DIRO
IFT 1215

EXAMEN INTRA

Max Mignotte

DIRO, Département d'Informatique et de Recherche Opérationnelle, local 2377.
Http : //www.iro.umontreal.ca/∼mignotte/ift1214/
E-mail : mignotte@iro.umontreal.ca

Date : *10/03/2006*

I	Conversion et Arithmétique en Base n (30 pts).
II	Représentation des Données Entières (15 pts).
III	Simplification des Fonctions Binaires (25 pts).
IV	Circuits Logiques (10 pts).
V	Circuits Séquentiels (26 pts).
Total	106 points.

Directives

- TOUTE DOCUMENTATION EST PERMISE
- TOUTE CALCULATRICE EST INTERDITE

○ Les réponses devront être clairement présentées et justifiées (elles peuvent être concises mais devront néanmoins contenir les résultats intermédiaires nécessaires permettant de montrer sans ambiguïté que vous êtes arrivés au résultat demandé).

○ Si vous ne comprenez pas une question, faites en une interprétation, et proposez une réponse.

I. Conversion et Arithmétique en Base n (30 pts)

1. Convertir 352_{10} en binaire (base 2). Utiliser le résultat de sa conversion en binaire pour trouver son équivalent octal (base 8) et hexadécimal (base 16). $<$**5 pts**$>$
2. Convertir 0.62_{10} en base 2, base 8 et base 16 en utilisant comme notation une barre au dessus d'un éventuel motif de plusieurs bits qui se répéterait ou 10 bits après la virgule si ce n'est pas le cas et au moins trois digits après la virgule en base 8 et 16. $<$**5 pts**$>$
3. En déduire la représentation de 20.62_{10} en base 2, base 8 et base 16. $<$**5 pts**$>$
4. Convertir le nombre $0.1\overline{101}_2$ (i.e., $0.1\,101\,101\,101\ldots$ avec le motif 101 se répétant à l'infinie) en une fraction décimale[1]. $<$**5 pts**$>$
5. Quel serait l'algorithme en base 10 qui permettrait de convertir un nombre entier exprimé en suite de caractères ASCII codé en Hexadécimal en sa valeur en base 10. (Nota : le caractère 0 (i.e., zéro) est représenté en codage ASCII par la représentation hexadécimal 30 et vous pouvez faire un dessin ou un graphique pour simplifier votre explication). $<$**5 pts**$>$
6. Représenter $0.1\overline{8}_{16}$ en une fraction décimale[1]. $<$**5 pts**$>$
   NOTA : Si vous n'y arrivez pas, donner une approximation de cette valeur en base 10 avec quatre chiffres significatifs après la virgule (pour la moitié des points de la question).

Réponse

$\boxed{\textbf{1.}}$

$352_{10} = 1\,0110\,0000_2 = 540_8 = 160_{16}$. $<$**5pts**$>$

$\boxed{\textbf{2.}}$

$0.62_{10} = 0.1\overline{00\;111\;101\;011\;100\;001\;010}_2 = 0.4753412_8 = 0.9EB85_{16}$. $<$**5 pts**$>$

$\boxed{\textbf{3.}}$

$20.62_{10} = 10\,100.100\;111\;100\;011_2 = 24.4753412_8 = 14.9EB85_{16}$. $<$**5 pts**$>$

$\boxed{\textbf{4.}}$

$\triangleright$ 1−**ère méthode** : En utilisant les suites géométriques :

$0.1\,\overline{101}_2$ se traduit en base 10 par la valeur :

$$2^{-1} + (2^{-2} + 2^{-4})\big(1 + 2^{-3} + 2^{-6}\ldots\big) = 2^{-1} + (2^{-2} + 2^{-4})\,\frac{1-(2^{-3})^{\infty}}{1-2^{-3}} = \frac{1}{2} + \frac{5}{16}\,\frac{8}{7} = \frac{96}{112} = \frac{6}{7}.$$

$\triangleright$ 2−**ième méthode** : En utilisant la méthode expliquée aussi en cours et utilisant les décalage de bits :

