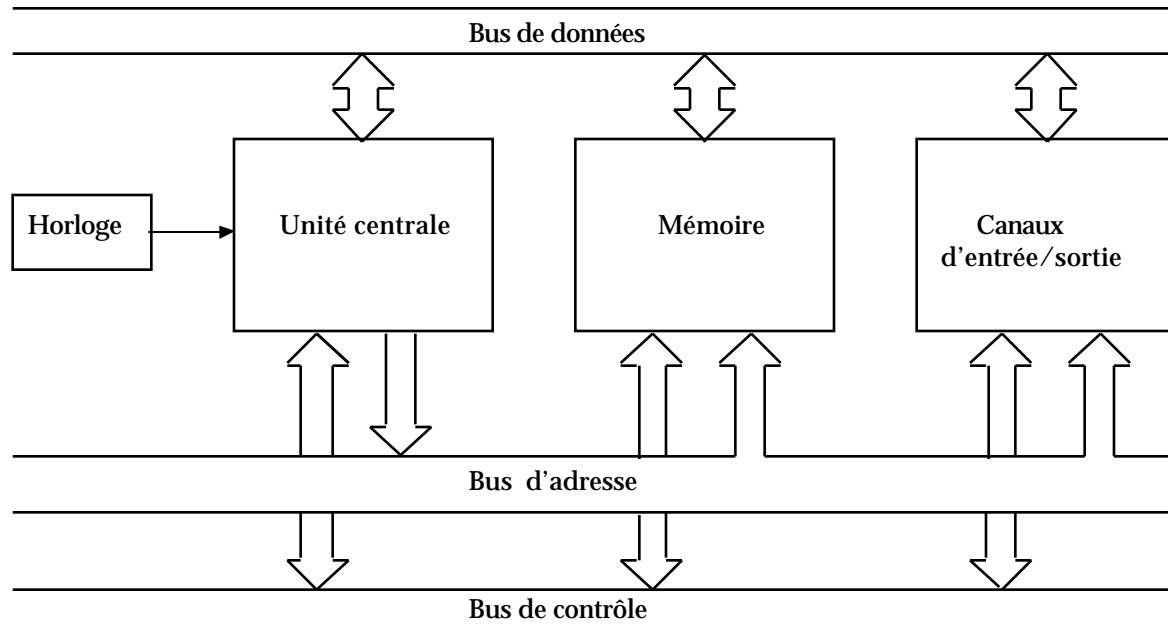
L'épreuve comporte 20 questions regroupées suivant les thèmes : structure des machines, algorithmique et programmation, systèmes d'exploitation, réseaux, graphes, électronique numérique.

DURÉE : 2H30

STRUCTURE DES MACHINES

Question 1

Soit l'organisation matérielle suivante:



- (A) Le bus d'adresse est mono-directionnel
- (B) Le bus de contrôle doit comporter un signal de lecture-écriture
- (C) Les mémoires de masse (disques) sont directement connectées à ces trois bus
- (D) La lecture du contenu d'une instruction en mémoire nécessite au moins 2 cycles d'horloge
- (E) Dans l'organisation ci-dessus, les bus sont multiplexés

Question 2

- (A) Dans une mémoire PROM, les bits sont représentés par des fusibles ou des diodes dont la jonction peut être claquée.
- (B) Une mémoire RAM dynamique est plus rapide qu'une mémoire RAM statique
- (C) Il faut un bus d'adresse plus large pour adresser une mémoire de 256 koctets en mots de 16 bits que la même mémoire en mots de 8 bits.
- (D) Une mémoire dynamique est une mémoire volatile
- (F) Une mémoire dynamique peut perdre son contenu si elle n'est pas rafraîchie

Question 3

- (A) Une unité centrale d'ordinateur contient une unité arithmétique et logique
- (B) Une file d'attente d'instructions s'appelle également structure « pipeline ».
- (C) Une unité centrale contient un compteur ordinal pour séquencer les instructions
- (D) La file d'attente permet de décoder les instructions avant de les exécuter.
- (E) Une unité centrale mémorise toutes les instructions d'un programme

Question 4

Soit l'opération suivante :

$$(MEMA) - (MEMB) - \$C \quad MEMR$$

MEMA, MEMB, MEMR sont homogènes à des adresses mémoires qui pointent sur des mots de deux octets (poids fort, poids faible dans l'ordre croissant des adresses).

