



DIRO
IFT 1215

EXAMEN INTRA

Max Mignotte

DIRO, Département d'Informatique et de Recherche Opérationnelle, local 2377

Http: [//www.iro.umontreal.ca/~mignotte/ift1215/](http://www.iro.umontreal.ca/~mignotte/ift1215/)

E-mail: mignotte@iro.umontreal.ca

Date: 22/02/2022

I	Conversion et Arithmétique en Base n (30 pts).
II	Représentation des Données Entières (12 pts).
III	Simplification des Fonctions Binaires (14 pts).
IV	Circuits Logiques (16 pts).
V	Circuits Séquentiels (44 pts).
Total	116 points.

Directives

- TOUTE DOCUMENTATION EST PERMISE
- TOUTE CALCULATRICE (OU ORDINATEUR) 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.
- Utiliser le recto et verso de votre cahier.
- À la fin, remettez votre énoncé d'examen dans votre cahier d'examen.

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

1. Convertir 111_{10} en binaire (base 2). Utiliser le résultat de sa conversion en binaire pour trouver son équivalent octal (base 8) et hexadécimal (base 16).
<6 pts>
2. Convertir $0.\overline{1}_{10}$ 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 à l'infini ou 12 bits après la virgule si ce n'est pas le cas et au moins trois digits après la virgule en base 8 et 16. Convertir aussi le nombre $0.\overline{1}_{10}$ en une fraction décimale.
<7 pts>
3. En déduire la représentation de $222.\overline{2}_{10}$ et $55.\overline{5}_{10}$ en base 2, base 8 et base 16.
<7 pts>
4. Convertir le nombre $0.\overline{000111}_2$ (i.e., $0.000111000111001001\dots$ avec le motif '000111' se répétant à l'infinie) en une fraction décimale.
<5 pts>
5. Représenter $0.\overline{1F}_{16}$ en une fraction décimale¹.
<5 pts>
NOTA: Si vous n'y arrivez pas, donner une approximation de cette valeur en base 10 avec deux chiffres significatifs après la virgule (pour la moitié des points de la question et en laissant une fraction ou somme de fractions).

Réponse

1.

$$111_{10} = 0110\ 1111_2 = 157_8 = 6F_{16}.$$

<5pts>

2.

$$0.\overline{1}_{10} = 0.000111\overline{000111}_2 = 0.070\overline{7}_8 = 0.\overline{1C7}_{16}.$$

En exploitant la méthode expliquée en cours et utilisant les décalages de digits, on a:

$$\begin{array}{r} p = 0.\ \overline{1}_{10} \\ 10p = 1.\ \overline{1}_{10} \\ \hline 10p - p = 9p = 1.\overline{10} - 0.\overline{10} = 1_{10} \\ p = \frac{1}{9} \end{array}$$

<6 pts>

3.

En utilisant les deux dernières questions, i.e., le fait que $111.\overline{1}_{10} = 01101111.00011\overline{000011}_2$ et qu'un décalage de 1 bit vers la gauche du . représente une division par deux et qu'un décalage de 1 bit vers la droite du . représente une multiplication par deux, on obtient donc facilement:

$$222.\overline{2}_{10} = 011011110.0011\overline{000011}_2 = 336.\overline{14}_8 = DE.\overline{30D}_{16}$$

et

$$55.\overline{5}_{10} = 0110\ 111.100011\overline{000011}_2 = 67.4\overline{30}_8 = 37.8\overline{C30}_{16}$$

<7 pts>

4.

▷ 1-**ère méthode**: En utilisant les suites géométriques:

$0.\overline{000111}_2$ se traduit en base 10 par la valeur:

$$(2^{-4} + 2^{-5} + 2^{-6})(1 + 2^{-6} + 2^{-12} \dots) = (7/64) \left(\frac{1-(2^{-6})^\infty}{1-2^{-6}} \right) = (7/64) \cdot \frac{64}{63} = \frac{1}{9}$$

▷ 2-**ième méthode**: En utilisant la méthode expliquée aussi en cours et utilisant les décalages de bits:

1. Soit $p = 0.\overline{000111}_2$. On a

$$\begin{array}{r} p = 0. \overline{000111}_2 \\ 64p = 111. \overline{000111}_2 \\ \hline \end{array}$$

$$64p - p = 63p = 111_2 - 0_2 = 7_{10} - 0_{10} = 7_{10}$$

$$p = \frac{7}{63} = \frac{1}{9}$$

<5 pts>

5.