1. Soit $p = 0.1\,\overline{101}_2$. On a

$$\begin{array}{rcl} p = & 0. & 1\,\overline{101}_2 \\ 2\,p = & 1. & \overline{101}_2 \\ 16\,p = & 1101. & \overline{101}_2 \\ \hline \end{array}$$

$$16\,p - 12\,p = \quad 14\,p = \quad 1101_2 - 1_2 = 12_{10}$$

---

1. On pourra utiliser la méthode des suites géométriques ou toutes autres méthodes de son choix que l'on détaillera.

$$p = \tfrac{12}{14} = \tfrac{6}{7}.$$

$\triangleright$ 3−**ième méthode** : En notant que $2p = 1.\overline{101}_2 = 1_{10} + 0.\overline{101}_2$ et en trouvant par la méthode de son choix que $0.\overline{101}_2 = \frac{5}{7}$, on en déduit que $p = \frac{6}{7}$.
<5 pts>

$\boxed{\textbf{5.}}$

Pour un entier exprimé en caractères ASCII (codé en hexa) $\triangleright$ base 10

En partant des codes hexadécimaux regroupés par deux de gauche à droite ; Le nombre en base 10 est obtenu en concaténant un caractère sur deux ou en soustrayant 30 à chaque couple de digit de gauche à droite. **<5 pts>**

35 37 38 32

5 7 8 2

$\boxed{\textbf{6.}}$

Pour avoir la valeur exacte sous la forme d'une fraction décimale de $0.1\overline{8}_{16}$, on peux utiliser les deux méthodes mentionnées à l'exercice 4. La plus simple reste encore celle utilisant les décalage de digit.

$$\begin{array}{rcrcl}
 & x & = & 0. & 1\,\overline{8}_{16} \\
16_{10} & x & = & 1. & \overline{8}_{16} \\
256_{10} & x & = & 18. & \overline{8}_{16} \\
\hline
\end{array}$$

$$256\,x - 16\,x = \quad 240\,x = \quad 18_{16} - 1_{16} = 17_{16} = 23_{10}$$

$$x = \tfrac{23}{240} \approx 0.0958\overline{3}.$$

<5 pts>

Sinon, on a besoin de quatre chiffres après la virgule en base 16 pour arriver à la précision demandée en base 10. Dans ce cas, on a :

$$0.1\overline{8}_{16} = \frac{1}{16} + \frac{8}{16^2} + \frac{8}{16^3} + \frac{8}{16^4} \approx 0.9582519531.$$

II. Représentation des Entiers Signés (15 pts)

1. Représenter puis calculer $56 - 122$ sur 8 bits en complément à un (complément logique ou restreint) puis en complément à deux (complément arithmétique ou vrai) en indiquant à chaque fois lorsqu' apparaît un débordement (dépassement de capacité) ou une retenue ainsi que le résultat en base 10 obtenu. **<8 pts>**
2. À quoi correspond l'expression (i.e., le calcul) suivante en base 10 ? **<2 pts>**

   $$1101\,1101 - 1110\,0001$$

   exprimées en complément à deux et faites cette opération (calcul sur 8 bits en complément à deux) en indiquant lorsqu'apparaît un débordement ou une retenue. **<3 pts>**
   Quel est le nombre minimal de bits qui permettrait de faire ce calcul en complément à deux sans problème (sans dépassement de capacité) ? Justifier votre réponse. **<2 pts>**

Réponse

$\boxed{\textbf{1.}}$

Sur 8 bits, $56_{10} = 0011\,1000_2$ et $-122_{10} = -0111\,1010_2$

En complément à un (sur 8 bits) on a $56_{10} = 0011\,1000_2$ et $-122_{10} = 1000\,0101_2$ et l'opération suivante,

$$0011\,1000 + 1000\,0101_2 = 1011\,1101 = -0100\,0010_2. \quad < \textbf{3 pts} >$$

On obtient donc un chiffre négatif (premier des 8 bits a "1") et finalement une valeur égale à -66_{10}.
<**1 pt**>