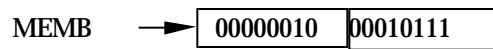
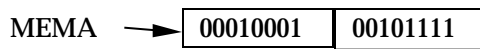
La notation (MEMA) signifie contenu du mot codé sur 16 bits d'adresse MEMA.

Le caractère « \$ » précédant une valeur numérique est réservé pour indiquer que la valeur est codée en base 16.

$$MEMA = \$1000$$

$$MEMB = \$1500$$

$$MEMR = \$2000$$



Après exécution de l'opération :

- (A) (MEMR) = -3852
- (B) (MEMR) = \$0F0C
- (C) (MEMR) = \$2000
- (D) (MEMR) = \$3852
- (E) (MEMR) = 3852

Question 5

Lors d'une interruption :

- (A) Le processeur sauvegarde dans la pile le registre d'état
- (B) Le processeur masque toutes les autres interruptions
- (C) Le processeur charge dans le compteur programme l'adresse d'une procédure d'exception
- (D) Le processeur interrompt l'instruction qui est en train de s'exécuter
- (E) Le processeur sauvegarde systématiquement tous les registres dans la pile

Question 6

Soit la soustraction suivante en notation binaire complément à deux (représenté en hexadécimal). Les opérandes et le résultat sont codés sur 16 bits :

$$\begin{array}{rcccc} & \mathbf{C} & \mathbf{5} & \mathbf{6} & \mathbf{B} \\ - & \mathbf{A} & \mathbf{7} & \mathbf{F} & \mathbf{D} \\ & \mathbf{C} & \mathbf{V} & \mathbf{N} & \mathbf{Z} \\ & \mathbf{-} & \mathbf{-} & \mathbf{-} & \mathbf{-} \\ & \mathbf{-} & \mathbf{-} & \mathbf{-} & \mathbf{-} \end{array}$$

- où C est l'indicateur de retenue
V est l'indicateur de débordement
N est l'indicateur de signe
Z est l'indicateur de résultat nul

L'état des indicateurs sera le suivant après l'opération:

- (A) C=0 ; V = 0 ; N = 0 ; Z = 0
- (B) C=1 ; V = 0 ; N = 1 ; Z = 0
- (C) C=1 ; V = 1 ; N = 0 ; Z = 0
- (D) C=0 ; V = 1 ; N = 0 ; Z = 0
- (E) C=0 ; V = 0 ; N = 1 ; Z = 0

ALGORITHMIQUE ET PROGRAMMATION

Question 7

Soit :

le tableau de caractères TAB : "programmation"

le tableau de caractères CLE : "xyz"

chaque tableau se termine par le caractère nul '\0'.

On considère l'algorithme suivant écrit en pseudo-langage :

I ← 0

L ← la longueur du tableau CLE

TANT QUE (TAB[I] différent de '\0') FAIRE

TAB[I] ← TAB[I] XOR CLE[I modulo L]

(où XOR est le OU exclusif bit à bit)

I ← I+1

FIN FAIRE

- (A) Il est possible de retrouver le tableau initial "programmation" à partir du résultat en appliquant une seconde fois l'algorithme sur le tableau TAB.
- (B) Il est possible de retrouver le tableau initial "programmation" en appliquant un OR (OU bit à bit) au lieu du XOR selon le même principe
- (C) Il faut appliquer le même algorithme mais avec le tableau CLE "zyx" pour retrouver le tableau initial
- (D) Pour que l'algorithme fonctionne, il faut que la longueur du tableau TAB soit un multiple de la longueur L de la CLE
- (E) Si le tableau CLE était plus long que le tableau TAB, l'algorithme fonctionnerait malgré tout