Pour avoir la valeur exacte sous la forme d'une fraction décimale de $q = 0.\overline{1F}_{16}$, on peut utiliser les deux méthodes mentionnées à l'exercice 4. La plus simple reste encore celle utilisant les décalages de digit.

$$\begin{array}{r} q = 0. \overline{1F}_{16} \\ 256_{10} \times q = 1F. \overline{1F}_{16} \\ \hline \end{array}$$

$$256q - q = 255q = 1F_{16} = 31_{16}$$

$$q = \frac{31}{255} (\approx 0.121568627)$$

ou tout simplement:

$$0.\overline{1F}_{16} = \frac{1}{16} + (15 \times \frac{1}{16^2}) \approx \frac{31}{256} (\approx 0.12109375)$$

qui nous donne une approximation à $1/16^3$ près *i.e.*, avec moins de $1/1000$ d'erreur ou trois chiffres après la virgule.

<5 pts>

II. Représentation des Entiers Signés (12 pts)

1. Représenter puis calculer $-52 + 123$ 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 ?

$$-1111\ 0100 - 0111\ 1010$$

exprimée 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.

<4 pts>

Réponse

1.

Sur 8 bits, $123_{10} = 0111\ 1011_2$ et $-52_{10} = -0011\ 0100_2$

En complément à un (sur 8 bits) on a $123_{10} = 0111\ 1011_2$ et $-52_{10} = 1100\ 1011_2$ et l'opération suivante:

$$1100\ 1011_2 + 0111\ 1011_2 = 0100\ 0110_2 + 1_2 = 0100\ 0111_2$$

On obtient une retenue (+1) que l'on a ajouté et finalement un résultat positif (premier des 8 bits a "0") et finalement une valeur égale à 71_{10} .

<4 pts>

En complément à deux (sur 8 bits) on a $123_{10} = 0111\ 1011_2$ et $-52_{10} = 1100\ 1100_2$ et l'opération suivante:

$$1100\ 1100_2 + 0111\ 1011_2 = 0100\ 0111_2 = 0100\ 0111_2$$

Dans ce cas, la somme est la même ! mais on a ignoré la retenue (*i.e.*, que l'on n'ajoute pas cette retenue au résultat contrairement au complément à un). On obtient donc aussi un résultat positif (premier des 8 bits a "0") et finalement une valeur égale à 71_{10} .

<4 pts>

2.

$-1111\ 0100$ est égale à $-(-12_{10}) = 12_{10}$.

En effet, pour exprimer 12_{10} en complément à deux, on a fait l'opération: $12_{10} = 0000\ 1100_2$. et ensuite [en complément à deux], $-12_{10} = 1111\ 0011_2 + 1_2 = 1111\ 0100_2$.

$0111\ 1010_2$ est égale à 122_{10}

Donc on a l'opération $12_{10} - 122_{10} (= -110_{10})$.

En base 2, et complément à deux (sur 8 bits), on calcule donc l'opération suivante:

$$-1111\ 0100_2 - 0111\ 1010_2 = 0000\ 1100_2 + 1000\ 0110_2 = 1001\ 0010_2$$

Il y a une retenue, non prise en compte, aucun débordement de capacité et un résultat négatif qui a une magnitude de $0110\ 1110_2 = -110_{10}$ (car $110_{10} = 110\ 1110_2$ et en C2 sur 8 bits $-110_{10} = -0110\ 1110_2 = 1001\ 0001_2 + 1_2 = 1001\ 0010_2$).

<4 pts>

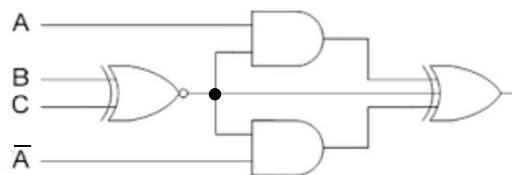
III. Simplification des Fonctions Binaires (13 pts)

1. En utilisant exclusivement l'algèbre de Boole, démontrez que:

<7 pts>

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

2. En utilisant ce que vous voulez, montrez à quoi est égale la sortie de ce circuit:



<7 pts>

Réponse

1.