En complément à deux (sur 8 bits) on a $56_{10} = 0011\,1000_2$ et $-122_{10} = -0111\,1010_2 = 1000\,0110$ et l'opération suivante,

$$0011\,1000_2 + 1000\,0110_2 = 1011\,1110 = -(0100\,0001 + 1_2) = -(0100\,0010_2). \quad < \textbf{3 pts} >$$

On obtient donc un chiffre négatif (premier des 8 bits a "1") et finalement une valeur égale à -66_{10}.
<**1 pt**>

$\boxed{\textbf{2.}}$

Équivalent à $1101\,1101 + 0001\,1111$, on additionne un nombre négatif ($-(\,0010\,0010_2 + 1_2 = -(0010\,0011_2) = -35_{10}$) avec un nombre positif ($0001\,1111_2 = 31_{10}$) ; c'est à dire l'opération en base 10 suivante $\triangleright$

$$-35_{10} + 31_{10}$$
<**2 pts**>

et on a en base 2, et complément à deux (sur 8 bit), on calcul donc l'opération suivante,

$$1101\,1101_2 + 0001\,1111_2 = 1111\,1100_2 = -(0011_2 + 1_2) = -(100_2) = -4_{10}.$$
<**3 pts**>

Pour pouvoir faire le calcul $-35_{10} + 31_{10}$ en complément à deux, il faut un nombre de bits nécessaire pour pouvoir représenter -35_{10}, $+31_{10}$ et le résultat, c'est à dire -4. Soit le premier entier n tel que $2^{n-1} - 1 > 35$, c'est à dire 7 bits. <**2 pts**>

III. Simplification des Fonctions Binaires (25 pts)

1. Sans utiliser la table de Karnaugh (donc seulement en utilisant les propriétés de l'algèbre de Boole), simplifier la fonction binaire suivante, <**5 pts**>

$$G_1 = \overline{a} + a\,b + \overline{c}\,\overline{b} + c\,\overline{b}$$

2. En utilisant la table de Karnaugh, simplifier le plus possible la fonction binaire suivante,

$$G_2 = a\,\overline{c}\,d + \overline{a}\,b\,c + \overline{a}\,b\,c\,\overline{d} + \overline{b}\,\overline{c}\,d + \overline{a}\,b\overline{c}\,d$$

   (a) sous la forme d'une **SOP**, <**5 pts**>
   (b) puis ensuite en utilisant un **POS** (*Produit de Somme*). <**5 pts**>

3. En utilisant la table de Karnaugh, donner sous forme de SOP (Somme de Produit) la forme (la plus) simplifiée de la fonction binaire suivante, <**5 pts**>

$$G_3 = a\,\overline{b}\,\overline{c}\,\overline{d} + a\,\overline{b}\,c\,\overline{d} + \overline{a}\,\overline{b}\,\overline{c}\,\overline{d} + a\,b\,c\,\overline{d} + \overline{a}\,b\,c + a\,\overline{b}\,c + \overline{b}\,c\,d$$

4. En utilisant la table de Karnaugh, simplifier de nouveau la fonction G_3 en utilisant en plus les états indifférents (i.e., "don't care") (soit G_4 cette fonction) ; <**5 pts**>

$$\overline{a}\,b\,\overline{c}\,\overline{d},\ a\,b\,c\,d$$

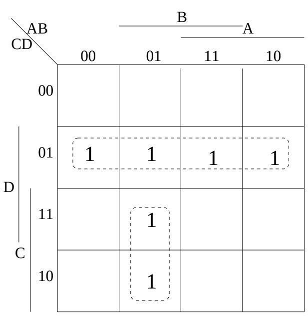
Réponse

$\boxed{\textbf{1.}}$

$$\begin{aligned} G_1 &= \overline{a} + a\,b + \overline{c}\,\overline{b} + c\,\overline{b} \\ &= \overline{a} + \overline{a}\,b + a\,b + \overline{c}\,\overline{b} + c\,\overline{b} \qquad \text{[Théorème de l'absorption]} \\ &= \overline{a} + b\,(\overline{a} + a) + \overline{b}(\overline{c} + c) \\ &= \overline{a} + b + \overline{b} \\ &= \overline{a} + 1 \\ &= 1. \qquad <\textbf{5 pts}> \end{aligned}$$