Question 8

Soit TAB un tableau de nombres entiers, de longueur N, indicé de 0 à N-1, sur lequel on applique l'algorithme suivant :

```
FAIRE
  PERMU ← 0
  POUR (I=0 à N-2) FAIRE
    SI (TAB[I] < TAB[I+1])
      Permuter TAB[I] et TAB[I+1]
      PERMU ← 1
  FIN SI
FIN POUR

(*)
TANT QUE (PERMU est égal à 1)
```

- (A) L'algorithme classe le tableau TAB dans l'ordre croissant
- (B) L'algorithme permute les valeurs du tableau initial de telle sorte que le tableau final est identique au tableau initial, indexé de N-1 à 0
- (C) En ajoutant $N \leftarrow N-1$ à la ligne (*), l'algorithme serait plus rapide, tout en assurant le même traitement du tableau
- (D) Le temps de calcul de l'algorithme dépend des valeurs contenues dans le tableau initial
- (E) Si $N > 1024$, l'algorithme ne fonctionne plus

Question 9

Soient U_0 , U_1 et U trois variables de type entier et N une constante entière ($N > 1$).

On applique l'algorithme suivant :

```
U0 ← 0, U1 ← 1
POUR (I= 2 à N) FAIRE
  U ← 2*U0+U1
  U0 ← U1
  U1 ← U
FIN POUR
```

- (A) En sortie, U contient la somme des $N-1$ premiers entiers
- (B) Si $N=3$, U vaut 3 en sortie de l'algorithme
- (C) L'algorithme calcule le terme général de la suite $U(n) = U(n-1) + 2*U(n-2)$
- (D) En initialisant U_0 et U_1 à 0, U sera nul quelque soit N
- (E) Si l'on avait initialisé U_0 à 1 et U_1 à 0, la valeur de U serait identique en sortie de l'algorithme

Question 10

Lors de l'exécution d'un programme développé en langage C, la mémoire allouée par le système d'exploitation est divisée en quatre zones (le code, les données statiques, la pile, le tas).

- (A) L'instruction « malloc » du langage C alloue de la mémoire dans la zone « pile »
- (B) Le tas est utilisé pour le stockage des variables de type entier uniquement
- (C) Lors d'un appel de fonction, les valeurs des arguments de la fonction sont recopiées dans la pile
- (D) Les variables locales aux blocs sont stockées dans la pile
- (E) Le code contient le programme exécutable et la mémoire nécessaire au stockage des variables globales

Question 11

Soit le programme suivant écrit en langage C :

```
#include <stdio.h>

void fonction(int *a, int *b)
{
    int temp;
    temp = *a;
    *a = *b;
    *b = temp;
}

void main()
{ int x,y;
  scanf("%d",&x);
  scanf("%d",&y);
  printf("x= %d y= %d\n",x,y);
  fonction(&x,&y);
  printf("x= %d y= %d\n",x,y);
}
```

- (A) Les variables « temp, x, y » sont des variables dites « automatiques » dont les contenus sont placés dans la zone mémoire appelée « tas », allouée pour le programme.
- (B) La variable « temp » dans « fonction » est inutile.
- (C) La fonction principale pourrait également s'écrire :

```
void main()
{ int *x, *y;
  x=(int *) malloc (sizeof(int)) ;
  y=(int *) malloc (sizeof(int)) ;
  scanf("%d",x);
  scanf("%d",y);
  printf("x= %d y= %d\n",*x, *y);
  fonction(x,y);
  printf("x= %d y= %d\n",*x, *y);
}
```

- (D) Dans la fonction principale ci-dessus les contenus des variables x et y ne sont plus placés dans la pile allouée pour le programme.
- (E) L'écriture suivante du programme est également correcte et exécuterait le même traitement

```
#include <stdio.h>
```

```
    temp = a;
    a = b;
    b = temp;
}
void main()
{ int x,y;
scanf("%d",&x);
scanf("%d",&y);
printf("x= %d y= %d\n",x,y);
fonction(x,y);
printf("x= %d y= %d\n",x,y);
}
```

Systèmes d'exploitation

Question 12

- (A) Le système d'exploitation UNIX est un système multi-processus
- (B) Le mécanisme de mémoire virtuelle d'UNIX permet d'utiliser des machines sans mémoire vive.
- (C) La totalité du noyau d'UNIX est écrit en langage C
- (D) Pour développer puis exécuter une application en langage C, il est indispensable de disposer d'un compilateur et d'un éditeur de liens.
- (E) Pour développer puis exécuter une application en langage JAVA, il est indispensable de disposer d'un compilateur et d'un éditeur de liens.