$$\begin{aligned}
 a\bar{c} + \bar{b}c + \bar{a}b &= a\bar{c}(b + \bar{b}) + \bar{b}c(a + \bar{a}) + \bar{a}b(c + \bar{c}) \\
 &= a\bar{c}b + a\bar{c}\bar{b} + a\bar{b}c + \bar{a}\bar{b}c + \bar{a}b\bar{c} + \bar{a}bc \\
 &= (a\bar{c}b + \bar{a}b\bar{c}) + (a\bar{c}\bar{b} + a\bar{b}c) + (\bar{a}\bar{b}c + \bar{a}bc) \\
 &= b\bar{c} + a\bar{b} + \bar{a}c
 \end{aligned}$$

<7 pts>

2.

Soit X la sortie du NON-OU EXCLUSIVE(B, C), on a:

$$\begin{aligned}
 S &= AX \oplus X \oplus \bar{A}X = (AX\bar{X} + \bar{A}\bar{X}X) \oplus \bar{A}X \\
 &= [(\bar{A} + \bar{X})X] \oplus \bar{A}X \\
 &= \bar{A}X\bar{A}\bar{X} + \bar{A}\bar{X}\bar{A}X \\
 &= \bar{A}X\bar{A}\bar{X} \\
 &= \bar{A}X(A + \bar{X}) \\
 &= 0
 \end{aligned}$$

<7 pts>

IV. Circuits Logiques (16 pts)

Faire un circuit logique combinatoire qui prend en entrée 2 nombres A et B codés sur 2 bits ($A_1 A_0$ et $B_1 B_0$, avec A_0 et B_0 le LSB) et qui donne en sortie le produit Z de ces nombres. Z sera codé sur 4 bits ($Z_3 Z_2 Z_1 Z_0$ avec Z_0 le LSB).

1. Construire la table de vérité.

<8 pts>

2. Donner les équations logiques (simplifiées au maximum) des signaux de sortie de ce circuit.

<8 pts>

Réponse

1.

	A1	A0	B1	B0	Z3	Z2	Z1	Z0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0
2	0	0	1	0	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	1	0	0	0	0	0	0
5	0	1	0	1	0	0	0	1
6	0	1	1	0	0	0	1	0
7	0	1	1	1	0	0	1	1
8	1	0	0	0	0	0	0	0
9	1	0	0	1	0	0	1	0
10	1	0	1	0	0	1	0	0
11	1	0	1	1	0	1	1	0
12	1	1	0	0	0	0	0	0
13	1	1	0	1	0	0	1	1
14	1	1	1	0	0	1	1	0
15	1	1	1	1	1	0	0	1

Table de vérité

<8 pts>

2.

Pour $Z3$, aucunement besoin de table de Karnaugh.

On trouve: $Z3 = A1 A0 B1 B0$

Les cases vides sont égales à zéro.

Pour $Z2$: $Z2 = A1 \overline{A0} B1 + A1 B1 \overline{B0}$

		A1 A0			
		00	01	11	10
B1 B0	00				
	01				
	11				1
	10			1	1

Pour $Z1$: $Z1 = \overline{A1} A0 B1 + A1 B1 \overline{B0} + A1 \overline{B1} B0 + A1 \overline{A0} B0$

		A1 A0			
		00	01	11	10
B1 B0	00				
	01			1	1
	11		1		1
	10		1	1	

Pour $Z0$: $Z0 = A0 B0$

		A1 A0			
		00	01	11	10
B1 B0	00				
	01		1	1	
	11		1	1	
	10				

<8 pts>

V. Circuits Séquentiels (44 pts)

On va considérer dans cet exercice, la conception d'un système (simplifié) automatique de machine qui permet à l'utilisateur de se faire rembourser la consigne de chaque bouteille (vide) de vin en verre (qui sera ensuite recyclée).

La machine en question vous donne 25¢ pour chaque bouteille en verre consignée. Pour que l'utilisateur reçoive son argent, il existe trois possibilités. Soit l'utilisateur presse le bouton **Done** après chaque retour de bouteille et la

machine lui donnera automatiquement une pièce de 25¢. Soit l'utilisateur donne successivement quatre bouteilles, l'une après l'autre, et la machine lui donnera automatiquement, à la fin de cette opération, sans avoir à presser de bouton (*i.e.* sans avoir à appuyer sur D) une pièce de 1 \$. Soit l'utilisateur donne deux ou trois bouteilles, l'une après l'autre et appuie ensuite sur le bouton **Done**, à la fin de cette opération et la machine lui donnera automatiquement le nombre correct de pièces de 25¢ (*i.e.*, deux ou trois pièces de 25¢ dans notre exemple). Si le bouton **Done** est pressée sans que l'utilisateur ait rendu de bouteille en verre, alors aucune pièce de monnaie n'est donnée.