$\boxed{\textbf{2}}$

2.(a)



$$G_2 = \overline{c}\,d + \overline{a}\,b\,c \qquad <\textbf{5 pts}>$$

2.(b)

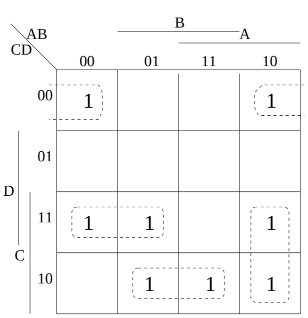
On fait des groupement de "0" et on obtient la forme SOP suivante,

$$\overline{G_2} = \overline{c}\,\overline{d} + ac + \overline{b}c$$

puis la forme POS pour G_2 ;

$$G_2 = (c + d).(\overline{a} + \overline{c}).(b + \overline{c}) \qquad <\textbf{5 pts}>$$

$\boxed{\textbf{3}}$



$$G_3 = \overline{b}\overline{c}\,\overline{d} + \overline{a}\,c\,d + b\,c\,\overline{d} + a\,\overline{b}\,c \qquad <\textbf{5 pts}>$$

---

0. Pour les questions III.2a,III.2b,III.3,III.4, la forme la plus simplifiée était exigée. Pour une expression juste (binairement parlant) mais non minimal, le barême est [Nb.Max Points - x] où x est le nombre de terme supplémentaire qui ne servent à rien pour une SOP et pour une POS inversement proportionnel au prorata du nombre de terme.

ou $\quad G_3 = \overline{b}\overline{c}\,\overline{d} + \overline{b}\,c\,d + \overline{a}\,b\,c + a\,c\,\overline{d}$

$\boxed{4}$

CD \ AB	00	01	11	10
00	1	d		1
01				
11	1	1	d	1
10		1	1	1

$G_4 = \overline{b}\overline{c}\,\overline{d} + c\,d + b\,c + a\,c \qquad$ <**5 pts**>

IV. Circuits Logiques (10 pts)

On désire construire un circuit logique qui a pour entré, quatre variables binaires A, B, C, D et une sortie V. On veut que $V = 1$ si le code binaire donné en entrée (avec A le LSB et D, le MSB) soit un multiple de trois. (Nota : 0 est un multiple de trois.)

1. Construire la table de vérité de V en fonction des quatre variables Booléennes A, B, C, D.
   <**3 pts**>
2. Implémenter l'expression brute (i.e., non simplifiée) de cette fonction V avec un $\text{MUX}_{4\times1}$ en utilisant quelques portes logiques (seulement des NON et NOR) et les variables D et C comme variables de contrôle du $\text{MUX}_{4\times1}$.
   <**7 pts**>

   Nota : Si vous n'y arrivez pas, alors implémenter l'expression de cette fonction V avec un $\text{MUX}_{8\times1}$ pour la moitié des points de cette question.