RÉSEAUX

Question 13

Dans un réseau local Ethernet 100 Base T :

- (A) Le délai de propagation (p) imposé, ne permet pas de mettre en cascade plus de deux commutateurs Ethernet
- (B) Il est possible de mettre en cascade 4 répéteurs en respectant $2p \leq 512 * \text{durée d'un bit}$
- (C) Le média utilisé peut être de la fibre optique
- (D) Il faut utiliser obligatoirement des commutateurs
- (E) Il n'est pas possible d'utiliser un routeur

Question 14

Le protocole ARP (Address Resolution Protocol) :

- (A) Emet parfois des trames Ethernet en diffusion
- (B) N'est jamais nécessaire au niveau d'un routeur assurant l'interconnexion d'un réseau local Ethernet au réseau Internet
- (C) N'est jamais utilisé sur un réseau local supportant une méthode d'accès (MAC) de type "Anneau à Jeton"
- (D) Assure une translation d'adresse entre le niveau 3 et le niveau 2 du modèle OSI
- (E) Est incompatible avec le modèle OSI

Question 15

Le protocole IP (Internet Protocol) :

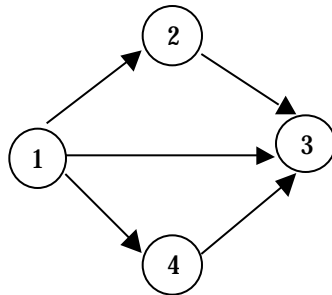
- (A) Effectue de la commutation de paquets en mode connecté
- (B) Assure le routage de datagrammes

- (D) Peut fonctionner dans un réseau étendu dans lequel le protocole ICMP (Internet Control Message Protocol) n'est pas implanté.
- (E) N'a pas été conçu pour supporter des flux de données isochrones

GRAPHES

Question 16

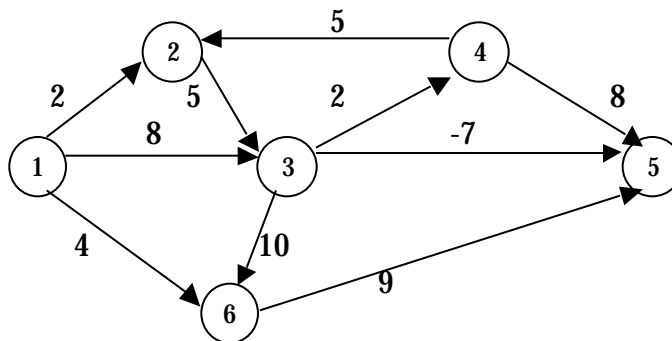
Soit G le graphe suivant :



- (A) Le graphe G est complet.
- (B) Le graphe G est symétrique.
- (C) G est un graphe sans cycle.
- (D) G est un graphe connexe.
- (E) Les sommets 2 et 4 sont adjacents.

Question 17

Soit le graphe valué suivant :



- (A) Le plus court chemin du sommet 1 au sommet 5 vaut 0.
- (B) Il n'existe pas de chemin de longueur maximale entre les sommets 1 et 5.
- (C) Ce graphe possède un seul cycle.
- (D) Il n'existe pas de chemin entre les sommets 4 et 6.
- (E) Le plus long chemin du sommet 6 au sommet 5 vaut 9.

ELECTRONIQUE NUMÉRIQUE

- représente la somme logique
- représente le produit logique

Question 18

(A) f est une fonction de 4 variables définie par :

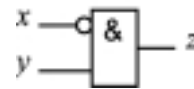
$$f = \bar{d}.c.\bar{a} \quad d.\bar{c}.a \quad c.b.a \quad \bar{d}.\bar{c}.\bar{b}.\bar{a} \quad d.c.\bar{b}.a \quad \bar{d}.\bar{c}.b$$

Si l'on dispose des variables a, b, c, d et également de leurs compléments $\bar{a}, \bar{b}, \bar{c}, \bar{d}$, il est possible de réaliser un circuit correspondant à cette fonction par le seul emploi de 3 portes NAND à 2 entrées et une porte NAND à 3 entrées.

