

IFT 1215 — Examen final — Automne 2016

Introduction aux systèmes informatiques

Professeur : Michel Boyer

Date : Le vendredi 9 décembre 2016, 13 : 30 – 16 : 20, local N-615, Pavillon Roger-Gaudry.

Directives :

- *Documentation permise : une feuille manuscrite format A4 ou lettre (8.5 × 11).*
- *Aucun appareil électronique autorisé (calculatrice, téléphone, iPod, etc).*
- *Répondre sur le questionnaire dans l'espace disponible. Utiliser le verso si nécessaire.*
- *Attention, les questions ne sont pas nécessairement en ordre de difficulté.*
- *Total : $100 \times 0.4 = 40\%$ de la note finale.*

Nom _____

Prénom _____

Matricule _____

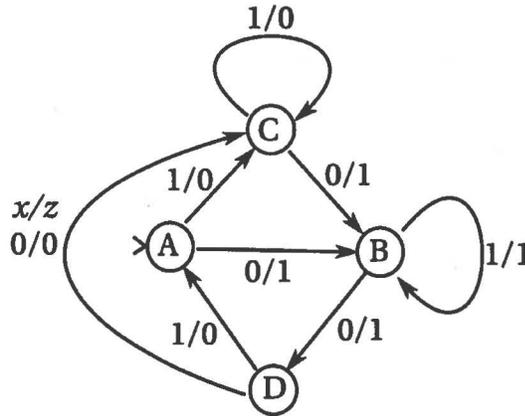
Question 1	/20
Question 2	/25
Question 3	/15
Question 4	/10
Question 5	/20
Question 6	/10
Total	/100

Signature : _____

1. Circuits séquentiels (20 points)

- a) (5 points) Donnez la table de vérité d'un circuit combinatoire qui permet à partir d'un compteur modulo 4, c'est-à-dire d'un circuit dont les sorties bouclent à chaque top d'horloge sur les valeurs 00, 01, 10 et 11, de construire un compteur dont les sorties hexadécimales bouclent sur les valeurs BC, CD, DE et EF.
- b) (5 points) Donnez un *diagramme d'états (automate)* qui prend en entrée, séquentiellement, un à un, des bits, et met 1 en sortie lorsque les trois derniers bits lus sont 010 ou 101 et qui met 0 en sortie sinon. Il faut bien sûr mettre 0 en sortie quand moins de deux bits ont été lus.

c) (10 points) Le diagramme suivant décrit un automate avec les quatre états A, B, C et D, qui prend en entrée un bit x et retourne en sortie le bit z .



Donnez la suite des sorties et des états de cet automate si l'état initial est A et si la suite des entrées lues séquentiellement, un bit lu à la fois, du temps 1 au temps 5, est

temps	1	2	3	4	5
Entrée	1	1	0	1	0
Sortie					
Etat d'arrivée					

Complétez la table d'états (*table de transition*) de cet automate en spécifiant dans chaque case les valeurs s'/z où s' est le nouvel état et z la sortie :

$s \backslash x$	$x = 0$	$x = 1$
A		
B		
C		
D		

2. La machine LMC (25 points)

La machine LMC a ses instructions et valeurs numériques (en complément à 10) codées avec trois décimales, ses adresses sont de deux décimales, de 00 à 99. Si on désigne par $M[xy]$ le contenu de la cellule mémoire d'adresse xy (xy de 00 à 99), le mnémonique de l'assembleur, le code opération et l'action effectuée par chaque instruction est alors comme suit :

Mnémonique	Code	Action
ADD	1xy	$M[xy]$ est additionné au contenu de A
SUB	2xy	$M[xy]$ est soustrait du contenu de A
STO	3xy	le contenu de A est copié dans $M[xy]$
LDA	5xy	$M[xy]$ est copié dans A
BR	6xy	Brancher sans condition à l'adresse xy
BRP	8xy	Brancher à l'adresse xy si $A \geq 0$
BRZ	7xy	Brancher à l'adresse xy si A est égal à 0
IN	901	L'entrée est mise dans A
OUT	902	Le contenu de A est copié en sortie.
HLT	000	L'exécution s'arrête
DAT		Met une valeur numérique à l'adresse courante.