Réponse

$\boxed{\textbf{1.}}$

D	C	B	A	V
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

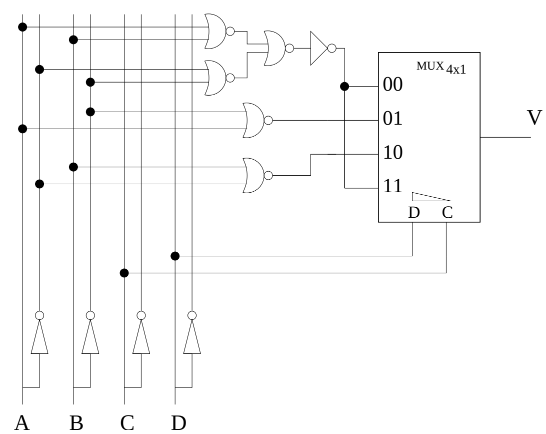
<**3 pts**>

$\boxed{\textbf{2.}}$

1. Si on utilise les variables D et C comme variable de contrôle du MUX, on devrait fournir à l'entrée du $\text{MUX}_{4\times1}$, pour successivement le code 00, 01, 10 et 11, les relations :
   (a) pour DC=00 $\triangleright$ $\overline{A \bigoplus B}$
   (b) pour DC=01 $\triangleright$ $B\,\overline{A}$
   (c) pour DC=10 $\triangleright$ $\overline{B}\,A$
   (d) pour DC=11 $\triangleright$ $\overline{A \bigoplus B}$

<4 pts>

et on obtiendrait le circuit suivant :

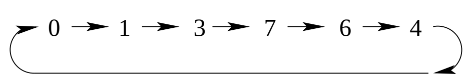


<3 pts>

Nota : Pour un $\text{Mux}_{8\times1}$, il suffisait de mettre en entrée respectivement $\overline{A}, A, 0, \overline{A}, A, 0, \overline{A}, A$ en gardant D, C, B comme entrée de contrôle.

V. Circuits Séquentiels (26 pts)

On voudrait réaliser le compteur synchrone (appelé compteur Johnson) parcourant le cycle principal suivant (en binaire),



i.e., un compteur affichant en binaire successivement 000, 001, 011, 111, 110, 100, 000, ...

On aimerait de plus, que ce circuit séquentiel ait aussi une entrée IN qui lorsque IN= 0, permet d'incrémenter ce compteur selon le cycle précédent (i.e., ...,000, 001, 011, 111, 110, 100, 000, ...) et lorsque IN= 1, décrémente ce compteur (i.e., ...100, 110, 111, 011, 001, 000, 100, ...).

1. Dire pourquoi il est 1) possible et 2) intéressant dans ce cas de figure de choisir comme code binaire, pour coder chacun des états de ce compteur, les mêmes codes binaires affichés en sortie des différents états de ce compteur ? **<2 pts>**
2. Dessiner le diagramme de transition d'états de ce système. **<6 pts>**

3. Donner ensuite la table d'état de ce système et établir la table de vérité de ce système. <**5 pts**>
4. Donner les équations logiques (simplifiées au maximum en utilisant les états indifférents si il y en a) des signaux de sortie et des états futurs de ce système. <**6 pts**>.
5. Donner le logigramme de ce système en utilisant des bascules D et les portes logiques que vous voudrez. <**5 pts**>
6. S'agit-il d'un circuit de Moore ou d'un circuit de Mealy ? Justifier votre réponse. <**2 pts**>

Réponse

$\boxed{\textbf{1.}}$

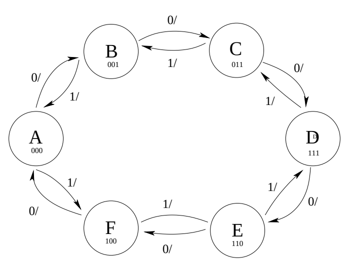
Le circuit séquentiel que l'on nous demande de faire aura indéniablement 6 états correspondant aux instants où celui-ci affichera successivement en sortie le code binaire 000, 001, 011, 111, 011, 100. Trois bits sont nécessaires pour donner un code binaire différent à ces six états. Il est donc avantageux de donner à ces états le même code binaire que celui affiché en sortie de ce compteur : D'une part, cela est possible car ce code est différent d'un état à l'autre, d'autre part cela permettra de simplifier la circuitrie de ce compteur séquentiel car les états futurs seront aussi les sorties de notre circuit séquentiel.