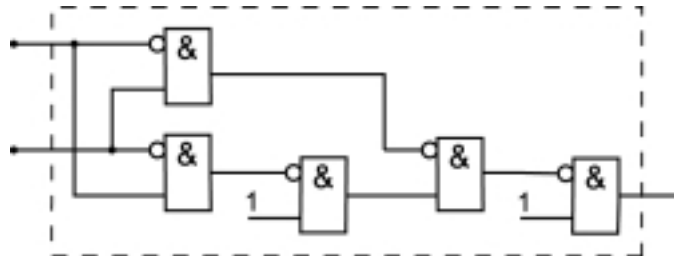
(B) Le circuit le plus rapide correspondant à cette fonction, réalisé en utilisant seulement des portes NAND dont chacune présente un temps de retard de propagation de 10 ns, présente lui-même un temps de retard de propagation de 40 ns.

(C) L'opérateur ci-contre réalise la fonction $z = \bar{x} \cdot y$

On ne peut pas réaliser n'importe quelle fonction combinatoire en utilisant seulement ce type d'opérateur.



(D) Le schéma suivant est celui d'un opérateur OU-EXCLUSIF :



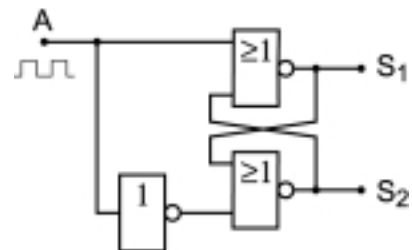
(E) La représentation numérique de l'état d'un bus de 16 lignes à un instant donné, exprimée en hexadécimal, est A5CE .

A cet instant, 8 lignes du bus sont à l'état bas, 8 à l'état haut.

Question 19

(A) Le circuit ci-contre, composé d'un inverseur et de 2 portes NOR, reçoit en A un signal carré.

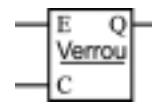
Si l'on considère que le temps de retard de propagation des 3 opérateurs est nul, le signal d'entrée A est reproduit en S_1 , son complémentaire en S_2 .



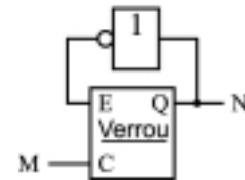
(B) Si l'on considère que le temps de retard des 3 opérateurs est de 10 ns, S_1 et S_2 ne sont jamais

(C) L'opérateur Verrou présente le fonctionnement suivant:

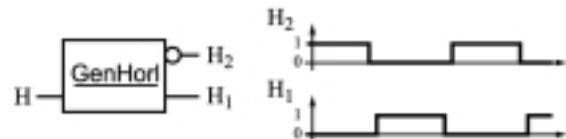
- si $C=0$, Q conserve l'état qu'avait E à l'instant qui a précédé le retour de C à 0,
- sinon ($C=1$), Q recopie les évolutions de E.



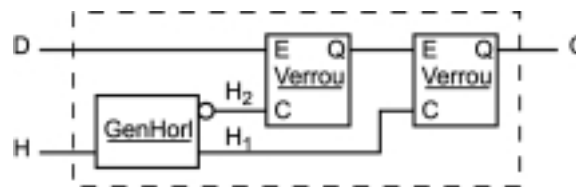
Dans l'association ci-contre de cet opérateur avec un inverseur, le signal appliqué en M et celui délivré en N sont dans un rapport de fréquences de 2.



(D) Le circuit GenHorl est un générateur d'horloge. Il reproduit en sortie H_1 le signal appliqué sur l'entrée H, en sortie H_2 le signal complémentaire, et ceci d'une façon telle que H_1 et H_2 ne sont jamais simultanément à 1.



L'association ci-contre du générateur d'horloge et de 2 opérateurs Verrou correspond à une bascule D déclenchée sur les fronts montants du signal d'horloge H.