L'exécution utilise les registres A (accumulateur), MAR (memory address register), MDR (memory data register), PC (compteur ordinal) et IR (registre instruction); on note $IR[adr]$ la partie adresse (contenant la valeur xy) du registre instruction. Attention, tous les programmes débutent à l'adresse 00.

- a) (5 points) Donnez les transferts de registres pour le fetch et l'exécution de l'instruction BR de la LMC (telle que décrite ci-dessus).

- b) (4 points) Traduisez dans le langage machine de la LMC le code assembleur suivant en indiquant bien en première colonne les adresses où sont mises les instructions (les programmes débutent toujours à l'adresse 00); les entrées de la colonne "Code LMC" sont chacune trois chiffres décimaux.

		Adresse	Code LMC
A	IN	00	901
	BRP B		
	HLT		
B	OUT		
	BR A		

- c) (4 pts) En supposant que les instructions et données suivantes se trouvent en mémoire de la LMC aux adresses indiquées et que le registre A contient 20

```
...  
20  ADD 50  
21  SUB 51  
...  
50  124  
51  006
```

Quel est le contenu de IR, PC, MAR, MDR et A immédiatement après que l'instruction à l'adresse 20 ait terminé son exécution ?

IR	PC	MAR	MDR	A

- d) (5 points) Quelle valeur faut-il mettre en entrée pour que le programme suivant mette 25 en sortie ? Justifiez votre réponse.

```
IN  
STO A  
A  DAT 000  
OUT  
HLT  
B  DAT 025
```

- e) (7 points) Écrire en assembleur LMC un programme qui lit en entrée des entiers positifs ou nul, les additionne, et retourne en sortie la somme et s'arrête dès qu'une entrée négative est tapée. Si la première entrée lue est négative, le programme doit retourner 0 et s'arrêter.

3. Mémoire cache (15 points).

Dans cette question $K = 2^{10}$, $M = 2^{20}$, $G = 2^{30}$. Supposons une mémoire adressable de 256M octets (les adresses ont 28 bits) et une mémoire cache (cache direct) de 64K octets au total, et dont les lignes sont de 16 octets.

- a) Combien la mémoire cache contient-elle de lignes et combien de bits ont respectivement l'index et l'étiquette (tag) ?

- b) Soit l'adresse mémoire hexadécimale, 9ABCDEF, quels sont (en notation hexadécimale) le déplacement (offset), l'index et l'étiquette de cette adresse ?

- c) Si l'accès à l'adresse 9ABCDEF ne cause pas de défaut de cache, donnez la liste explicite de toutes les adresses mémoire dont on peut être certain que leur accès ne causera pas de défaut de cache.
- d) Si ce même ordinateur a une mémoire dont le temps de latence, i.e. le temps pour accéder au premier mot, est de 50 ns ; le temps de transfert des mots suivants est de 10 ns, quel est le temps de chargement d'une ligne de cache ?
- e) Sous les mêmes hypothèses quel est le temps moyen d'accès au cache si le taux de succès (hit rate) est de 85%, si le temps d'accès au cache est de 4 ns et si lors d'un défaut (miss) de cache, il suffit de d'abord charger la ligne puis ensuite d'y accéder. Écrivez votre réponse sous forme de formule puisque vous n'avez pas la calculatrice.

4. Adressage et format des instructions (10 points)

- a) Un ordinateur a une instruction ADD de 16 bits de format

op code	R _{dst}	R _{src}
---------	------------------	------------------

; l'op-code occupe les bits 15 à 8, R_{dst} occupe les bits 7 à 4 et R_{src} occupe les bits 3 à 0; l'instruction ADD R1 R2 a pour effet d'additionner le contenu de R1 à celui de R2. Combien cette machine peut-elle avoir de registres R_i ?
- b) Sur une machine, l'instruction LOAD R4, 6(R3) dont l'effet est de faire la copie de mémoire à registre $R[4] \leftarrow M[6+R[3]]$ est codée dans le format

Opcod	R _{dst}	depl	R _{src}
-------	------------------	------	------------------

 sur 32 bits (ici R_{src} est R3, depl est 6 et R_{dst} est R4. Si la machine a 16 registres généraux R_n et si l'opcod a 7 bits, quelles sont les valeurs possibles pour le déplacement signé en représentation complément à deux ? Faites clairement mais succinctement ressortir les étapes de votre raisonnement (ne serait-ce que pour avoir des points partiels).