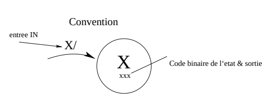
<**2 pts**>

$\boxed{\textbf{2.}}$

En utilisant sur les flèches de transition d'états, la convention suivante : Entrée IN/ et à l'intérieur des rond, l'état du système, son code binaire et la sortie du circuit séquentiel, on obtient le graphe suivant,

Graphe de Transition d'États : <**6 pts**>





$\boxed{\textbf{3.}}$

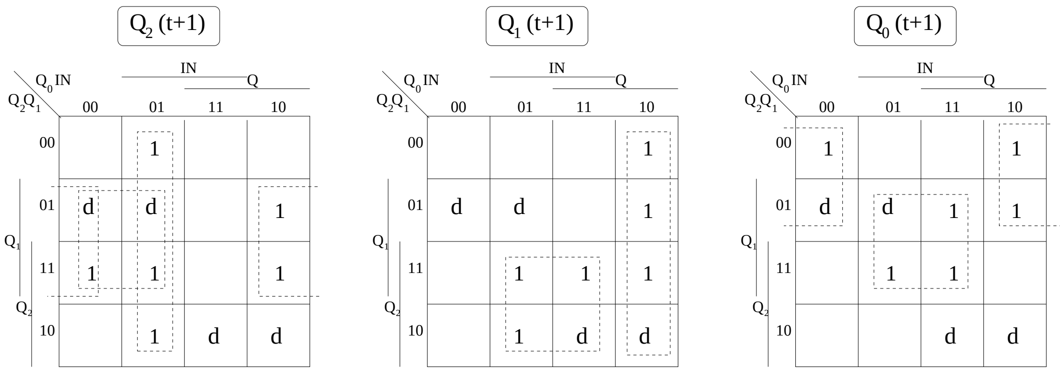
On peut donner directement la table de vérité de ce système en appelant $q_2\,q_1\,q_0$, les trois bascules du notre système séquentiel.

Table de Vérité : <**5 pts**>

$q_2(t)$ $q_1(t)$ $q_0(t)$	IN	$q_2(t+1)$ $q_1(t+1)$ $q_0(t+1)$
000	0	001
001	0	011
011	0	111
111	0	110
110	0	100
100	0	000
000	1	100
001	1	000
011	1	001
111	1	011
110	1	111
100	1	110

$\boxed{\textbf{4.}}$

On trouve, à partir de la table de vérité, et se servant des tableaux de Karnaugh et en considérant les états indifférents pour les combinaisons des bascules $q_2(t)q_1(t)q_0(t) = 010$ et $q_2(t)q_1(t)q_0(t) = 101$, les équations logiques suivantes, simplifiées au maximum,



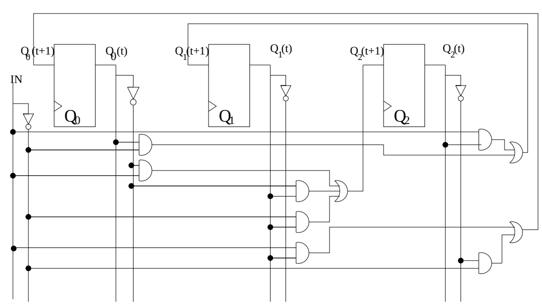
$$
\begin{aligned}
q_0(t+1) &= \mathrm{IN}\, q_1(t) + \overline{\mathrm{IN}}\, \overline{q}_2(t) \\
q_1(t+1) &= q_0(t)\, \overline{\mathrm{IN}} + q_2(t)\, \mathrm{IN} \\
q_2(t+1) &= \overline{q}_0(t)\, \mathrm{IN} + \overline{q}_0(t)\, q_1(t) + \overline{\mathrm{IN}}\, q_1(t)
\end{aligned}
$$

<6 pts>

Nota : Expression juste mais non simplifié : 3 pts seulement.

$\boxed{\textbf{5.}}$

On en déduit le schéma de notre circuit séquentiel :



<5 pts>

$\boxed{\textbf{6.}}$

Il s'agit d'une machine de **Mealy** car les sorties dépendent de l'état présent du système mais aussi de la valeur de l'entrée IN. Si il n'y avait pas eu l'entrée IN, cette machine aurait été une machine de **Moore** (les sorties n'auraient dépendu que de l'état du système).

<2 pts>