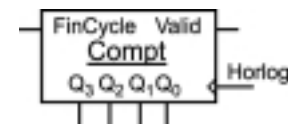
(E) La permutation de H_1 et H_2 conduit à un comportement sans rapport avec celui d'une bascule.

Question 20

Compt est un compteur hexadécimal synchrone (cycle de 16 états, code binaire naturel, Q_3 : bit de poids le plus fort, Q_0 : bit de poids le plus faible).

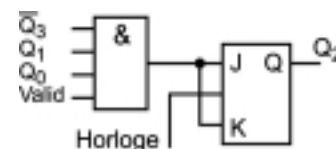
Valid est une entrée d'autorisation de comptage. Elle définit le comportement du compteur à chaque front montant d'horloge :

- si $Valid=0$, le compteur n'évolue pas,
- sinon ($Valid=1$) le compteur s'incrémente.



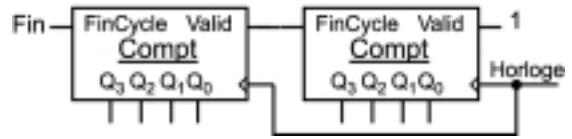
(A) Le compteur est réalisé à partir de 4 bascules JK. Le schéma ci-contre est un schéma partiel du compteur.

Il représente la bascule liée à la sortie Q_2 et son circuit de commande.

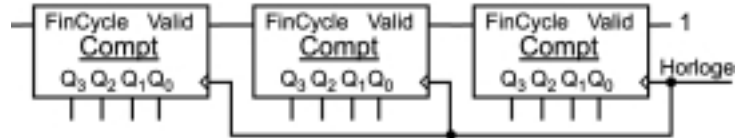


(B) Compt possède une sortie FinCycle qui correspond à $FinCycle = Q_3 Q_2 Q_1 Q_0$

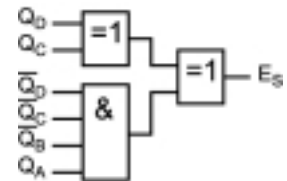
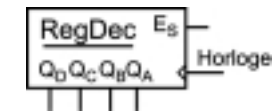
Dans cette association, la fréquence de Fin est 256 fois plus faible que la fréquence de Horloge.



(C) Cette association forme un compteur à 2048 états.



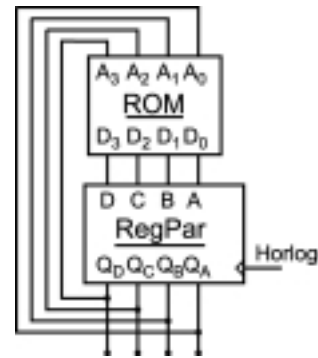
(D) RegDec est un registre à décalage 4 bits. E_S est son entrée série, $Q_D Q_C Q_B Q_A$ ses sorties parallèles. A chaque front montant de l'horloge, le contenu du registre est décalé d'une cellule, de A vers D. L'entrée série dépend des sorties à travers un circuit combinatoire constitué d'une porte ET et 2 portes OU-EXCLUSIF, selon le schéma ci-contre:



A partir de $Q_D Q_C Q_B Q_A = 1111$, l'ensemble décrit un cycle de 12 états.

(E) RegPar est un registre parallèle 4 bits. A chaque front montant de l'horloge, le mot appliqué en DCBA est recopié et mémorisé en $Q_D Q_C Q_B Q_A$.

Le contenu de la mémoire ROM est défini par la table ci-jointe. Adresses (A_3-A_0) et données (D_3-D_0) sont exprimées en hexadécimal, avec l'indice 3 associé au bit de plus fort poids, l'indice 0 au plus faible.



| Adresse | Donnée |
|---------|--------|
| 0 | 0 |
| 1 | 2 |
| 2 | 4 |
| 3 | 6 |
| 4 | 9 |
| 5 | B |
| 6 | D |
| 7 | F |
| 8 | 1 |
| 9 | 3 |
| A | 5 |
| B | 7 |
| C | 8 |
| D | A |
| E | C |
| F | E |

A partir de $Q_D Q_C Q_B Q_A = 1111$, l'ensemble décrit le même cycle que le système précédent.