Dans ce système séquentiel, les sorties seront donc les trois variables $S_3 S_2 S_1$ indiquant:

- ▷ Si $S_3 S_2 S_1 = 000$, aucune pièce de 25¢ n'est rendue.
- ▷ Si $S_3 S_2 S_1 = 001$, une pièce de 25¢ est rendue.
- ▷ Si $S_3 S_2 S_1 = 010$, deux pièces de 25¢ sont rendues.
- ▷ Si $S_3 S_2 S_1 = 011$, trois pièces de 25¢ sont rendues.
- ▷ Si $S_3 S_2 S_1 = 100$, une pièces de 1\$ est rendue.

L'entrée de ce système est le bouton D (**Done**) ($D = 1$ pour pressée et $D = 0$ pour non pressée). Une autre entrée est C (pour consigne). Un capteur détecte le rendu d'une bouteille par $C = 1$. Lorsqu'aucune bouteille n'est détectée ou rendue: $C = 0$. Il n'y a pas de RESET sur cette machine. Le fait d'appuyer sur le bouton **Done** permet de rendre la monnaie et de remettre le système dans l'état initial d'attente de réception de bouteille. Dans cette machine, D et C ne peuvent être simultanément à un.

1. Dessiner le diagramme de transition d'états de ce système automatique (machine de Mealy).

<16 pts>

Nota: puisque la machine séquentielle est une machine de Mealy, On utilisera sur les flèches de transition d'états, la convention suivante $C D/S_3 S_2 S_1$ [*i.e.*, $C D/S_3 S_2 S_1$ du style 00/000 indiquant, dans cet exemple, que l'on n'a pas détecté de bouteille ($C = 0$) et que l'utilisateur n'a pas appuyé sur D ($D = 0$) et qu'en sortie aucune pièce de 25¢ ou de 1 \$ n'est rendue]. Aide: c'est une machine séquentielle avec 4 états.

2. Donner ensuite la table d'état de ce système.

<7 pts>

3. Établir la table de vérité de ce système.

<7 pts>

4. Donner les équations logiques (**simplifiées au maximum**) des signaux de sortie et des états futurs de ce système.

<7 pts>

5. Donner le logigramme de ce système en utilisant des bascules D.

<7 pts>

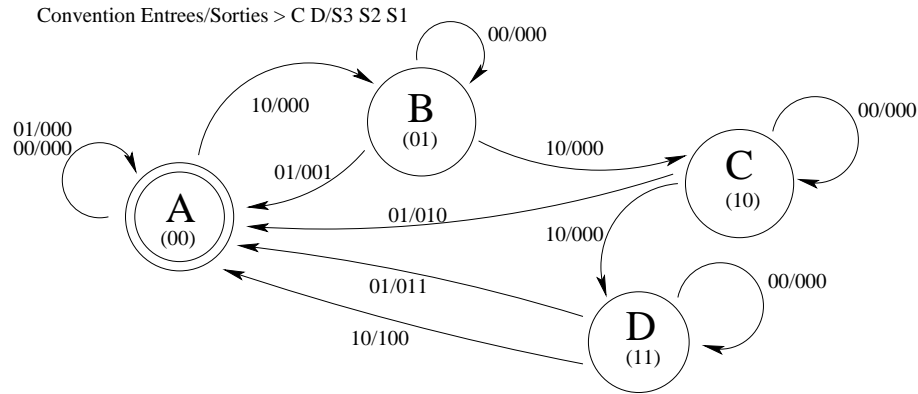
Réponse

1.

Graphe de Transition d'États :

Dans ce diagramme d'états (C pour consigne et D pour bouton Done) et:

- ▷ A est l'état initial
- ▷ B, C, D sont les états où la machine a reçue une, deux ou trois bouteilles et l'utilisateur n'a pas encore appuyé sur le bouton D.
- ▷ Lorsqu'on est arrivé à l'état D, on peut soit appuyer sur D et recevoir 3 pièces de 25¢ ou soit donner une autre bouteille et la machine donnera automatiquement (sans que l'utilisateur doive appuyer sur D) une pièce de 1\$ puis, dans ces deux cas, se remettre en position initiale (état A).