5. Infixe et postfixe (20 points).

a) Soit l'expression $((x - y) * (x + z)) - ((x + y) * y - y * (x - z))$.

(i) (5 points) Donnez sa représentation sous forme arborescente.

(ii) (5 points) En déduire sa représentation en notation postfixe.

b) (i) (3 points) Traduisez en postfixe l'expression $F(z) + G(y, x)$

(ii) (7 points) Écrivez le code `.asm` pour la WKC correspondant à la fonction définie par

```
function H(x,y,z){  
    return F(z) + G(y,x)  
}
```

où F et G sont des fonctions supposées déjà définies (toutes les fonctions ont des arguments entiers et retournent une valeur entière ; note : les instructions de la WKC sont en appendice).

6. La machine WKC, implantation et exécution (10 points).

a) Un programme pour la WKC contient les instructions suivantes aux adresses 32, 33, 34

```
PUSH 3(FP)
PUSH F
CALL 2
```

Si FP contient 1031 avant l'appel de F et si les valeurs aux adresses 1031 à 1035 sont 1, 2, 3, 4, 5, quelle sera la sortie donnée par F si le code de F est

```
F: PUSH 1(FP)
   PROUT
   PUSH 2(FP)
   PROUT
   RETFROM 2
```

b) Écrire les transferts de registres pour l'instruction BR de la WKC (décrite en page 13). La WKC a pour registres PC, SP, FP, IR, MAR, MDR et un registre Y utile pour les calculs arithmétiques (supposez le fetch déjà complété et faites uniquement les transferts propres à l'instruction BR).

Instructions de la WKC

ADD	dépile deux données et empile leur somme
MUL	dépile deux données et empile leur produit
DIV	dépile y, depile x, puis empile x/y
SUB	dépile y, depile x, puis empile x-y
AND	dépile deux valeurs et empile leur AND
OR	dépile deux valeurs et empile leur OR
NOT	remplace le dessus de pile par sa négation
NEQ	dépile deux valeurs, empile 1 si elles sont inégales, sinon empile 0
EQ	dépile deux valeurs, empile 1 si elles sont égales, sinon empile 0
LT	dépile y, dépile x, empile 1 si $x < y$, sinon empile 0
GT	dépile y, dépile x, empile 1 si $x > y$, empile 0 sinon
BRIF	dépile une adresse, dépile un "booléen", branche à l'adresse si le "booléen" n'est pas 0.
BR	dépile une adresse et y branche, i.e. met dans le registre PC l'adresse dépilée
READIN	lit en entrée et empile la valeur lue
PROUT	dépile une valeur et l'imprime en sortie
NOOP	ne fait rien
PUSH n	empile l'entier n (i.e. incrémente SP et met n à l'adresse dans SP)
PUSH n(FP)	empile la valeur qui est contenue à l'adresse FP+n
POP n(FP)	dépile et met la valeur dépilée dans l'adresse FP+n
CALL n	dépile l'adresse du code où l'on va brancher empile la valeur contenue dans le registre PC (déjà incrémenté) empile la valeur contenue dans le registre FP met dans le registre FP la valeur $SP - n - 1$ met dans le registre PC l'adresse qui avait été dépilée.
RETFROM n	met dans PC le contenu de l'adresse FP+n met dans SP la valeur contenue dans FP met dans FP la valeur à l'adresse FP+n+1
CHBIT15	inverse le bit le plus significatif (le bit de signe) de la valeur en dessus de pile.
HALT	arrête l'exécution

Sur cette machine 0 représente Faux et toute valeur non nulle donne Vrai.