<16 pts>

2.

Table d'États :

Etat present \ Input	C D		
	00	01	10
A : 00	A:00/000	A:00/000	B:01/000
B : 01	B:01/000	A:00/001	C:10/000
C : 10	C:10/000	A:00/010	D:11/000
D : 11	D:11/000	A:00/011	A:00/100

<7 pts>

3.

Table de Vérité :

C	D	$q_1(t)$	$q_0(t)$	S3	S2	S1	$q_1(t+1)$	$q_0(t+1)$
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	1	0
0	0	1	1	0	0	0	1	1
0	1	0	0	0	0	0	0	0
0	1	0	1	0	0	1	0	0
0	1	1	0	0	1	0	0	0
0	1	1	1	0	1	1	0	0
1	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	1	0
1	0	1	0	0	0	0	1	1
1	0	1	1	1	0	0	0	0

avec des "don't care" pour la configuration: $C = 1, D = 1$.

<7 pts>

4.

Équations Logiques :

On trouve, à partir de la table de vérité, et se servant (ou non) des tableaux de Karnaugh, les équations logiques suivantes, simplifiées au maximum:

$$\begin{aligned}
 S3 &= Q_1(t) Q_0(t) C \overline{D} \\
 S2 &= Q_1(t) \overline{C} D \\
 S1 &= Q_0(t) \overline{C} D
 \end{aligned}$$

ou en prenant en compte les “don’t care”:

$$\begin{aligned}
 S3 &= Q_1(t) Q_0(t) C \\
 S2 &= Q_1(t) D \\
 S1 &= Q_0(t) D
 \end{aligned}$$

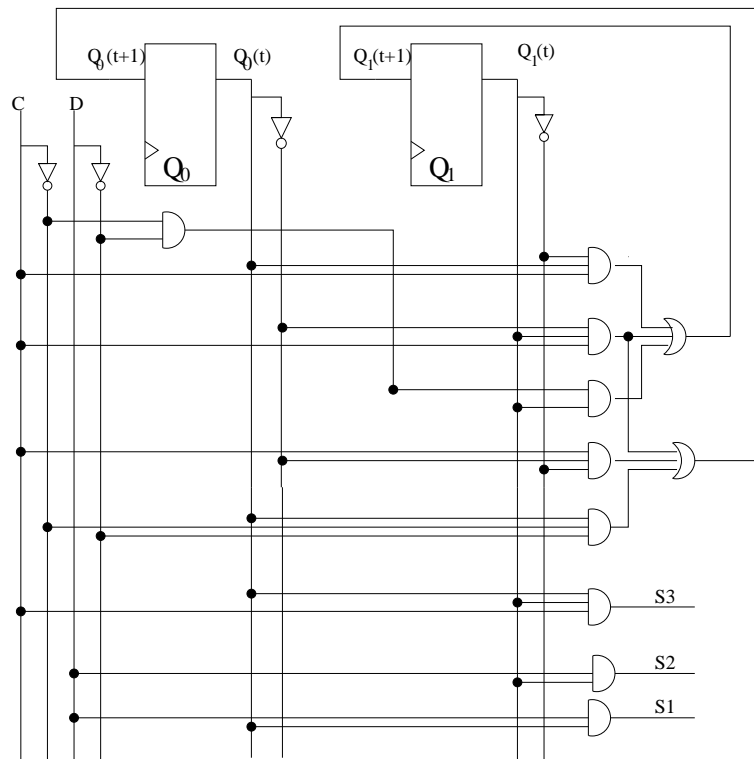
En utilisant les tableaux de Karnaugh avec des états indifférents pour la configuration $CD = 11$.

$$\begin{aligned}
 Q_1(t+1) &= \overline{Q_1(t)} Q_0(t) C + Q_1(t) \overline{Q_0(t)} C + Q_1(t) \overline{C} D \\
 Q_0(t+1) &= \overline{Q_1(t)} \overline{Q_0(t)} C + Q_1(t) \overline{Q_0(t)} C + Q_0(t) \overline{C} D
 \end{aligned}$$

<7 pts>

5.

On en déduit le schéma de notre circuit séquentiel:



<7 pts>